STATUS OF THESIS

Title of thesis:	QoS Based Energy Efficient Routing in Wireless Sensor Network

I,______________________________BABAR NAZIR______________________________

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

☐ Confidential

√ Non-confidential

If this thesis is confidential, please state the reason:

____

____

The contents of the thesis will remain confidential for _______ years.

Remarks on disclosure:

____

____

Endorsed by

________________________________ ________________________________

Signature of Author — Signature of Supervisor

Permanent Address:
Banda Sinjilian
Abbottabad, Pakistan

Name of Supervisor
Dr. Halabi Bin Hasbullah

Date :_________________________ Date :_________________________

UNIVERSITI TEKNOLOGI PETRONAS

QoS BASED ENERGY EFFICIENT ROUTING IN WIRELESS SENSOR NETWORK

by

BABAR NAZIR

The undersigned certify that they have read, and recommend to the Postgraduate Studies Programme for acceptance this thesis for the fullfilment of the requirements for the degree stated.

Signature: __________________________________

Main Supervisor: Dr. Halabi Bin Hasbullah

Signature: __________________________________

Head of Department: Dr. Mohd Fadzil Bin Hassan

Date: __________________________________

QoS BASED ENERGY EFFICIENT ROUTING IN WIRELESS SENSOR NETWORK

by

BABAR NAZIR

A Thesis

Submitted to the Postgraduate Studies Programme

As a requirement for the degree of

DOCTOR OF PHILOSOPHY

INFORMATION TECHNOLOGY

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SERI ISKANDAR,

PERAK,

MALAYSIA

NOVEMBER 2011

DECLARATION OF THESIS

Title of thesis	QoS Based Energy Efficient Routing In Wireless Sensor Network

I,___________________________BABAR NAZIR___________________________ hereby declare that the thesis is based on my original work except for quotations and citations, which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Witnessed by

__________________________	__________________________
Signature of Author	Signature of Supervisor
Permanent address:	Name of Supervisor
Banda Sinjilian	Dr. Halabi Bin Hasbullah
Abbottabad, Pakistan	
Date : ____________________	Date : _____________________

To my beloved parents

ACKNOWLEDGEMENT

I would like to begin, thanking Allah SWT, for giving me strength and courage for the successful completion of this work.

It brings me immense pleasure to express my deepest sense of gratitude and appreciation to my thesis supervisor Dr. Halabi Bin Hasbullah for his encouragement, support and guidance throughout this work. I owe my most sincere gratitude to him for generously sharing his time and knowledge during my candidature. His ever-encouraging attitude and constant facilitation, has always been source of motivation and guidance for me, to remain focused on achieving my goals.

I would like to express my gratefulness to Universiti Teknologi PETRONAS, Malaysia for providing financial support under the Graduation Assistantship Scheme, to complete my degree. I also appreciate efforts and cooperation of management of Universiti Teknologi PETRONAS, Malaysia and Computer & Information Sciences department for their support and help during my candidature.

I wish to convey my heartfelt thanks to my parents, brothers, sister and everyone in my family. They gave me endless support and love, and provided me with the opportunity to reach this far with my studies. They stood by me at all times and are always source of constant support, encouragement and love. There is nothing, I could have accomplished without them by my side.

I owe my sincere gratitude to all my friends and colleagues in UTP, especially Sheikh Tahir Bukhsh, Tadiwa Elisha and Irshad Sumra. I shall always cherish the help and support received from them.

Last, but not least I would like to thank all those researchers, academicians and anonymous reviewers, who review this work and generously helped me during course of this research. Their expert opinion and valuable suggestions have helped me to improve the quality of this research work

ABSTRACT

A Wireless Sensor Networks (WSN) is composed of a large number of low-powered sensor nodes that are randomly deployed to collect environmental data. In a WSN, because of energy scarceness, energy efficient gathering of sensed information is one of the most critical issues. Thus, most of the WSN routing protocols found in the literature have considered energy awareness as a key design issue. Factors like throughput, latency and delay are not considered as critical issues in these protocols. However, emerging WSN applications that involve multimedia and imagining sensors require end-to-end delay within acceptable limits. Hence, in addition to energy efficiency, the parameters (delay, packet loss ratio, throughput and coverage) have now become issues of primary concern. Such performance metrics are usually referred to as the Quality of Service (QoS) in communication systems. Therefore, to have efficient use of a sensor node's energy, and the ability to transmit the imaging and multimedia data in a timely manner, requires both a QoS based and energy efficient routing protocol. In this research work, a QoS based energy efficient routing protocol for WSN is proposed. To achieve QoS based energy efficient routing, three protocols are proposed, namely the QoS based Energy Efficient Clustering (QoSEC) for a WSN, the QoS based Energy Efficient Sleep/Wake Scheduling (QoSES) for a WSN, and the QoS based Energy Efficient Mobile Sink (QoSEM) based Routing for a Clustered WSN.

Firstly, in the QoSEC, to achieve energy efficiency and to prolong network/coverage lifetime, some nodes with additional energy resources, termed as super-nodes, in addition to normal capability nodes, are deployed. Multi-hierarchy clustering is done by having super-nodes (acting as a local sink) at the top tier, cluster head (normal node) at the middle tier, and cluster member (normal node) at the lowest tier in the hierarchy. Clustering within normal sensor nodes is done by optimizing the network/coverage lifetime through a cluster-head-selection algorithm and a sleep/wake scheduling algorithm. QoSEC resolves the hot spot problem and prolongs network/coverage lifetime.

Secondly, the QoSES addressed the delay-minimization problem in sleep/wake scheduling for event-driven sensor networks for delay-sensitive applications. For this purpose, QoSES assigns different sleep/wake intervals (longer wake interval) to potential overloaded nodes, according to their varied traffic load requirement defined a) by node position in the network, b) by node topological importance, and c) by handling burst traffic in the proximity of the event occurrence node. Using these heuristics, QoSES minimizes the congestion at nodes having heavy traffic loads and ultimately reduces end-to-end delay while maximizing the throughput.

Lastly, the QoSEM addresses hot spot problem, delay minimization, and QoS assurance. To address hot-spot problem, mobile sink is used, that move in the network to gather data by virtue of which nodes near to the mobile sink changes with each movement, consequently hot spot problem is minimized. To achieve delay minimization, static sink is used in addition to the mobile sink. Delay sensitive data is forwarded to the static sink, while the delay tolerant data is sent through the mobile sink. For QoS assurance, incoming traffic is divided into different traffic classes and each traffic class is assigned different priority based on their QoS requirement (bandwidth, delay) determine by its message type and content. Furthermore, to minimize delay in mobile sink data gathering, the mobile sink is moved throughout the network based on the priority messages at the nodes. Using these heuristics, QoSEM incur less end-to-end delay, is energy efficient, as well as being able to ensure QoS.

Simulations are carried out to evaluate the performance of the proposed protocols of QoSEC, QoSES and QoSEM, by comparing their performance with the established contemporary protocols. Simulation results have demonstrated that when compared with contemporary protocols, each of the proposed protocol significantly prolong the network and coverage lifetime, as well as improve the other QoS routing parameters, such as delay, packet loss ratio, and throughput.

ABSTRAK

Rangkaian Penderia Tanpa-wayar (*Wireless Sensor Network, WSN*) adalah terdiri daripada sejumlah besar nod-nod penderia berkuasa-rendah yang ditempatkan secara rawak untuk mengumpulkan data persekitaran. Di dalam WSN, disebabkan oleh keterbatasan tenaganya, kecekapan tenaga dalam pengumpulan maklumat yang dideriakan merupakan suatu isu yang kritikal. Oleh itu, kebanyakan protocol-protokol peroutan WSN yang terdapat di dalam kepustakaan telah mempertimbangkan peka-tenaga sebagai kunci isu reka-bentuk. Bagaimanapun, factor-faktor celusan, masa pendam, dan lengah telah tidak dipertimbangkan sebagai isu kritikal di dalam protokol-protokol tersebut. Hari ini, aplikasi-aplikasi WSN baru muncul melibatkan deriaan multimedia dan pengimejan yang memerlukan lengah hujung-ke-hujung di dalam batasan boleh-terima. Oleh itu, sebagai tambahan kepada kecekapan tenaga, parameter-parameter yang telah dikenalpasti, iaitu celusan, masa pendam dan lengah, sekarang menjadi isu pertimbangan yang utama. Metrik prestasi ini selalunya dirujuk sebagai kualiti perkhidmatan (*Quality of Service - QoS*) di dalam sistem komunikasi. Oleh yang demikian, untuk mencapai kecekapan tenaga oleh nod-nod penderia and kebolehan untuk menghantar data multimedia dan pengimejan dalam masa yang ditetapkan, ianya memerlukan protokol-protokol yang peka-QoS dan cekap-tenaga. Di dalam kerja penyelidikan ini, protokol peroutan untuk WSN yang peka-QoS dan cekap tenaga telah dicadangkan. Untuk mencapai matlamat tersebut, tiga protokol telah dicadangkan, yang dinamakan masing-masingnya sebagai *QoS-aware Energy Efficient Clustering (QoSEC)*, *QoS-aware Energy Efficient Sleep/wake Scheduling (QoSES)*, dan *QoS-aware Energy Efficient Mobile Sink (QoSEM)*.

Pertama, di dalam QoSEC, untuk mencapai kecekapan tenaga dan memanjangkan jangkahayat rangkaian/liputan, beberapa nod dengan punca tenaga berlebihan, dipanggil *super-nodes*, sebagai tambahan kepada nod-nod berkeupayaan biasa, adalah ditempatkan. Pengklusteran berbagai-peringkat dilakukan dengan *super-node* (bertindak sebagai sink tempatan) berada pada tingkat atas, kepala kluster (nod biasa) berada pada tingkat tengah, dan ahli kluster (nod biasa) berada pada tingkat bawah di

dalam susunan hiraki. Pengklusteran dalaman untuk nod-nod penderia biasa dibuat dengan mengoptimumkan jangkahayat rangkaian/liputan melalui algoritma *cluster-head-selection* dan algoritma *sleep/wake scheduling*. QoSEC dapat menyelesaikan masalah titik-panas dan memanjangkan jangkahayat rangkaian/liputan.

Kedua, QoSES menyelesaikan masalah meminimumkan-lengah di dalam penjadualan *sleep/wake* untuk rangkaian penderia dipacu-kejadian ke atas aplikasi-aplikasi sensitif-lengah. Untuk tujuan ini, QoSES memberikan *sleep/wake* berbeza kepada nod-nod yang mungkin terbeban, mengikut keperluan beban trafik yang berubah yang ditentukan a) oleh kedudukan nod di dalam rangkaian, b) oleh kepentingan topologi nod, dan c) oleh keupayaan mengendalikan letusan trafik di dalam kawasan kejadian berdekatan nod. Dengan heuristik ini, QoSES meminimumkan kesesakan pada nod-nod yang mempunyai beban trafik tinggi, dan akhirnya mengurangkan lengah hujung-ke-hujung dan memaksimumkan celusan.

Ketiga, QoSEM menyelesaikan masalah titik-panas, meminimumkan lengah, dan jaminan QoS. Untuk masalah titik-panas, sink bergerak digunakan yang bergerak di dalam rangkaian untuk mengumpul data dengan keupayaan bahawa non-nod berdekatan sink bergerak akan berubah mengikut setiap pergerakan, dan seterusnya masalah titik-panas diminimumkan. Untuk meminimumkan lengah, sink statik digunakan sebagai tambahan kepada sink bergerak, di mana data sensitif-lengah dimajukan kepada sink static, sementara data hadterima-lengah dihantar melalui sink bergerak. Untuk jaminan QoS, trafik masukan dibahagikan kepada beberapa kelas berbeza, dan setiap kelas trafik diberikan keutamaan berbeza berdasarkan kepada keperluan QoS (umpamanya jalurlebar dan lengah) yang ditentukan oleh jenis mesej dan kandungan mesejnya. Seterusnya, untuk meminimumkan lengah di dalam pengumpulan data sink bergerak, sink bergerak tersebut digerakkkan di dalam rangkaian berdasarkan kepada keutamaan mesej-mesej pada non-nod. Dengan menggunakan heuristik ini, QoSEM melibatkan lengah hujung-ke-hujung yang kurang, cekap tenaga, serta berupaya menjamin QoS.

Simulasian telah dilakukan untuk menilai prestasi protokol-protokol yang dicadangkan, iaitu QoSEC, QoSES dan QoSEM, dengan membandingkan prestasi mereka dengan protocol kontemporari yang tersedia wujud. Keputusan-keputusan simulasian telah menunjukkan bahawa setiap protokol yang dicadangkan telah dengan signifikannya memanjangkan jangkahayat rangkaian dan jangkahayat liputan, serta juga telah memperbaikki prestasi parameter-parameter peroutan QoS lainnya, iaitu celusan, masa pendam, lengah, dan kadar kehilangan paket.

TABLE OF CONTENT

TABLE OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADV	Advertisement
ANN	Announcement
BS	Base Station
CH	Cluster Head
CLS	Clear to send
CPCP	Coverage Preserving Clustering Protocol
EEMC	Energy Efficient Multilevel Clustering
GUI	Graphical User Interface
LEACH	Low Energy Adaptive Clustering Hierarchy
MAC	Medium Access Control
MANET	Mobile Adhoc Network
MEMS	Micro-Electro-Mechanical-System
OMNeT++	Objective Modular Network Testbed in C++
QoS	Quality of Service
RAM	Random Access Memory
RFID	Radio Frequency Identification
ROM	Read Only Memory
RTS	Request to Send
S-MAC	Sensor Medium Access Control
TDMA	Time Division Multiple Access
WI	Wake Interval
WSN	Wireless Sensor Network

CHAPTER 1

INTRODUCTION

This chapter presents a general overview of a Wireless Sensor Network (WSN). A brief description of QoS based energy efficient routing in a WSN in the perspective of the clustering approach, sleep/wake scheduling approach and mobile sink approach are presented. Next to follow is the statement of the problem, motivation to carry out this research work, and objectives to be accomplished during the research work. The balance of the chapter covers the scope, assumptions and limitations of the research work, research approach adopted and the research activities followed during this work. Finally, the chapter is concluded with the contributions and organization of the thesis.

1.1 Introduction to the Wireless Sensor Network

Recent development in Micro-Electro-Mechanical Systems (MEMS) has made the deployment of small sized sensor nodes possible and inexpensive (Akyildiz et al. 2002; Jennifer et al. 2008). The deployment of these sensor nodes for monitoring different events in a targeted environment is termed as a Wireless Sensor Network (WSN); this is illustrated in Figure 1.1. In a typical WSN setup, the sensors are randomly deployed in a region of interest. Placed near to these nodes is a Base Station (BS) or Sink, which is ultimately connected to the Internet. The primary task of the BS is to give commands to these nodes and to gather sensed data from the sensor nodes. Data is then accessible to end users through the Internet. To avoid redundancy in the gathered data, the sensor nodes sense the physical phenomenon and perform local data aggregation, before further transmitting the sensed data to the BS

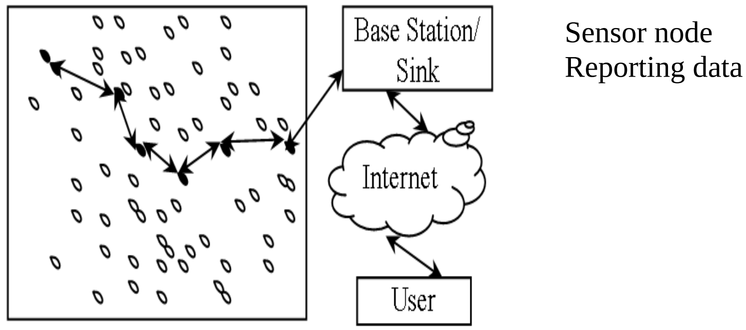


Figure 1.1 A typical WSN Setup with sensor nodes and Base Station/Sink

1.2 QoS Based Energy Efficient Routing in the WSN

Quality of Serivce (QoS) refers to the ability of communication system to give different priorities to various traffics, based on their requirements to guarantee certain level of performance in the network, by controlling the resource sharing. QoS assures to provide certain level of service attributed by performance metrircs such as delay, packet loss ratio, throughput and coverage. Such performance metrics are usually known as the Quality of Service (QoS) of the communication network.

In typical WSN, sensor nodes are generally equipped with low processing power, low sensing power, less memory, and short-range communication, and most importantly have very limited battery power (Edgar and Callaway 2003). Randomly deployed in difficult terrain makes it almost impossible to recharge or replace the dead battery. Therefore, battery power in WSNs is considered a scarce resource and should be used efficiently. The terrm energy efficiency refers to the economical and efficient use of battery/energy of sensor node in WSN communication/routing protocols that can lead to prolong network lifetime.

Recently, routing protocols for the WSN proposed in literature, mostly consider energy efficiency as a key design issue. However, the advent of imaging sensors and multimedia applications of the WSN has defined new challenges in WSN protocol design. In such applications, in addition to energy efficiency, other issues of primary concern are delay, packet loss ratio, throughput and coverage. Such performance metrics are usually referred to as the Quality of Service (QoS) of the communication network. Thus, routing multimedia data needs careful handling to have energy efficiency as well as timely delivery of data within an acceptable delay range. In this thesis, a QoS based energy efficient routing protocol in a WSN is considered. To achieve QoS and energy efficiency in this thesis, three potential prospects are targeted, namely the clustering protocol, sleep/wake protocol and mobile sink based protocol. The brief descriptions of clustering approach, sleep/wake approach, and mobile sink approach are shown below:

1.2.1 Clustering Approach

In the cluster-based approach, the whole WSN is divided into a number of small regions known as clusters. In each cluster, one node is elected as the Cluster Head (CH). The elected CH is responsible for aggregating the sensed data from its cluster member nodes and forwarding it to a BS or to the next CH. As the CH has to relay the data of all member nodes, and if it is continuously selected as CH for a longer period, its energy will be depleted earlier than the other cluster member nodes. Ultimately, these nodes die, which leads to the emergence of energy holes in the WSN. To solve this problem, the task of CH is periodically assigned in a round fashion to other nodes. In each round, responsibility of the CH is alternately taken by a different cluster member node in a cluster. A new CH is selected randomly or based on parameters such as residual energy, distance from BS, connected nodes, and topology structure. To forward data to a BS, multi-hop CH communication is performed, where every CH sends sensed data to the other CH in its neighborhood, which is near to the BS.

1.2.2 Sleep/wake Scheduling Approach

Sleep/wake scheduling is an effective energy-conservation mechanism. The breakup of the energy expenditure has shown that radio components consume much more energy than sensing and processing sub-systems. The radio transceiver of a sensor node performs two tasks: transmitting and receiving data. It has three states/modes:

active state, idle state and sleep state. In the active (transmit) state, the transmit part of the transceiver is active and the antenna radiates energy. In the active (receive) state the receive part is active. A transceiver that is ready to receive but not receiving is said to be in an idle state. In the idle state, many parts of the receive circuitry are active. When a sensor node is in the low power state and significant parts of the transceiver are switched off, it said to be in the sleep state.

Generally WSNs operate for a long time in idle mode and only occasionally send data. Energy consumption when listening to the idle channel is equivalent to its energy consumption when sending or receiving, and much higher than the energy consumption while in the sleep mode (Akyildiz et al. 2002). This fact is exploited by sleep/wake scheduling techniques and effort is made to reduce this energy wastage in the idle mode. The mechanism of alternately switching between the sleep and wakeup mode to avoid energy wastage is termed as sleep/wake scheduling or the duty cycling scheme. Sleep/wake scheduling is regarded as the most effective mechanism for energy conservation and significantly prolongs the network lifetime in a WSN. To achieve the minimum energy consumption, the transceiver should be mostly in the sleep state and be activated (wakeup state) only when required. Clearly, the best scenario is that a sensor node wakes up once and finishes all operations continuously, but practically it is not possible to have such an ideal coordinated and synchronized environment. Thus, the main objective of sleep/wake scheduling is to put the transceiver in the sleep state during idle periods (when there are no events). Significant energy savings can be accomplished by designing sleep/wake schemes that can let the sensor's transceivers remain in sleep mode for as long as possible, and at the same time retain network connectivity and coverage.

A variety of sleep/wake scheduling protocols has been proposed in the literature. Most of them use periodic sleep/wake intervals and provide effective energy conservation at the cost of delay and throughput. For example, for a source node to transmit data, it has to know the sleep/wakeup schedule of the neighbor node and has to wait for the neighbor to come to the active state. The same is repeated until the data reaches the final destination thus resulting in unprecedented delays. This increase in delay is equal to the product of the number of intermediate forwarders times the length of the wakeup interval (Lee et al. 2008). Such increase in end-to-end delay incurred due to the latency-energy trade off has the potential to become a major problem in many emerging delay-sensitive WSN applications, which require a fast response and real-time control.

1.2.3 Mobile Sink Approach

In this section, QoS based energy efficient routing in the WSN is achieved by introducing mobility in the network. It could be done by introducing mobile sensor nodes, a mobile relay or a mobile sink.

Routing data from source sensor nodes to a sink is a common and challenging task in most of WSN applications. In a typical WSN architecture, the sink is static and sensor nodes use multi-hop communication to forward their data to the sink. Each sensor node sends data to neighboring sensor nodes, and so on until the data reaches the BS. Ultimately, nodes nearest to a BS send data directly to the BS. However, the potential disadvantage of this technique is that the sensor nearest to the BS may over utilize their energy, resulting in depletion of the sensor node's energy nearest to the BS. This problem is termed as a hot spot or energy-hole problem. A hot spot problem refers to the phenomenon when the nodes near the sink quickly drain their energy on account of relaying the data of the nodes farther from the sink. As a result, although the nodes farther from the sink have significant energy left, their energy cannot be utilized as the nodes near to the sink have depleted their energy. Hence, data cannot be sent to the sink across the hot spot or energy hole near the sink. Consequently, they cannot forward their data to the BS and it limits the network lifetime. The hot spot problem is an inherent phenomenon in WSNs with a static sink, and can significantly decrease the lifetime of the overall network. The static nature of the sink is the main reason behind the hot spot problem, as the same nodes near the sink have to forward the data all the time. The main advantage, though, of the static sink approach is that it involves less end-to-end delay.

In recent years, contrary to the static sink, the mobile sink approach has attracted much research interest because of an increase in its potential with WSN applications and its potential to improve network performance such as energy efficiency and throughput. However, the main disadvantage of the mobile sink approach is that it encounters long end-to-end delay.

1.3 Problem Statement

In this thesis, the QoS based energy efficient routing protocol for the WSN is presented, which can extend the network lifetime as well as be able to full fill the QoS requirement of the data. The proposed protocols consider energy efficiency as well as QoS as primary design objectives. To achieve QoS and energy efficiency in this thesis, the main problem is divided in to three sub-problems namely: clustering, sleep/wake and mobile sink protocols. A brief description of the problems addressed in these three issues is as follows:

1.3.1 Clustering Approach

Clustering has been identified to be an energy-efficient routing technique in WSNs. With the clustering approach, data forwarding to the sink/BS is performed by using multi-hop communication; in this way, each CH sends data to its neighboring CH, and this continues until the data finally reaches the BS. Ultimately, the BS receives data from the nodes nearest to it directly. However, a possible disadvantage of this technique is that, the CH nearest to the BS may utilize too much of its energy, which results in depletion of the sensor node's energy nearest to the BS. This situation is called a *hot spot* or *energy-hole problem* and it is an inherent phenomenon in WSNs, and can cause a significant decrease the lifetime of the overall network. As in this scenario, though the nodes which are a farther distance away from the BS still have a significant amount of energy left, but their energy cannot be utilized as the nodes nearer to the BS have already died. As a result, they cannot forward their data to the BS. The other problem is that, in most of the algorithms presently available for CH selection, the optimization is done on a single parameter; this could be residual energy, distance from the BS, connected nodes, coverage lifetime, and so forth. Moreover, the optimization of this single parameter has resulted in performance degradation for the other parameters. For instance, if residual energy is considered as the optimization parameter, it may optimize the lifetime of the network, but the coverage of the network will be the trade-off and vice versa.

To address these two problems, a QoS based energy efficient clustering protocol for the WSN has been proposed. (Section 3.1.1 contains the detailed problem statement of the proposed protocol).

1.3.2 Sleep/wake scheduling approach

Sleep/wake scheduling has been used to save energy and extend the network lifetime. Energy efficiency has an inherent tradeoff with delay; thus, generally in such sleep/wake scheduling strategies, maximization of network lifetime is achieved at the expense of increase in delay. This delay is unacceptable for many delay-sensitive applications, which requires the event reporting delay to be small; such situations would be in military surveillance, tsunami warnings, seismic detection, biomedical health monitoring, hazardous environment sensing, and fire detection. In most sleep/wake schemes found in literature, all nodes have the same generic sleep/wake schedule and each node takes autonomous wake up decisions without considering their neighbors in order to save energy. However, as WSNs use multi-hop communication, every node has one designated next-hop forwarding node in the neighborhood and to do the transmission, the sender node has to wait for the arrival of the wakeup time of the next hop forwarding node. Similarly, the next relay nodes have to wait for the wakeup interval of its next hop and so on, until the message reaches the BS. Consequently, due to the different wakeup intervals, additional delays are added at each hop as the packet is forwarded along the path to the sink; this is because each node has to wait for its next-hop node to wake up before it can transmit the packet. All these delays at each hop contribute to the final end-to-end delay of the packet. The previous studies in sleep/wake scheduling are able to achieve energy efficiency, but do not consider these delays, which occur during sleep/wake scheduling.

To address the delay minimization problem, a QoS based energy efficient sleep/wake scheduling protocol for the WSN have been proposed. (Section 3.2.1 contains the detailed problem statement of the proposed protocol).

1.3.3 Mobile Sink Approach

In WSNs, sensors near the static sink have to relay the data of the nodes away from the sink and as a result they drain their energy very quickly. It results in network partitioning and can significantly limit the network lifetime; this problem is termed as a hot spot problem. Contrary to the static sink, in recent years, the mobile sink approach has been used to address the hot spot problem but it increases end-to-end delay which is not acceptable for delay sensitive applications. Additionally, recently with an increase in multimedia WSN applications, QoS in routing has emerged as a critical issue. Sensor nodes generate different types of traffic which has differentiated forwarding requirements based on their bandwidth and delay.

To address three issues namely, the hot spot problem, delay minimization and QoS assurance, a QoS based energy efficient mobile sink based routing protocol for the WSN have been proposed. (Section 3.3.1 contains the detailed problem statement of the proposed protocol).

1.4 Motivation

Sensor nodes have limited energy and low bandwidth constraints. These constraints are added to by large scale deployment made energy aware as core design issues at different layers of the network protocol stack including the network layer. Due to the large scale, random deployment, limited hardware resources, un-rechargeable battery power, hostile environment and failure prone nature routing in WSNs has been made a challenging area in WSNs. In routing, most of the protocols found in literature have considered energy awareness as a key design issue in order to maximize the network (Chalermek et al. 2000; Estrin et al. 1999; Heinzelman 2000; Shah and Rabaey 2002; Sohrabi et al. 2000; Wendi Rabiner et al. 1999; Younis et al. 2002). Factors like throughput, latency and delay are not issues of primary concern in these protocols and the approach is acceptable, since they are mostly dealing with a small amount of data flowing in low rates. However, with the emergence of new WSN applications that involve multimedia and imagining sensors, routing in WSNs has come across new challenges. Reporting of data in these multimedia and imaging WSN applications, requires minimum end-to-end delay within acceptable limits. In such applications, in addition to energy efficiency; latency, throughput and delay also become issues of primary concern. Such performance metrics are usually referred to as the quality of service (QoS) of the communication network.

Therefore, to have efficient use of sensor node energy and ability to report the imaging and multimedia data in a timely manner within an acceptable range, requires both energy and a QoS based routing protocol. Thus, there is need for a paradigm shift from conventional energy efficient protocols to the recent QoS based energy efficient protocols. WSN applications which benefit from QoS based protocols include military surveillance, real time target tracking in battle environments, tsunami alarms, smart hospitals, seismic detection, biomedical health monitoring, hazardous environment sensing, fire detection, intrusion detection, disaster monitoring, and real-time control. Generally, these applications deal with real time data and need a certain bandwidth with the minimum possible delay. In such scenarios, to guarantee the reliable delivery of real-time data, a service differentiation mechanism is needed. QoS based energy efficient routing in WSNs will provide the energy efficient path as well as guarantee certain bandwidth with the minimum possible delay. The need for QoS based energy efficient schemes that can prolong the network lifetime as well as guarantee QoS motivated this research. The main motivation behind this work is to study various issues and strategies of QoS based energy efficient data gathering and propose a QoS based energy efficient routing protocol for wireless sensor networks. The proposed protocols consider energy efficiency as well as QoS as primary design objectives and can be applied to emerging delay constrained WSN applications. To achieve QoS and energy efficiency in this thesis, three heuristics namely: clustering, sleep/wake and mobile sink protocols are proposed.

1.5 Objectives

This work is focused on routing that can provide energy efficiency as well as guarantee QoS, while reporting the data from sensor nodes to the BS. The main objective of this thesis is to provide QoS based energy efficient routing for WSNs at three levels: clustering approach, sleep/scheduling approach and mobile sink approach. In this perceptive the primary objectives of this thesis are the following.

1.5.1 To study and analyze the QoS and energy efficiency problems in clustering. To design, develop and evaluate a clustering protocol for WSNs that can extend network lifetime as well guarantee QoS.

1.5.2 To study and analyze the QoS and energy efficiency problems in the sleep/wake scheduling approach. To design, develop and evaluate a sleep/wake scheduling algorithm for WSNs that can that can extend network lifetime as well guarantee QoS.

1.5.3 To study and analyze the QoS and energy efficiency problems in mobile sink based routing in WSNs. To design, develop and evaluate a mobiles sink based routing protocol for WSNs that can extend network lifetime as well guarantee QoS.

1.5.4 To verify through extensive simulation the performance of the proposed protocols with the established and state of the art contemporary protocols.

1.6 Scope and Limitations

The solutions proposed in this study provide WSNs routing, which can increase network lifetime as well as guarantee QoS with low delay, high throughput, low packet loss and better coverage. This research studies the QoS based energy efficient routing in WSNs. It considers solutions to the QoS based energy efficient routing at three levels: clustering approach, sleep/wake scheduling approach, and mobile sink based approach. It develops protocols for QoS based energy efficient clustering (QoSEC) protocol, QoS based energy efficient sleep/wake (QoSES) scheduling protocol and QoS based energy efficient mobile sink (QoSEM) based routing protocol for WSNs. Afterwards, simulations are carried out to evaluate the performance of the proposed protocols in comparison with contemporary protocols. The proposed protocols deals with QoS based routing considering the above three issues; issue like network deployment, data aggregation, data fusion and security issues, are out of scope of this research. The proposed protocols also deal with static sensor nodes; how routing is done in sensor nodes with mobility is out of the scope of this work. The proposed protocols work at the network layer and use available protocols for other layers. In-sight details on how data is handled and communication takes place at other layers are also not in the scope of this thesis.

1.7 Assumptions

This section outlines the assumptions that were made in this thesis. Supported by literature review and evident from most of emerging WSN applications, it is assumed that the WSNs considered in this research deal with static sensor nodes with non-rechargeable limited batteries. Nodes are randomly deployed across a sensing field, nodes cannot communicate directly with the BS, and use multi-hop communication to propagate the data to the BS, which is located away from sensing field. Supported by the majority of WSN applications, it is assumed that the BS is rich in resources with a constant power supply and can directly communicate with all the nodes in the network. The focus of this research work is on the network layer; operations at other layers are assumed to be working appropriately. It is important to note that in proposed model, no assumptions are made about 1) the homogeneity of node dispersion in the field, 2) the network density or diameter, 3) the distribution of energy consumption among sensor nodes, 4) the proximity of querying observers, 5) the ability to communicate with the BS, and 6) each node having a fixed communication range.

1.8 Research Approach

In order to develop a QoS based energy efficient routing protocol for WSNs, the steps followed are described below:

a. Study and analyze the different types of QoS based energy efficient routing protocols in WSNs. Analyze the potential issues and problems in exiting QoS based energy efficient routing protocols.
   i. QoS based energy efficient clustering protocol for WSNs.
   ii. QoS based energy efficient sleep/wake scheduling protocol for WSNs.
   iii. QoS based energy efficient mobile sink based routing protocol for WSNs.
b. Design the QoS based energy efficient clustering protocol for WSNs to achieve maximum energy efficiency and guarantee QoS.
c. Design the QoS based energy efficient sleep/wake scheduling protocol for WSNs to achieve maximum energy efficiency and guarantee QoS.
d. Design the QoS based energy efficient mobile sink based protocol for WSNs to achieve maximum energy efficiency and guarantee QoS.
e. Implement, prototype and simulate the architecture of the three proposed protocols. The protocols are implemented using OMNet++, which is a well known discrete event simulator. Extensive simulations are carried out for the performance comparison of the proposed protocols with contemporary protocols.

1.9 Research Activities

The overall approach or methodology of the thesis can be described as follows (see Figure 1.2):

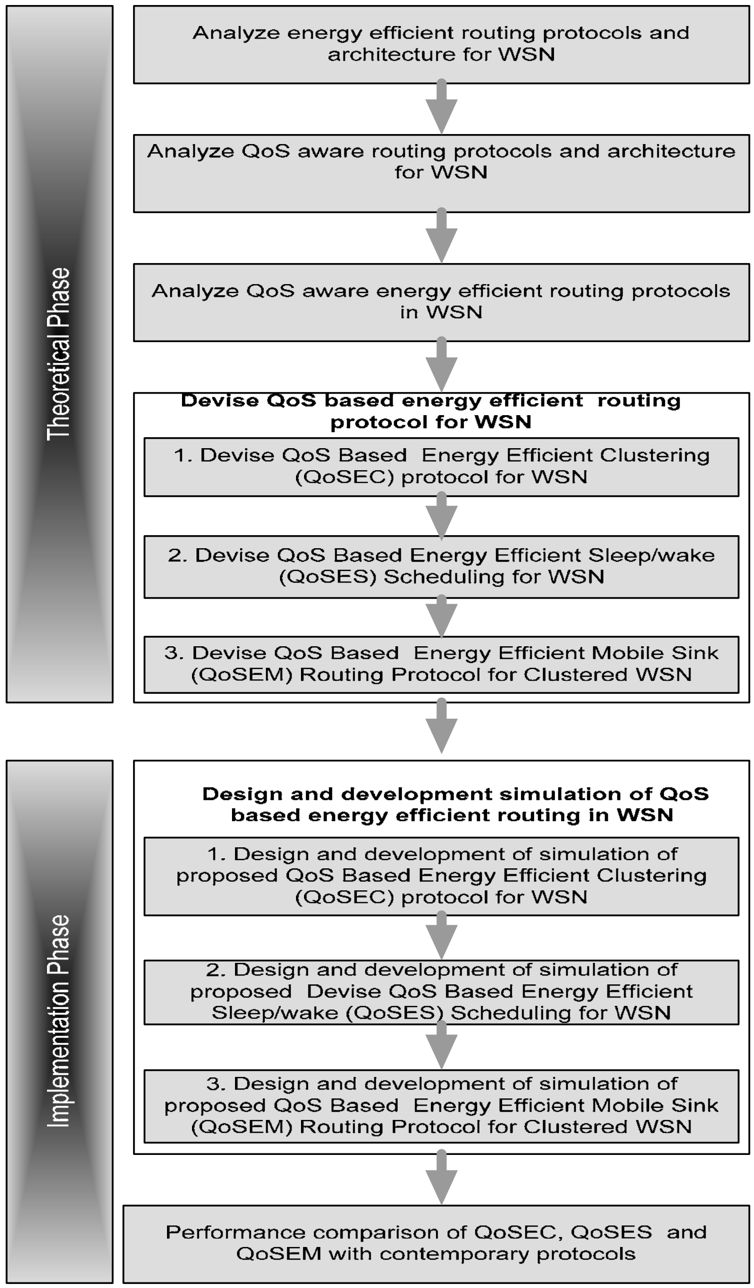


Figure 1.2 Research Activities

1.10 Research Contributions

This thesis makes the following research contributions to the current state-of-the-art in the field of QoS based energy efficient routing in WSNs. The main contributions of this thesis work are as follows.

1.10.1 Propose, develop and analyze a new QoS based energy efficient clustering (QoSEC) protocol for WSNs to achieve QoS and energy efficiency by guaranteeing minimum end-to-end delay and maximum energy efficiency (Section 3.1 contains the detailed discussion of the proposed QoSEC protocol). There are two (2) main contributions of this protocol.

1.10.1.1 Firstly, a QoS based energy efficient clustering protocol for WSNs is proposed, which addresses two major problems: a) To address the hot spot problem, extra capability nodes in addition to normal nodes are introduced in the WSN. These nodes act as local sinks and collect data from the CH in their vicinity. Thereby, significantly decreasing traffic flow towards the BS and ultimately minimizing the hot spot problem. b) Two algorithms for CH selection and sleep/wake scheduling are proposed. These algorithms use coverage as well as energy, as selection parameters in CH selection and in sleep/wake scheduling. The objective is to increase network lifetime without affecting coverage.

1.10.1.1 Secondly, extensive simulations are carried out that have shown that the performance of the proposed QoSEC protocol is better than the performance of other contemporary protocols like LEACH, CPCP and EEMC. This is because QoSEC prolongs the network and coverage lifetime, as well as improves the other QoS routing parameters, such as delay, packet loss ratio, and throughput.

1.10.2 Propose, develop and analyze a new QoS based energy efficient sleep/wake (QoSES) protocol for WSNs to achieve QoS and energy efficiency by guaranteeing minimum end-to-end delay and maximum energy efficiency (Section 3.2 contains the detailed discussion of the proposed QoSES protocol). The main contributions of QoSES can be summarized as follows:

1.10.2.1 Firstly, the QoS based energy efficient sleep/wake protocol for WSNs has been introduced to address the delay minimization problem. In order to reduce delay, QoSES identifies nodes for different sleep/wake intervals based on their traffic load at three levels: a) Bbased on their distance from the BS. b) Based on their topological importance in the network. c) Based on handling burst traffic in the vicinity of the specific node where the event occurs. It then assigns different active intervals to the nodes, according to their variable traffic load requirement. Using these heuristics the proposed protocol reduces end-to-end delay and maximizes the throughput by minimizing the congestion at nodes with heavy traffic loads.

1.10.2.2 Lastly, extensive simulations are carried out to evaluate the performance of the QoSES, by comparing its performance with Anycast and S-MAC protocols. The results demonstrate that QoSES has successfully minimized the end-to-end delay, as well as has improved the other QoS routing parameters like average energy per packet, packet loss ratio, throughput and coverage lifetime.

1.10.3 Propose, develop and analyze a new QoS based energy efficient mobile sink (QoSEM) based protocol for Clustered WSNs to achieve QoS and energy efficiency by guaranteeing minimum end-to-end delay and maximum energy efficiency (Section 3.3 contains the detailed discussion of the proposed QoSEM protocol). The main contributions of QoSEM can be summarized as follows:

1.10.3.1 Firstly, the QoS based energy efficient mobile sink based protocol for Clustered WSNs is proposed, which addresses three major problems (hot-spot problem, delay minimization and QoS assurance): a) To address the hot-spot problem, a mobile sink is used, which moves across the network to gather data. b) To address delay minimization in the mobile sink approach, a static sink is used in addition to the mobile sink. Delay sensitive data is forwarded to the static sink while the delay tolerant data is sent through the mobile sink. c) For QoS assurance, incoming traffic is prioritized and divided into different classes based on the message type and content. The objective is to increase network lifetime and provide QoS.

1.10.3.2 Lastly, extensive simulations are carried out to evaluate the performance of the QoSEM, by comparing its performance with static sink and mobile sink approaches. The results demonstrate that QoSEM has successfully minimized the end-to-end delay, as well as improved the other QoS routing parameters like average energy per packet, packet loss ratio, and throughput.

1.11 Thesis Organization

The rest of this thesis is organized as follows:

- Chapter 2 provides the overall background knowledge, and discusses the related work relevant to this research work that forms the core of this thesis. It includes an overview of WSNs, parts, type and hardware configuration of a typical sensor node, applications of WSNs and challenges encountered in protocol design for WSNs communication protocol. Furthermore, it discusses the literature review in the context of three prospects: clustering approach, sleep/wake scheduling approach, and mobile sink based approach. The clustering approach sub-section outlines the clustering objectives, discusses the detailed literature review of clustering protocols (Variable convergence time algorithms, Constant convergence time algorithms), and ends the sub-section with a brief description of the problem statement. Next to follow is the sleep/scheduling approach sub-section which discusses the detailed literature review of sleep/scheduling schemes (On-demand Schemes, Scheduled rendezvous schemes, Asynchronous Schemes) and concludes the sub-section with a brief description of the problem statement. Lastly, the mobile sink based approach sub-section discusses the existing literature on mobile sink based approaches (Mechanisms using mobile sinks, Mechanisms using mobile sensors redeployment, Mechanisms using mobile relays) and contains a brief description of the problem statement at the end of sub-section.

- Chapter 3 discusses the three proposed protocols in detail, namely: QoS based Energy Efficient Clustering (QoSEC) for WSNs, QoS based Energy Efficient Sleep/wake (QoSES) Scheduling for WSNs, and QoS based Energy Efficient Mobile Sink (QoSEM) based Routing Protocol for Clustered WSNs. Each protocol discussion starts with the detailed problem description, network model and assumptions, preliminaries (proposed protocol overview, data structures used at sensor node), and a description of the workings of the proposed protocols.
- Chapter 4 starts with the Network Simulator types (Discrete Event Simulator, Continuous Time Simulator), overview of OMNeT++, Selection of the WSN Simulator, Modeling and Simulation using OMNeT++, (OMNeT ++ overview, OMNeT++ Framework and Simulation Design (hardware, radio, network)). The chapter is concluded with a detailed discussion of simulation results (Simulation setup, Event Summary, energy model and simulation results/discussion) for the three proposed protocols: QoSEC, QoSES and QoSEM.
- Chapter 5 presents the conclusion of this research, followed by the key contribution of this research and the chapter is concluded with the highlights of some of the recommendations for future work.
- Lastly, a list of our publications during this research work is provided.

1.12 Summary

In this chapter, a basic overview of the thesis has been presented, covering the fundamentals, like: general overview, statement of problem, motivation, objectives, scope, assumptions and limitations, research approach adopted, research activities, and contributions of the thesis.

CHAPTER 2

BACKGROUND KNOWLEDGE AND LITERATURE REVIEW

Chapter 2 hapter 2 provides the overall background knowledge, and discusses the related work relevant to this research work that forms the core of this thesis. It includes an overview of WSNs, parts, type and hardware configuration of a typical sensor node, applications of WSNs and challenges encountered in protocol design for WSNs communication protocol. A systematic review of the related work and issues in the existing QoS aware energy efficient approaches is presented. Related work on QoS aware routing in WSNs is presented in the perspective of the clustering approach, sleep/wake scheduling, and mobile sink approach. This chapter discusses in detail the clustering approach and the important review of the literature under two major clustering categories: variable convergence time and constant convergence time clustering algorithms. Next to follow is a detailed discussion of different sleep/wake scheduling algorithms, including the important Sleep/wake scheduling scheme under three major categories: On-demand Schemes, Scheduled rendezvous schemes and Asynchronous Schemes. Lastly, the chapter presents a detailed discussion of related work on the mobile sink approach including three main categories: Mechanisms using mobile sinks (Algorithms with pre-determined sink mobility path, Algorithms with autonomous sink movement), Mechanisms using mobile sensors redeployment and Mechanisms using mobile relays are discussed. At the end of each of these three related work sections, potential problem statements are briefly discussed and supported, and proposed solutions are overviewed.

2.1 Overview of Wireless Sensor Network

Recent development in Micro-Electro-Mechanical Systems (MEMS) has made the deployment of small sized sensor nodes possible and inexpensive (Akyildiz et al. 2002; Jennifer et al. 2008). The deployment of these sensor nodes for monitoring different events in a targeted environment is termed as a Wireless Sensor Network (WSN). Karl and Willig (2005).define WSNs as follows:

> *"These networks consist of individual nodes that are able to interact with their environment by sensing or controlling physical parameters; these nodes have to collaborate to fulfill their tasks as, usually, a single node is incapable of doing so; and they use wireless communication to enable this collaboration."*

Olariu (2006) defined a WSN as follows:

> *"A sensor network is a deployment of massive numbers of small, inexpensive, self-powered devices that can sense, compute and communicate with other devices for the purpose of gathering local information to make global decisions about a physical environment."*

In the WSN, tiny sensor nodes equipped with sensors (such as temperature sensors, pressure sensors, and light sensors) are randomly or deterministically deployed over a region of interest (termed as a sensing field) to detect physical phenomenon. Sensor nodes monitor the sensing field for the occurrence of the required physical phenomenon. They send the sensed data to the Base Station (BS), either directly or using multi-hop communication. The WSN design significantly depends on the type of application, and is subject to change depending upon the sensor nodes' hardware resources, environment, cost, and other system constraints (Jennifer et al. 2008).

2.1.1 Typical WSN

In a typical WSN setup, the sensors are randomly deployed in a region of interest or sensing field. Placed near to these nodes is a Base Station (BS) or Sink, which is ultimately connected to the Internet. The primary task of the BS is to give commands to these nodes and to gather sensed data from the sensor nodes. Data is then accessible to end users through the Internet. To avoid redundancy in the gathered data, the sensor nodes sense the physical phenomenon and perform local data aggregation, before further transmitting the sensed data to the BS.

A typical WSN consists of a sensor node, relay or router node, base station, and sink. Sensor nodes do the actual task of sensing the environment and reporting the sensed data to the network. A *relay node* or *router node* is responsible for forwarding the sensed data from the sensor node to the BS. Depending upon the application, the relay/router node can be an ordinary sensor node that forwards the sensed data received from its neighbors to the BS using multi-hop communication or it can be other specialized resource rich nodes responsible for collecting data from the sensor node and relaying it to the BS. In this thesis, the first case is considered, and thus terms node, relay node, and forwarding nodes are interchangeably used. A *base station (BS)*, *access point* or *gateway* is responsible for sending the command/query to all the sensor nodes in the network and collecting the sensed data from the sensor nodes. The BS is always a sink and all the traffic to be reported by the sensor nodes draws to it. It is generally a resource rich device and has no resource constraints as compared to the sensor node and can be connected to another network, for example, the Internet. Using the Internet, the user can access and analyze the sensed data gathered at the BS.

2.1.2 Sensor Node

A *sensor node* in a WSN is an entity that performs the actual task of sensing the environment. It monitors the environment for occurrence of physical phenomenon within its sensing field, and passes the sensed data to the neighboring sensor node. Each sensor node also acts as a *relay node* or *router node* and forwards the packets from their neighboring sensor nodes, until the packet reaches the BS. A typical sensor node consists of processor, memory, sensing, power supply, and transceiver; sometimes it may even include mobility, location, and actuator module. Generally, a sensor node is composed of a low-power CPU, tiny memory (RAM/ROM), RF module, many kinds of sensor devices, and low-power batteries. Four basic components: a sensing unit, a processing unit, a transceiver unit and a power unit, comprise the basic sensor node and are depicted in Figure 2.1 redrawn from (Akyildiz et al. 2002).

2.1.2.1 Basic Components of typical sensor node

The following is a brief discussion about the basic components of a typical sensor node (see Figure 2.1).

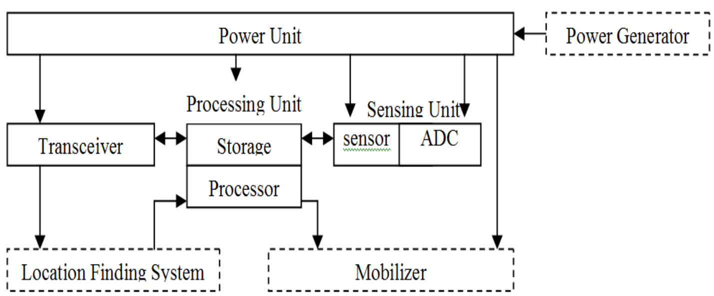


Figure 2.1 Components of a typical Wireless Sensor Node

a) *Sensing units*

The sensing unit is responsible for sensing the required physical phenomenon and makes it available to other units for further processing. The sensing unit is generally composed of two components, namely: the sensor and the analog to digital converters (ADCs). The sensor is responsible for sensing the environment for any occurrence of the required physical phenomenon, which is in the form of an analog signal. ADCs convert the analog signals produced by the sensors into digital signals, which are then fed into the processing unit. They serve as the actual interface to the physical world; they are devices that can observe or control the physical parameters of the environment. Although application dependent, a sensing unit can be composed of a single sensor or a group of sensors to sense the phenomenon like temperature or humidity. A group of sensors may include mechanical sensors, thermal sensors, acoustic sensors, optical sensors, and electromagnetic sensors (Lewis 2004).

b) The processing unit

The processing unit contains a micro-controller, small memory unit, and storage unit to store and buffer sensed data during processing (Vieira et al. 2003). It is the controller that processes all the relevant data, capable of executing an arbitrary code. The processing unit is responsible for processing the sensed information which may involve executing arbitrary codes, processing all the relevant data (checking the observed data against required information, aggregation of observed data) and other processing involved at different layers. It runs algorithms and protocols. It contains some memory to store programs and intermediate data; usually, different types of memory are used for programs and data.

c) A transceiver unit

A transceiver unit is responsible for sending /receiving the sensed data over a wireless channel. It connects the node to the network. A transceiver could use a radio frequency, infrared, or a laser for communication (Vieira et al. 2003).

d) Power units

A power unit is used to supply power with battery or energy scavenging techniques (Muruganathan et al. 2005). Considering the ad hoc nature of WSN deployment, no tethered power supply is available for sensor nodes and thus batteries are necessary to provide energy. It makes the power unit one of the most important components of a sensor node. This unit is responsible for providing the power necessary to work the senor node unit. Although, like other sub-units of sensor nodes, it is application dependent, generally sensor nodes have very low non-rechargeable battery power. For some applications, sensor nodes are provided with a power scavenging unit such as solar cells and other forms of recharging which obtain energy from the environment.

2.1.2.2 Energy consumption of sensor node

A typical sensor node consists of a processor, memory, sensing module, power supply, and transceiver. A sensor node consumes its battery in the following four operations: sensing data, receiving data, sending data, and processing data. Generally, the most energy-consuming component is the RF module that provides wireless communications. Consequently, out of all the sensor node operations, sending/receiving data consumes more energy than any other operation. The energy consumption for transmitting 1 bit of data on the wireless channel is equivalent to the energy required to execute thousands of cycles of CPU instructions (Jennifer et al. 2008). The breakup of the energy expenditure has shown that the radio component consumes much more energy than sensing and processing sub-systems. A radio transceiver of a sensor node performs two tasks: transmitting and receiving data. It has three states/modes: *active state*, *idle state* and *sleep state*. In the *active (transmit) state*, the transmit part of the transceiver is active and the antenna radiates energy. In the *active (receive) state* the receive part is active. A transceiver that is ready to receive but not receiving is said to be in an *idle state*. In the idle state, many parts of the receive circuitry are active. When a sensor node is in thelow power state and significant parts of the transceiver are switched off, it said to be in the *sleep state*. There can be different types/levels of sleep state, depending upon the amount of circuitry switched off, associated recovery times and startup energy (Wang et al. 2001). Interested readers may check out (Myers et al. 2000) for types and explanations of the different sleep states for IEEE 802.11 transceivers. In general, in the sleep state, transceiver circuitry is turned completely off and nodes wake up again with the use timers. In conclusion, the transceiver's energy consumption in different states shows that the transceiver in idle mode consumes the same amount of energy as when the in transmitting or receiving mode. Conversely, the transceiver consumes significantly less energy in the lower power sleep mode as compared to the idle, sending or receiving mode (Akyildiz et al. 2002).

Efficient use of energy in WSN communication protocols largely affects the network lifetime of the WSN and could lead to a longer network lifetime if the energy is used efficiently in the communication protocols. Hence, any MAC, network, and transport layer protocols designed for WSNs should make sure of efficient use of the RF module by minimizing MAC collision, control message overhead in routing, efficient sleep/wake scheduling and so on.

2.1.2.3 Typical sensor node hardware configuration

A typical sensor node consists of a processor, memory, sensing, power supply, and transceiver, and may even include mobility, location, and actuator module. Generally, a sensor node is composed of a low-power CPU, tiny memory (RAM/ROM), RF module, many types of sensor devices, and low-power batteries (Jennifer et al. 2008). For instance, Berkeley's MICA motes only have an 8-bit CPU, 4KB RAM, and only two AA-Alkaline batteries (Akyildiz et al. 2002).

2.2 Applications of WSN

WSNs have facilitated many application areas over the years and triggered the emergence of entirely new applications. Some of the important WSN application areas are as follows, the list is not exhaustive and only mentions important classes of applications:

2.2.1 Medicine and health care

The WSN is considered as a key area in health science applications (Alemdar and Ersoy 2010) and hence, is able to find diverse applications. Use of WSNs in medicine and health care could assist in the healthcare services from emergency response, in-hospital communications, out-of-hospital monitoring, environmental monitoring, surveillance of patients, automatic drug administration, and patient and doctor tracking systems within hospitals. Some of the successful projects using WSNs in medicine and health care are CodeBlue (Lorincz et al. 2004), Scalable Medical Alert Response Technology (SMART) (Waterman et al. 2005), MobiHealth (Konstantas et al. 2002), and Ubiquitous Monitoring Environment for Wearable, and Implantable Sensors (UbiMon) (Ng et al. 2004).

2.2.2 Automobiles

The application of wireless sensor networks includes collecting technical information during an automobile journey e.g. acceleration and fuel consumption, wrong tire pressure value, illumination failures (turn lights, brake lights, front lights, and register plate lights), tire pressure monitoring sensors, air conditioning sensors, oil sensors, and vital sign of the driver (Tavares et al. 2008).

2.2.3 Smart buildings

Smart buildings are another promising area of WSN implementation, where WSN's are used to enhance energy-efficiency and user comfort, as well as the monitoring and safety of the buildings. Sensor networks are used in multiple smart building applications that include Heating, Ventilation, and Air Conditioning systems (HVAC). It is used in lighting, shading, air quality and window control systems, switching off devices, smart metering, smart standard household applications (e.g. televisions, washing machines) security and safety (access control) (Shargal and Houseman 2009a; Shargal and Houseman 2009b; Weber and Vickery 2009).

2.2.4 Precision agriculture

The WSN has found promising applications in precision agriculture. Precision agriculture provides the means for observing, assessing and controlling agricultural practices. It covers agricultural applications from daily herd management through horticulture to field crop production, and other pre- and post-production aspects of agricultural enterprises.

WSNs used in precision agriculture include microclimates, where sensor nodes are deployed for monitoring local temperature/humidity for frost detection, and pest and mold breakout. In other areas of agriculture, WSNs are also used for irrigation management, which includes monitoring soil moisture to ensure water delivery to specific irrigation blocks, moisture monitoring for stress irrigation to improve crop quality, remote irrigation control (valve actuation, pump control), (monitoring irrigation line flow/pressure) and water delivery to correct the destination (López Riquelme et al. 2009; Pierce and Elliott 2008; Raul et al. 2008).

2.2.5 Environmental Monitoring

Environmental monitoring includes sensing the state of the environment for doing the required data collection (Hui et al. 2007). This real-time data collection about the environment helps in getting the precise knowledge of the ecosystem. Thus, assists in better decision making by providing a better understanding of current conditions such as landslides, solar and wind farms, creating landfills, wildfire detection, landfill monitoring, landslide warning systems, and global warming trends (Jue et al. 2009).

2.2.6 Military Applications

WSN applications for military encompass deployment of sensor nodes at the border, for detection of enemy troop's invasion, for battlefield surveillance, monitor vehicular traffic, track the position of the enemy or even safeguard the equipment on the side deploying sensors (Sang Hyuk et al. 2009).

A large quantity of sensor nodes is deployed over a battlefield to detect enemy intrusion instead of using landmines. When the sensors detect the event being monitored (heat, pressure, sound, light, electro-magnetic field, vibration, etc), it is reported to one of the BS, which can take appropriate action (e.g., send a message on the internet or to a satellite).

2.2.7 Disaster relief applications

Disater relief is a very important type of WSN application. It includes forest fire detection, where sensors are dropped from an airplane. They sense the temperature and send the sensed data to the fire fighter; thus, helping in controlling the fire earlier by providing a complete detail of the temperature of different areas on fire. Similar scenarios could be possible for the control of accidents in chemical factories. Some of these disaster relief applications have commonalities with military applications, where sensors should detect, for example, enemy troops rather than wildfires. In such an application, sensors should be cheap enough to be considered disposable since a large number is necessary; moreover, lifetime requirements are not particularly high.

2.2.8 Inventory Tracking

In inventory tracking WSN applications, products are equipped with a simple sensor that allows tracking of these goods. It helps during transportation of goods and facilitates inventory tracking in stores or warehouses. In these applications, sensor nodes are generally passive and the reader of the data is placed at the checkpoints/exits (Estrin et al. 1999 ; Kahn et al. 1999).

2.2.9 Wild Life Tracking

The WSN application for wild life tracking (Weber and Vickery 2009) has enabled farmers to observe the behavior of cattle which includes grazing habits, herd behavior and interaction with the surrounding environment (Wark et al. 2007). The information also helps the farmers to understand the state of the pasture and use the resources to their optimal potential. For these applications, generally sensors are attached to the cattle collars (Wark et al. 2007). The sensors contain GPS and use multi-hop communication to report the data to the BS. Collected data can be used to ascertain the cattle's individual and herd behavior, which could be modeled and then generalized to develop more general models. Ultimately, using all this information, farmers can manage the environmental resources to their optimum and accordingly plan grazing areas to prevent environmental problems such as overgrazing and land erosion.

2.2.10 Transport and logistics

WSN potential to increase the transport efficiency has been exploited in many transport and logistics applications. The Intelligent Transport System (ITS) has been developed using WSNs. It includes a WSN used for better tracking of goods and vehicles which might result in lower level of inventories and thus energy savings from less inventory infrastructure as well as a reduced need for transportation (Atkinson and D. October 2008). ITS are able to detect public transport and can be used to keep their schedule accordingly (Veloso et al. 2009). Furthermore, sensors and sensor networks are used for road traffic monitoring systems to measure the intensity, congestion of traffic and information; this is then used to dynamically control traffic lights according to traffic inflow. In addition, sensors are used for motorway tolling purposes where they detect vehicle RFID tags and retrieve the required information (Veloso et al. 2009). Sensors also monitor the state of physical infrastructures such as bridges by detecting “vibrations and displacements”(Veloso et al. 2009).

2.2.11 Urban terrain tracking and civil structure monitoring

WSNs are used to monitor the structural health of civil structures. It includes monitoring the strain of the bridge structure when trains are crossing the bridge. Civil Structure monitoring systems detect, localize, and estimate the extent of damage.

2.2.12 Security Applications

Sensory networks can be used in cases of unauthorized access to buildings or misuse of facilities provided. (Joohwan et al. 2008) discussed in detail, home security systems based on sensor networks and a robot.

2.2.13 Other Applications

Some of the other emerging WSN applications include resource explorations, such as mining and mineral analysis. Health applications, which involve tracking patients, monitoring drug administrations in hospitals, healthcare, smart household electronics, smart home/office environments, asset and warehouse management, automotive industry, home and building automation, industrial process control, military battlefield awareness, security, and surveillance.

2.3 Wireless Sensor Network Challenges

Unlike traditional networks, a WSN has its own design and resource constraints. These constraints impose some unique challenges for protocol design in WSNs. Some of these challenges involved in WSN protocol design are as follows:

2.3.1 Limited Physical Resources

Typically, sensor nodes are equipped with low processing power, low sensing power, less memory, and short-range communication, and most importantly have very limited battery power (Edgar and Callaway 2003). Randomly deployed in difficult terrain makes it almost impossible to recharge or replace the dead battery. Therefore, battery power in WSNs is considered a scarce resource and should be used efficiently. The breakdown of the sensor’s energy consumption shows that, nodes consume energy in sensing data, receiving data, sending data, and processing data. The most energy consuming component is the RF module that provides wireless communications. Out of all the operations of a sensor node, sending/receiving data consumes more energy than other operations. The efficient use of energy in WSN communication/routing protocols will affect the network lifetime, i.e. it could lead to longer network lifetime if the energy is used efficiently. Therefore, WSN protocol design should consider economical and efficient use of battery power and be able to cope with random deployment. In addition, the other limited resources of sensor nodes should also be considered in WSN communications, which includes low processing power, low sensing power, less memory, and short-range communication. All these constraints make protocol design in WSNs a challenging task. So the protocol should be simple and light-weight.

2.3.2 No Infrastructure

Typically, sensor nodes are randomly deployed and have little or no infrastructure. Additionally, a WSN involves a number of sensor nodes (a few tens to thousands) deployed in difficult terrain to sense some data about the environment. Thus, WSN protocols should be able to operate in an ad hoc manner, using distributed algorithms that don't require global knowledge or centralized coordination.

2.3.3 Fault Tolerance

In WSNs, sensor nodes are generally prone to failures. Failure involves physical damage in a hostile environment as well as dead nodes due to battery depletion. These failures could occur to different nodes during the course of the WSN lifetime, which results in dynamic changes in the topology. WSN protocol design should be responsive to these dynamic topological changes due to node failure and able to accommodate these changes speedily by doing route repair or finding alternate routes. Thus, considering the failure prone nature of the WSN, fault tolerance must be an essential feature of WSN protocols, rather than additional functionality.

2.3.4 Scalability

WSN applications involve deployment of many sensor nodes varying from a few tens to thousands. The scale of nodes in WSNs is considerably higher than traditional networks. Hence, protocols designed for WSNs, should specifically consider this feature of WSNs and should be scalable to work with large networks.

2.3.5 Network Lifetime

In most of the WSN deployment, the sensor node has limited battery power and it is not possible to replace or recharge the battery because of difficult terrain or a hostile environment. It makes battery power in WSNs the most scarce resource and extra care should be taken to use it efficiently.

WSN protocol design must consider the least energy consumption in their communication activities, especially in sending/receiving data. Nodes have to rely on a limited supply of energy (using batteries) and must operate with that battery as long as possible. The efficient use of energy in WSN communication protocols will influence the network lifetime and will lead to a longer network lifetime if the energy is used efficiently. Hence, the lifetime of a WSN becomes a very important design issue in devising any WSN protocol. It must ensure efficient use of onboard energy to extend the network lifetime.

2.3.6 Quality of Service

Some of the WSN applications involve real time reporting of a sensed event. Delay is not acceptable for such delay-sensitive applications, which include military surveillance, tsunami alarm systems, smart hospitals, seismic detection, biomedical health monitoring, hazardous environment sensing, fire detection, intrusion detection, disaster monitoring, and real-time control. In such applications, in addition to energy efficiency; latency, throughput and delay also become issues of primary concern. Such performance metrics are usually referred to as the quality of service (QoS) of the communication network. Thus, routing multimedia data needs careful handling to have energy efficiency as well as a timely delivery within an acceptable range. Thus, the WSN protocol must be able to meet such QoS requirements of the applications.

2.3.7 Maintainability and Self Healing

During the course of WSN deployment, sensor nodes may fail because of physical damage or energy depletion. It can result in changing topology and routing path breakage; thus, considering the inherent fault prone nature of WSNs, any protocol design must consider a self healing feature, so that, the protocol will be able to maintain a reasonable QoS with all these failures. Furthermore, it should be able to do self organization and be able to self heal without intervention by humans.

2.3.8 Remotely Managed

In WSNs, sensor nodes are close to the environment; after deployment, they are remotely managed and have a minimum or no interaction with humans. In this scenario, nodes need to process information as well as react flexibly to the dynamic changes in tasks. Thus, any protocol design for WSNs must consider flexibility and maintainability to assist remote management by providing fully distributed and self configurable solutions. WSN protocols should be capable of adapting to the changes in the environment and network itself; by self monitoring its own health and interacting with remote maintenance mechanisms, it ensures its extended operation at a required quality level.

2.3.9 Unreliable Communication

In WSNs, wireless links in low-power sensor networks are irregular and unreliable. If not addressed, transmitting data unreliably may lead to many problems like uncertainty, performance decline and so on. Thus, the design of communication stacks for WSNs is proposed, which must take into account these radio layer realities and ensure reliable transmitting over unreliable links.

2.3.10 Collisions and Latency

Packet Collision occurs when two or more close nodes attempt to transmit a packet at the same time. This can result in packet loss and impede network performance. Considering that dense WSNs deal with hundreds or thousands of nodes, it becomes even worse, due to burst-traffic and congestion around sinks. Consequences of packet collisions are serious to WSNs, as they cause packet loss, waste resources in wireless networks and increase latency which can cause the loss of critical control information from base stations as well as applications failing. Thus, WSN protocol design should take into account collision and latency characteristics of WSNs.

2.4 QoS Based Energy Efficient Routing in the WSN

In this section, related work relevant to this research work is presented. A systematic review of the related work and issues in the existing QoS based energy efficient approaches is presented. Related work on QoS based routing in WSNs is presented in the perspective of the clustering approach, sleep/wake scheduling, and mobile sink approach. This chapter discusses in detail the clustering approach and the important review of the literature under two major clustering categories: variable convergence time and constant convergence time clustering algorithms. Next to follow is a detailed discussion of different sleep/wake scheduling algorithms, including the important Sleep/wake scheduling scheme under three major categories: On-demand Schemes, Scheduled rendezvous schemes and Asynchronous Schemes. Lastly, the chapter presents a detailed discussion of related work on the mobile sink approach including three main categories: Mechanisms using mobile sinks (Algorithms with pre-determined sink mobility path, Algorithms with autonomous sink movement), Mechanisms using mobile sensors redeployment and Mechanisms using mobile relays are discussed. At the end of each of these three related work sections, potential problem statements are briefly discussed and supported, and proposed solutions are overviewed.

2.4.1 Clustering Approach

Routing in WSNs can be divided into three categories (Al-Karaki and Kamal 2004): flat-based routing, hierarchical-based routing, and location-based routing. In flat-based routing, as the name indicates, nodes have the same roles and share equal responsibilities for forwarding/routing to the data. In hierarchical-based routing, however, roles of the nodes vary depending upon on which hierarchy the nodes lie in.

Decisions about packet forwarding by a sensor node depend on the level of the hierarchy at which it lies and the role assigned to it. The roles of a particular sensor node may differ from time to time. In location-based routing, the location of sensor nodes plays a key role, and the decision of routing is influenced by the position of the sensor nodes.

Of the many data routing approaches available, the hierarchical-based routing (cluster-based data gathering) approach has been widely pursued by the research community as an effective architecture for routing in WSNs and is the focus of this section. In a typical clustering protocol (Heinzelman et al. 2000), the whole WSN is divided into small group of nodes known as clusters (see Figure 2.2). In each cluster, one node is elected as a Cluster Head (CH). Each CH is responsible for aggregating the sensed data from its cluster member node(s) and propagating it to the next CH or to the base station. As the CH has to relay the data of all member sensor nodes, it will soon deplete its energy, if repeatedly selected as CH. Hence, the phenomenon of CH selection is periodically divided into fixed time intervals, called rounds. In each round, a new CH is selected randomly or based on certain parameters like residual energy, distance from the base station, connected nodes or topology. Any energy-aware clustering tends to have a balanced use of sensor node energy and thereby increases the overall lifetime of the sensor network. During the clustering process, for efficient and effective clustering, it is necessary to take into account other design aspects, such as cluster size and form, CH selection criteria, control of inter-cluster and intra-cluster collisions, and energy saving issues. Efficient design of the clustering approach is one of the most important issues for correct functioning of the network, as benefits of clustering can only be achieved with effective hierarchical communications.

In (Abbasi et al. 2007), a comprehensive survey of different clustering algorithms for WSNs is given. It grouped clustering schemes into different categories based on their objectives, the desired cluster properties, and the clustering process. It compared and summarized features of clustering algorithms in terms of some clustering parameters, such as convergence time, node mobility, cluster overlapping, location awareness, energy efficiency, failure recovery, balanced clustering, and cluster stability. Furthermore, the section explained the effect of the network model on different clustering protocols, and summarized a number of algorithms, stating their strength and limitations based on the clustering approaches.

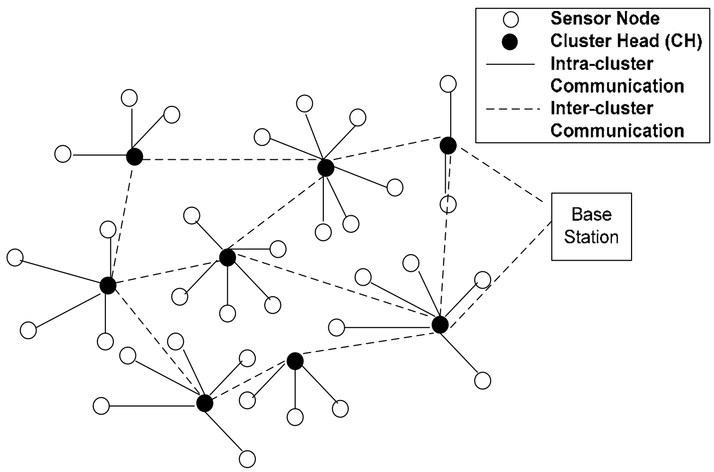


Figure 2.2 A typical Clustering based data gathering in a Wireless Sensor Network

The rest of this section outlines the objectives of a typical clustering protocol and discusses in detail related work on the clustering approach. The review of the literature on the clustering protocols presents the classification of clustering protocols under its two major categories: variable convergence time algorithms and constant convergence time algorithms. At the end this section, the potential problem statement of the clustering approach is briefly discussed and the proposed solution is overviewed.

2.4.1.1 Clustering objectives

The clustering objective refers to the main motivation behind clustering and is generally application dependent. With some applications, the data latency is the primary objective and others consider energy efficiency as the primary objective. Abbasi et al. (2007) summarized the popular clustering objective. Following is the brief discussion of these clustering objectives:

a) Load balancing

Load balancing refers to the number of nodes in a particular cluster. In clustering, it is encouraged to have equal sized clusters. It results in exhausting the energy of all the clusters in a balance manner and ultimately increasing the network lifetime. Furthermore, equal sized clusters help in data aggregation as the data from adjacent clusters are generated at the same time, which helps in making a combined data report.

b) Fault-tolerance

The WSN is inherently failure prone due to the limited battery of the sensor node and random hostile environment deployment. Thus, fault tolerance should be part and parcel for any clustering protocol. Fault tolerance can be provided in many ways for instance by doing re-clustering, selecting a backup cluster head and rotating the role of the cluster head. The choice of the appropriate fault tolerance strategy depends upon the application.

c) Increased connectivity and reduced delay

CH connectivity is the important perspective of any clustering protocol especially when the CH is picked up from the network, as it ensures the availability of a path from the CH to the BS. For reducing delay, intra cluster connectivity becomes the primary objective.

d) Minimal cluster count

A minimal cluster count is an important objective especially in scenarios where some resource rich nodes act as the CH. In those cases, the objective is to deploy the least number of these resource rich nodes to minimize cost. Another reason for reducing the number of resource rich CHs, is that they are visible, which is not viable in some WSN applications e.g. military applications.

e) Maximal network longevity

Generally sensor nodes in WSN have very limited energy and network lifetime is considered as an important objective in clustering. Thus, any clustering protocol should be able to improve the energy efficiency and increase the network lifetime.

2.4.1.2 Clustering Protocols

Considering the large scale deployment of sensor nodes spanning to hundreds or thousands, over the years, clustering has proven to be a scalable and an effective way for managing large WSNs. As clustering provides a scalable solution for large WSNs, Abbasi et al. (2007) surveyed clustering algorithms by categorizing them based on convergence time into two classes: variable convergence time algorithms and constant convergence time algorithms. The following is the review of literature about important clustering algorithms under the two sub-sections: variable convergence time algorithms and constant convergence time algorithms.

a) Variable convergence time algorithms

Variable convergence time algorithms (Abbasi et al. 2007) are the clustering algorithms that do not converge completely within a fixed number of iterations. They are generally useful for small networks and provide better control over cluster properties than constant time algorithms. The following is the review of literature for variable convergence time algorithms.

i) Linked cluster algorithm (LCA)

LCA (Baker and Ephremides 1981; Baker et al. 1984) is one of the very first clustering protocols. It is a distributed algorithm that forms an energy efficient topology to handle mobility of nodes. It forms the cluster where each node is directly connected with the CH. To do the clustering, in the LCA, each node broadcasts it's ID and listens to the communication of other nodes in its neighborhood. After some time, every node will broadcast the list of nodes from which it heard a communication; in this way, each node knows its first and second hop neighbors. Node *A* becomes a CH if it has the highest ID among all of its neighbors or there exists at least one neighbor node *B*, where *A* has a higher ID than all of *B's* first hop neighbors.

ii) Adaptive clustering

In adaptive clustering (Lin and Gerla 1997), to minimize delay for multimedia applications, MANET is clustered. It uses a single hop intra cluster communication. As the large cluster size decreases the data delivery delay, it decreases spatial reuse of the channel and vice versa. Thus, the cluster size is controlled by balancing the tradeoff between spatial reuse of the channel and data delivery delay.

iii) Random competition based clustering (RCC)

RCC (Kaixin and Gerla 2002) is proposed for the MANET with the objective of achieving cluster stability to accommodate mobile nodes. In RCC, CH selection is very simple and is based on First Come First Serve (FCFS) basis, where any node could become the CH of the nodes lying in its communication range if it is able to declare itself the CH before any other node does. After receiving the advertisement from the first node, other nodes join this CH as member nodes. The nodes do the re-clustering after a periodic interval to maintain clusters. To avoid the conflict situation where two nodes send the CH advertisement at the same time unaware of each others' advertisement, each node backs off for a random time delay before sending the CH advertisement. If even in the case of the random time back off, a conflict arises then the node with the lower node ID is selected as the CH.

iv) CLUBS

In CLUBS (Nagpal and Coore 1998), to select the CH, each node selects an integer random number from a specified fixed range. The node starts decrementing that number continuously until the number reaches zero. If it receives a CH advertisement from any other node it will join that CH as a follower and will not compete for the CH position. Otherwise, it will declare itself as the CH by sending the CH advertisement. CLUBS allows overlapping clusters and a node could become a follower of more than one CH. If during the CH advertisement message, a node observes a collision, it will conceive that some other node is also sending a CH advertisement message. Thus,a it will stop its transmission and become part of any other cluster as a follower. The process continues until all the nodes in the network become part of any cluster either as the CH or as a follower.

v) Hierarchical control clustering

In (Banerjee and Khuller 2001) a multi tier hierarchical clustering protocol is proposed. Any node in the network can initiate the cluster formation process; if two nodes have a conflict then the node with l or the lower node ID is given preference. Clustering is done in two phases: *tree discovery* and *cluster formation*. *Tree discovery* is done by breadth; the first search tree is formed, with the root being the initiator node. *Cluster formation* is done at the sub trees of a node.

vi) GS^3: scalable self-configuration and self-healing in wireless networks

In GS^3 (Zhang and Arora 2003), self configuring of the wireless network is done in the form of a cellular hexagon structure. In GS^3, two types of nodes exist: a *big node* and a *small node*. Big nodes are responsible for starting the cluster formation process, additionally, they act as mediators between small nodes and the Internet for doing the interfacing. In each virtual cellular hexagon, a big node starts the clustering process by selecting the cell heads in their neighborhood and unselected nodes in the cell switch to the role of cell member. After selection of the cell head, all the cell heads relocate themselves at the centre of their particular cell.

vii) Energy Efficient Hierarchical Clustering (EEHC)

EEHC (Seema and Coyle 2003) is a distributed clustering protocol with an objective to maximize the network lifetime. In EEHC, the clustering technique is based on two stages: *initial* and *extended*. In the *initial stage*, each node sends the CHs advertisement message with probability *p*. Nodes which receive that CH advertisement, either directly or through forwarding, become part of this cluster. These CHs are referred as *volunteer CHs.* If a node does not receive any CH advertisement from any volunteer cluster, it will become the CH itself and is referred to as a *forced CH*. In the *extended stage*, the same process of clustering is repeated for the CH to form different tiers of the clustering hierarchy. The CHs at level-1 pass the aggregated data to the CH at level-2 and so on till the data reach the BS.

viii) Energy-Efficient Multilevel Clustering (EEMC)

The Energy-Efficient Multilevel Clustering algorithm (EEMC) (Jin et al. 2008) organizes nodes into a hierarchy of clusters with the objective of minimizing the total energy spent in the network. The optimal expected number of CHs is calculated to achieve minimum energy consumption and minimum latency in a multilevel clustering scheme. In addition, the optimal number of levels is calculated asymptotically. For given *N* nodes, EEMC terminates in *O (log (logN))* iterations.

ix) Coverage preserving clustering protocol (CPCP)

In CPCP (Soro and Heinzelman 2009), the CH election problem was considered for applications that require full coverage. In such coverage-preserving applications, decisions for selecting CHs/data routers/active nodes are made both on the basis of the node's residual energy and its redundancy in the coverage. For that, a coverage preserving clustering protocol (CPCP) was proposed. In CPCP, clustering was based on a coverage-aware cost metric. For selecting CHs, data routers and active nodes favor nodes that are present in a densely deployed region.

b) Constant convergence time algorithms

Constant convergence time algorithms converge completely in a fixed number of iterations, irrespective of network size. In these algorithms, the cluster membership decisions are taken in a distributed manner whereby a node uses localized information about its neighbor to decide on the CH. The following is the review of literature for constant convergence time algorithms.

a) Low Energy Adaptive Clustering Hierarchy (LEACH*)*

Heinzelman et al. (2000) presented a clustering scheme called the low-energy adaptive clustering hierarchy (LEACH) for energy-efficient data gathering. In LEACH, periodically, every sensor node with some probability elects itself as the CH and broadcasts its decision within its neighborhood. For any round, any sensor node becomes a CH if the number is less than the following threshold:

$$T(i) = \begin{cases} \dfrac{p}{1 - p\left(r mod \dfrac{1}{p}\right)} & if\ i\ \in G \\ 0 & otherwise \end{cases} \tag{2.1}$$

where *p* refers to the percentage of CH nodes in the sensor population, *r* is the current round number, and *G* refers to the set of nodes that were not CHs in the last *1/p* rounds. The remaining nodes, on receiving the broadcast message, based on the received signal strength, join one CH. As the CH node consumes more energy than the other normal cluster member nodes, and to balance the energy consumption for its role as CH, it is periodically rotated among the nodes. It supports one hop communication between sensor nodes and a sink/CH, hence is not adequate for large networks. However, it provides a good model to reduce information overload and presenting reliable data to the end user. Furthermore, it works in a purely distributed manner, hence it requires no global knowledge of the network. LEACH exchanges messages to do dynamic clustering, which is overhead and consumes energy. This overhead can affect the energy gain achieved by clustering.

i) Fast Local Clustering service (FLOC)

FLOC Demirbas et al. (2004) is a distributed clustering algorithm that avoids overlapping clustering by having equal sized clusters. In FLOC, a radio model divides the node into two types based on their proximity to the CH, namely: i*nner node (i-band)* and *outer node (o-band). i-band nodes* suffers no or minimum interference while communicating to the CH as they are near to the CH, whereas, *o-band* nodes suffer high interference or may be lost messages while communicating to the CH as they are away from the CH. To have efficient intra cluster communication, FLOC favors i-band membership. To do the clustering, each node stays idle for in random time, to receive advertisement from any CH. If during that time, it receives no CH advertisement from any CH, it becomes a candidate CH and announces itself as the CH by broadcasting the CH advertisement message. When this CH advertisement message is received by some *node i*, which is already an i-band member of some *cluster I*, it will respond back to this CH advertisement by sending its cluster membership, to let the candidate CH node know about its cluster membership. As an o-band member, the candidate CH then joins that *cluster I*. If the candidate CH does not receive any conflicting massage from any of the neighboring nodes, it becomes the CH and sends an invitation to the neighboring nodes to join its cluster. If an idle node did not receive any invitation message from a closer CH, it will join some other CH with o-band membership. Later, it can change its decision if it receives a message from any closer CH.

ii) Algorithm for Cluster Establishment (ACE)

In ACE (Karl et al. 2004), a node before declaring itself as the CH, assess its potential as a CH and will defer its declaration as CH if currently it is not the best CH. *Spawning* a new cluster and *migrating* the existing one are the two important features of ACE. In *spawning*, a CH node broadcasts the message to its neighbor, the neighbor node, listening to that message, becomes a follower of that CH. A node could follow more than one CH. In *migration*, each CH periodically verifies the ability of its neighbor to be the CH and steps down if it finds a neighbor node which is more fit for being the CH. For verification, the CH checks the followers of its neighbor nodes and its overlap with the current CH. The neighbor node having more followers and the least overlap with the current cluster is selected as new the CH.

b) Hybrid Energy-Efficient Distributed Clustering (HEED*)*

HEED (Younis and Fahmy 2004) is a distributed clustering protocol where the CH is selected based on communication cost as well as energy. HEED is divided into three phases, namely: *Initialization*, *repetition* and *finalization* phases. In the *initialization phase*, to limit initial CH announcements, the initial percentage of the CH is set by using the percentage value C_{prob}. A node becomes the CH according to the following probability formula.

$$CH_{prob} = \frac{C_{prob} * E_{residual}}{E_{max}} \tag{2.2}$$

where

$E_{residual}$ is the residual energy of the sensor node.

E_{max} is the maximum energy, i.e., fully charged battery.

In the *repetition phase*, after many iterations, nodes select the CH to which they can communicate with minimum transmission power. If a node did not hear from any CH, it will make itself as the CH and broadcast the CH advertisement message to its neighboring nodes. Each sensor enters into the next repetition by increasing its CH_{prob} value to double. This phase is ended when the CH_{prob} reaches 1. During the course of the CH selection, each node may have two statuses regarding being the CH: Tentative status (if its CH_{prob} is less than 1) and final status (if its CH_{prob} reaches 1). In the *finalization phase*, the final decision regarding CH selection is taken. The sensor node either elects itself as the CH or elects the least cost CH from its neighbors.

iii) Distributed Weight-Based Energy-Efficient Hierarchical Clustering (DWEHC)

In DWEHC (Ding et al. 2005), to do the clustering and elect the CH, each sensor calculates its weight and in a particular neighborhood the, node with the maximum weight is elected as the CH. Weight is a function of the node residual energy and its distance from its neighbors. Selection of the CH on the maximum weight is the first level membership where the node directly communicate with the CH. Afterwards, the member node keeps on searching for the best CH (in terms of communication cost), by checking its non cluster neighbors' cost. Member sensor nodes accordingly access whether to stay within the current CH as first level membership or to change to the other the CH in the second level membership. The process continues until nodes find the best energy efficient path to reach the CH. To limit the number of levels in intra cluster communication, each cluster is assigned a range for the possible number of member nodes..

iv) MOCA

In the Multi-hop Overlapping Clustering Algorithm (MOCA) (Youssef et al. 2006), the benefit of having an overlapping cluster is discussed and the multi-hop overlapping clustering algorithm is proposed. Each node in the network could become a CH with probability p. The node sends the CH advertisement message among its neighbors, all the *k-hops* neighbor of the node receive that CH advertisement. In respond to the CH advertisement message, the node sends a join request (including ID of all CHs it heard from) to all CHs from whom it receives a CH advertisement message.

v) Attribute-based clustering

In attribute based clustering (Ke et al. 2005), to achieve energy efficient data dissemination, the idea of doing clustering based on the queries and attributes of the data is coined. The BS starts the clustering process by sending a request message to the sensor node to form clusters. The node which receives that request message, based on their energy decides whether to declare itself as the CH or not. The node will wait for a random time if it decides to declare itself as the CH. During that time if it receives a CH advertisement from any other node, it will relinquish its CH advertisement process, and resend the received CH advertisement message by incrementing the hop count value. In the end, n the ode will join the CH with the minimum hop count. Otherwise, it will declare itself as the CH by sending the CH advertisement to its neighboring nodes. To balance energy, the CH role is rotated among the cluster members. Fault tolerance is ensured by sending periodic heart beat messages from the CH to its members. When no heart messages are received by the member nodes they assume the CH is dysfunctional.

Through the extensive literature review, it is concluded that a number of papers can be found discussing energy efficient clustering in WSNs. However, still very little work can be found on QoS based energy efficient protocols. Therefore, QoS based energy-efficient clustering in WSNs is a hot research area, and an opportunity for a better solution is wide open.

Using this approach of multi-hop CH communication, the sensed data from a member node in a cluster is forwarded via a number of CHs, and finally it reaches the intended BS. However, the potential drawback of this approach is that nodes near to the BS will deplete their energy faster, as they are not only forwarding their own data but also data of many other nodes away from the BS. This problem is termed as a *hot spot* or *energy-hole* problem. It can significantly limit the lifetime of the WSN. The other problem is that in most of the currently available algorithms for CH selection, the optimization is done on a single parameter, which can be residual energy, distance from the BS, connected nodes, or coverage lifetime, and so forth. The optimization of this single parameter, however, has resulted in performance degradation for the other parameters. For instance, if residual energy is considered as the optimization parameter, it may optimize the lifetime of the network, but it will trade-off the coverage of the network and vice versa.

In this thesis, QoS based energy-efficient routing in WSNs is considered. For this reason, to address these two problems, QoS based Energy Efficient Clustering

(QoSEC) Protocol for the WSN is proposed. The main objective of QoSEC is to cluster the sensor nodes in a network in such a way that the energy of the sensor nodes is used in a balanced manner to increase the overall network/coverage lifetime of the network and improve the QoS metrics such as delay, throughput, and packet loss ratio. The detailed operations of QoSEC and these two algorithms are elaborated later in methodology section 3.1.

2.4.2 Sleep/wake scheduling schemes

Generally, WSNs operate for long time in idle mode and only occasionally send its data. Energy consumption of listening to the idle channel is equivalent to its energy consumption when sending or receiving, and much larger than the energy consumption of the sleep mode (Akyildiz et al. 2006). This fact is exploited by sleep/wake scheduling techniques and effort is made to reduce this energy wastage in the idle mode. The mechanism of alternatively switching between sleep and wakeup mode to avoid energy wastage is termed as *sleep/wake scheduling* or *duty cycling scheme*. Sleep/wake scheduling is regarded as the most effective mechanism for energy conservation and significantly prolongs the network lifetime in WSNs. To achieve minimum energy consumption, the transceiver should mostly be in the sleep state and be activated (wakeup state) only when required. Clearly, the best scenario is that a sensor node wakes up once and finishes all its operations continuously, but practically it is not possible to have and such an ideal coordinated and synchronized environment. Thus, the main objective of sleep/wake scheduling is to put the transceiver in a sleep state during idle periods (when there are no events). Significant energy savings can be accomplished by designing sleep/wake schemes that can let the sensor's transceivers remain in sleep mode for as long as possible, and at the same time retain network connectivity and coverage. In sleep/wake scheduling, a complete cycle consists of sleep state and wakeup state, and is termed as a *frame*. The ratio of wake intervals to the total frame length time is termed as a *duty cycle* and to avoid energy wastage the duty cycle should be kept small.

In this section, the second perspective to provide QoS based Energy Efficient routing is discussed; this is the sleep/wake approach. The rest of this section gives a detailed discussion of different sleep/wake scheduling algorithms, including the important sleep/wake scheduling scheme in its three major categories (Anastasi et al. 2009): On-demand Schemes, Scheduled rendezvous schemes and Asynchronous Schemes. At the end of this section, the potential problem statement in the sleep/wake scheduling approach is briefly discussed and the proposed solution is overviewed.

Anastasi et al. (2009) gives a comprehensive survey of sleep/wakeup scheduling schemes in WSNs. They divided the sleep/wakeup protocols into three main classes (Anastasi et al. 2009; Armstrong): on-demand, scheduled rendezvous, and asynchronous schemes. The following is the review of literature of these three sleep/wake protocols categories.

2.4.2.1 On-demand Schemes

On-demand sleep/wake protocols are based on the very simple idea, the node will wake only when any other node wants to communicate with it. It reduces the energy consumption and makes this strategy adequate for application with a low duty cycle. These applications include different event driven applications, namely: military surveillance, tsunami alarms, seismic detection, hazardous environment sensing, fire detection, intrusion detection, disaster monitoring, and real-time control. In these applications, sensor nodes constantly sense the environment (monitoring state) and communicate when a required event occurs (transmitting state). In this scenario, the on-demand scheme objective is to reduce the energy consumption during monitoring as well as to ensure minimum latency to transit from the monitoring into the transmitting state (Anastasi et al. 2009).

The main issue in these protocols is how the receiving node is informed that there is some node that wants to communicate with it. In these schemes, generally there is more than one radio: low power radio and high power radio. The low power radio is used to send/receive control message signaling and the high power is used to do the actual data communication. The downside of on-demand protocols is the additional cost for the second radio. A mismatch between the communication ranges of the two radios is another drawback associated with these schemes. Following is the review of literature of some important on-demand sleep/wake scheduling schemes (Anastasi et al. 2009).

a) STEM

Sparse Topology and Energy Management (STEM) (Schurgers et al. 2002b) uses two different radios: one for a wakeup signal and the other for data packet communication. Each node periodically turns on its wakeup radio. When the source node has to communicate with some destination node, then the source node will send periodic beacons on the wakeup channel of the destination node. When the destination node receives the wakeup signal it will turn on its data radio and respond with the wakeup acknowledgment to the source node. Whenever a collision occurs at the wakeup channel, all the nodes which hear this collision will turn on their radio channel, but send no wakeup acknowledgment. The source node keeps on sending the beacons until the maximum limit is reached or until the wakeup acknowledgment has been received. STEM-T (Schurgers et al. 2002a) is the proposed extension of STEM-B. STEM-T instead of the beacon using a wakeup tone, all the neighbor nodes of the source are awakened.

b) Pipelined Tone Wakeup (PTW)

The Pipelined Tone Wakeup (PTW) scheme (Xue and Vaidya 2004) also uses two different channels for sending/receiving wakeup signals and others for sending/receiving data. In PTW, responsibility of detecting the tone is given to the sender. The receiver wakes up periodically and the sender sends the wakeup tone when the event is detected. Furthermore, to reduce the wakeup latency, the wakeup signal is pipelined with the data propagation. Hence, the wakeup latency and the overall message latency is reduced.

c) Radio-Triggered Power Management scheme

In the Radio-Triggered Power Management scheme (Lin and John 2005), energy contained in the wakeup signal is used to activate the sensor node, similar to what is done in active Radio Frequency Identification (active RFID). To capture energy in the wakeup signal, a special radio triggered circuit is used in the sensor node hardware. The radio triggered approach is totally different than the standby radio, where the radio consumes energy while listening to the wakeup signal but in the radio triggered approach, the radio is powered by the wakeup signal. The main constraint of the radio triggered approach is that it is limited to a short distance, for instance, the maximum achievable distance is 3 meters. More complex architecture can considerably increase the distance, but the cost factor makes them infeasible.

2.4.2.2 Scheduled rendezvous schemes

An alternative solution consists in using a scheduled rendezvous approach. The basic idea behind the scheduled rendezvous schemes is that each node should wake up at the same time as its neighbors. Typically, nodes wake up according to a wakeup schedule, and remain active for a short time interval to communicate with their neighbors. Then, they go to sleep until the next rendezvous time.

In scheduled rendezvous schemes, all the nodes are required to wake up at the same time. For that, nodes wakeup after periodic intervals according to a wakeup schedule, and check any communication taking place. It remains active for a short time interval to make communication with its neighbors and go to the sleep state again until the arrival of the next time interval. The advantage of this approach is that, when one node wakes up, all its neighbors are also certain to wake up in the same time interval, which can ease the communication. The main limitation of this approach is that, it requires synchronization among the neighboring sensor nodes, which comes with the overhead of control message communication.

The following is the review of literature of some popular scheduled rendezvous schemes.

a) Fully Synchronized Pattern

In the fully synchronized Pattern schemes (Abtin et al. 2006) all the nodes are fully synchronized. All the nodes in the network wakeup periodically at the same time, remain awake for a fixed time, and then go into the sleep state until the arrival of next wake up interval. The potential drawback of this approach is that, all the nodes after a long sleep interval, wakeup at the same time and try to submit simultaneously, which results in a lot of collisions. Furthermore, as the size of the sleep/wake intervals are fixed, and do not adapt to the variable traffic pattern, hence this approach is inflexible. A MAC protocol such as S-MAC (Wei et al. 2002) and T-MAC (Tijs van and Koen 2003) use this type of sleep/wake scheduling because of its simplicity. Tijs van and Koen (2003) improved this approach by turning off the radio of the sensor node when there is no activity for at least one timeout value.

b) Sensor-MAC (S-MAC)

Sensor MAC (S-MAC) (Wei et al. 2002) is an example of a distributed coarse-grain passive protocol and is one of the first MAC layers designed to reduce power consumption in WSNs. In S-MAC, nodes are randomly turned on and turned off to save energy consumption. Traffic destined for these randomly sleeping nodes is temporarily stored by the neighbors, who are active. Sleeping nodes after periodic intervals wakeup and receive/send their data from/to their neighbors. Afterwards, the sensor nodes will go to sleep mode again and wakeup in the next cycle. Three power conservation schemes were used in S-MAC: *Fixed Sleep Scheduling, Overhearing avoidance using sleeping* and *message passing scheme*. To do sleep scheduling, first synchronization is done, to have harmony in the periodic sleeping of sensor nodes and to avoid over hearing problems. Synchronization involves additional complexity, time and power overhead which have to be considered in devising any sleep scheduling mechanism. Though a fair amount of energy is saved by the sleep scheduling algorithm, it also introduces delay which can be very critical in some applications.

c) Staggered Wakeup Pattern

In staggered wake up patterns (Karl and Willig 2005), nodes in a network that lies at different levels of data gathering, wakeup at different times. To have smooth communication, the wake intervals of the nodes which lie at different level should overlap. It will make it possible for that node's time interval when it sends/receives data from its child to be adjacent to the time interval when it sends/receives data from its parent. It will save the energy consumed in switching between the sleep and wake state. Staggered wakeup patterns can have two variations: *backward staggered pattern* and *forward staggered pattern*. In the backward staggered pattern (Karl and Willig 2005), packet latency is optimized in the backward direction from the leaf to the root (sink) node. In the forward staggered pattern (Karl and Willig 2005), packet latency is optimized in a forward direction from the root (sink) to the leaf node. The forward staggered pattern is not used practically as data flow in WSNs is from node to the sink. The Staggered Wakeup Pattern comes with different advantages, which include: 1) In the staggered wakeup pattern, the nodes, at different levels wakeup at different times, thus, at a particular instance of time only a small number of nodes are taking part in communication. Thus, it will suffer least collisions. 2) In the staggered wakeup pattern, nodes at different levels wakeup at different times, thus, they provide better energy savings than a full synchronized pattern. 3) It is suitable for data aggregation as parent nodes need to receive data from all its child before forwarding it to its own parent. It allows parent nodes to filter the data and do the data aggregation, after collecting data from its children nodes. Disadvantages of the staggered approach are as follows: 1) As the nodes at the same level wakeup at the same time, thus, the staggered approach still suffers from collisions. 2) It is inflexible due to the fixed duration of the sleep/wake schedule.

d) An adaptive and low latency staggered scheme

The Adaptive and low latency staggered scheme (Anastasi et al. 2009) keeps the wake interval minimum according to the current network activity. It saves the energy consumption as well as improves the delay per packet. Furthermore, using different lengths of wake intervals for nodes at the same level reduces collisions.

e) Power Scheduling (FPS)

Barbara et al. (2004) proposed a TDMA based scheme called Flexible Power Scheduling (FPS) is divided into slots, which are arranged to form a periodic cycle. Each node makes a schedule of different operations it needs to perform in one cycle and wakes up when it need to send/receive any data. The main disadvantages of the FPS are that: 1) It is not flexible. 2) It requires complete synchronization. To address the inflexibility issue, FPS proposed an on-demand reservation mechanism, where the node can reserve a slot in advance. To address the strict synchronization issue, long slots are used to have coarse grain synchronization.

f) Wakeup scheduling in wireless sensor networks

Abtin et al. (2006) proposed a multi parent based approach which can be used in conjunction with any of the above mentioned schemes. Here, multiple parameters with different wake intervals are assigned to a node in the network. It can improve the performance as compared to the single parent scheme.

2.4.2.3 Asynchronous Schemes

In asynchronous schemes (Rong et al. 2003), nodes can independently wakeup and still guarantee the overlap active period with its neighbor, to have communication. In the asynchronous scheme, nodes do not exchange any information. Following is the survey of different asynchronous sleep/wake scheduling schemes.

a) Power saving protocols for IEEE 802.11 ad hoc networks

Yu-Chee et al. (2002) first introduced the asynchronous scheme for IEEE 802.11 ad hoc networks. They proposed three asynchronous sleep/wakeup schemes: *Dominating-Awake-Interval*, *Periodically-Fully-Awake-Interval* and *Quorum-Based*. In the *Dominating-Awake-Interval*, a host has to stay awake long enough to ensure that the neighboring hosts each get wake intervals and deliver the packets. *Dominating awake* refers to the fact that the host should have a wake interval at least half of the length of the beacon interval. It guarantees the host beacon window overlaps with the neighboring host wake interval window. In this approach, the neighbor discovery response time is short, that makes it suitable for environments which have mobility. It is power-consuming but has the lowest neighbor discovery time. In *Periodically-Fully-Awake-Interval*, two types of beacon intervals are used: a low power interval and a fully awake interval. In the low power interval, the active window is set to minimum, whereas in the fully wake interval, the active window is set to maximum. When a host decides to go into the power saving mode, its time window is divided in to a fixed length beacon (lower power and fully wake intervals). Fully wake intervals are scheduled after fixed intervals and the rest of the intervals are low power intervals. In this approach, response time is a bit high and this protocol is more adequate for slowly mobile environments. It can get a balanced energy consumption and neighbor discovery time. In the *Quorum-based approach*, the concept of quorum is introduced. Here, the power saving node sends a beacon to $O(1/n)$ with all its beacon intervals. Of the three approaches proposed in this paper, the Quorum-based protocol has the maximum power consumption and has got the longest neighbor discovery time.

b) RAW

Random Asynchronous Wakeup (RAW) (Paruchuri et al. 2004) considers the high density of the node in WSNs and allows for the existence of more than one path between two nodes in a network. It combines routing with a random wakeup approach. The routing protocol is a variation of geographical routing. In geographical routing, the data is sent to the node that is closest to the destination. Whereas, in RAW, data is sent to any active node in the forwarding candidate set which contains nodes that meet the specified criteria. The notion behind random wakeup is that, nodes wakeup randomly once in every time slot. It will look for the other nodes that are also active at that time by applying neighbor discovery. As an example, there is *node A* which wants to communication to *node B*. In the forwarding set of *node A* it has got *m* number of nodes then the probability *P* that *node A* could find at least one of these nodes awake is calculated by the following equation.

$$P = 1 - \left(1 - \frac{2T_a}{T}\right)^n \tag{2.3}$$

where

T is time slot of fixed interval

T_a is the active time for each sensor node where $T_a < T$.

m is the neighbors in the forwarding set of *node A* to which a packet destined to *node B* can be transmitted.

In the WSN having a high network density, the number of nodes (n) in the forwarding candidate set will be high, then, by equation (2.3) probability *P* of finding active node will also be high. The advantage of the random wakeup method includes its simplicity and use of local information that makes it suitable for frequently changing topology. On the downside, it is not suitable for a sparse network. Additionally, RAW does not guarantee finding an active node, which makes it difficult to forward the data within the time frame.

c) Minimizing delay and maximizing lifetime for wireless sensor networks with anycast

In (Joohwan et al. 2008), the anycast packet-forwarding scheme is proposed, where each node has multiple next-hop relaying nodes in a candidate set referred to as a forwarding set. Thus, when a node has data to send it needs to wait for one specified next hop neighbour to wake, or rather, it forwards the packet to the *first* node that wakes up in the forwarding set. Therefore, it reduces the expected one-hop delay.

Through the extensive literature review, it is concluded that, sleep/wake scheduling protocols found in literature use periodic sleep/wake intervals and can provide effective energy conservation, but sacrifice delay and throughput. There is a trade-off between latency and energy savings, and most sleep scheduling schemes found in literatures have focused more on energy savings. The increase in end-to-end delay incurred due to latency-energy trade-off can result in delayed arrival of events at the BS, which has the potential to become a major problem in many emerging delay-sensitive WSN applications, which require a fast response and real-time control. Thus, the introduction of the periodic sleep mode in sleep/wake scheduling, results in high latency and end-to-end delays. This delay has to be dealt with, to achieve good power conversation under the limitations of the maximum latency.

Generally, in sleep/wake schemes in order to save energy, each node wakes up after periodic intervals independent of neighboring nodes and all the nodes in the network use the same generic periodic sleep/wake intervals (see Figure 2.3). However, as WSNs use multi-hop communication and in a traditional multi-hop data gathering algorithm, each node has one nominated next-hop forwarding node in the neighborhood, and it needs to wait for the next-hop node to wake up when it intends to forward a packet. Thus, to do the transmission, the sender node has to wait for the arrival of the wakeup time of the next hop forwarding node. Similarly, the next forwarding nodes have to wait for the wakeup interval of its next hop and so on, until the message reaches the BS. Consequently, due to the generic sleep/wake interval and independence in the node's wake-up process, additional delays are added in the packet dissemination at each node along the path to the BS because before transmitting the data, each node has to wait for the next hop neighbor to wake up. All these delays at each hop contribute to the final end-to-end delay of the packet. Along the forwarding path at each hop, this type of delay accumulates and ultimately causes a very long latency as the forwarding hop count increases. This increase in delay is equal to the product of the number of intermediate forwarders times the length of the wakeup interval. This delay is unacceptable for many emerging delay-sensitive WSN applications which require a fast response, such as military surveillance, tsunami alarms, smart hospitals, seismic detection, biomedical health monitoring, hazardous environment sensing, fire detection, intrusion detection, disaster monitoring, and real-time control.

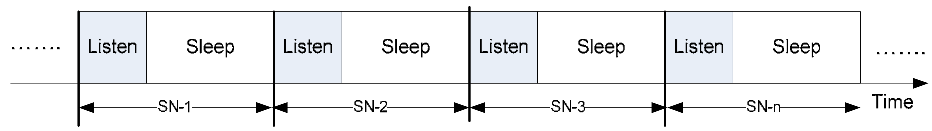


Figure 2.3 Generic Periodic sleep and listen schedule for network nodes

In this thesis, QoS based energy-efficient routing in WSNs is considered. To do so, the *delay-minimization problem* in sleep/wake scheduling is investigated for event-driven sensor networks in delay-sensitive applications. For this reason, the QoS based Sleep/wake Scheduling (QoSES) Protocol for WSNs is proposed. QoSES provides a distributed and low-complexity solution to the delay minimization problem is developed. QoSES differs from previous approaches as it concentrates on minimizing end-to-end delay and at the same time maintaining reasonable energy savings. The objectives of the scheme are to construct an appropriate sleep/wake schedule for the nodes according to their traffic load requirements and to attain a low level of end-to-end delay from the sensors to the BS, while at the same time maintaining reasonable power conservation. To reduce the delay, the proposed protocol does not use the generic sleep/wake schedule for all the nodes, rather it uses a heuristic one which maximizes the active duration of the nodes according to their expected traffic load at three different levels.

First, it is noticed that the forwarding requirement of the nodes are different according to their geographical distance from the BS. For instance, nodes near to BS have to do more forwarding jobs as compared to the nodes away from the BS. For the scenario, the approach in which the nodes have the same generic sleep/wake schedule, results in maximizing delay, because due to the heavy traffic the schedule misses increase as the packet is forwarded to the nodes near the BS. Thus, to cope with heavy traffic and to minimize delay, the nodes' wake intervals are increased as the distance of the node from the BS decreases. This is to make sure that although there is high traffic, the packet should not suffer high delay at relay nodes near the BS due to schedule misses.

Second, nodes in the network have different wake interval requirements, according to their variance in the traffic load due to their topological importance. Topological importance can be determined by node’s role in the network connectivity. For instance, the node which happens to be a cut vertex (only node to connect two parts of the network), is expected to have a greater traffic load as compared to the normal nodes; this is because the traffic of the two sub-networks will be forwarded through this node. In this case, the approach using a generic sleep/wake schedule for all the nodes result in maximizing delay, because due to generic sleep/wake schedule, the schedule misses increase as the packet is forwarded to these heavily loaded nodes. Thus, to avoid congestion and delay such heavily loaded nodes should be given a greater wake interval as they are connecting two parts of the network and expected to forward more data.

Third, in case of event occurrence at a node, by temporal and local dependency, the node is expected to have an event occurrence in the near future (temporal) and the vicinity node is expected to have an event occurrence (local), i.e. the traffic at the node where the event occurs and its vicinity nodes is expected to increase. In this case, nodes having generic sleep/wake schedules result in maximizing delay. Because of the generic sleep/wake schedule, the schedule misses increase as packets are forwarded to the nodes in the proximity of the event occurrence node due to high traffic. The event occurrence generates a burst of traffic which can be communicated in a timely manner when the wake intervals of the node itself and nodes in its vicinity are increased. Thus, the wake intervals of the node where the event occurs and of the neighboring nodes are increased upon an event occurrence.

The detailed operations of these QoSES are elaborated later in the methodology section 3.2.

2.4.3 Mobile Sink based Approach

Collecting data from source sensor nodes to the sink is a common and challenging task in many WSN applications. The sensor node in a WSN has limited energy and is required to work for several months or even years without recharging. Generally, the sensor node does not have enough power or the communication range to directly forward the sensed data to the BS. Hence, a sensor node would not only sense and send its own data but also act as router and propagate the data of its neighbors. In a typical WSN architecture, the sink is static and using multi-hop communication all the nodes in the network forward their data to the sink. In WSNs with static sinks, due to excessive loads, the nodes near the static sink expire earlier; this problem is termed as a *hot spot problem*. *Hot spot problems* refer to the phenomenon when the nodes near to the sink are quickly drained of their energy on account of relaying the data of the nodes farther from the sink. As a result, though the nodes farther from the sink have significant energy left, their energy cannot be utilized as the nodes near the sink have already depleted their energy and hence, they cannot send their data to the sink across the hot spot or energy hole near the sink. It can significantly minimize the network lifetime. Thus, the lack of energy of the nodes near the static sink causes the bottleneck in WSNs. The static nature of the sink is the main reason behind the hot spot problem, as the same nodes near the sink have to forward the data continuously. The main advantage of the static sink approach is that it involves less end-to-end delay.

In recent years, contrary to the static sink approach, the mobile sink approach has attracted much research interest because of the increase in its potential in WSN applications and its potential to improve network performance such as energy efficiency and throughput. Thus, to improve the network lifetime, the notion of mobility is used to have a balanced use of sensor nodes in WSNs. Yang et al. (2010) have done a comprehensive survey of the research efforts that use mobility to improve the network lifetime. It divided the protocols into three classes: mechanisms using mobile sinks, mechanisms using mobile sensors redeployment, and mechanisms using mobile relays (Yang et al. 2010).

2.4.3.1 Mechanisms using mobile sinks

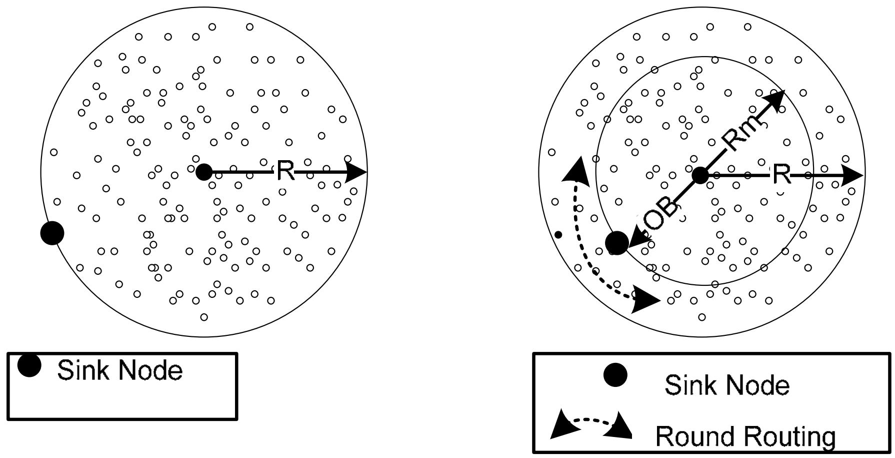
In these mechanisms, the sink is mobile and moves around in the whole network to collect data from the sensor nodes. Nodes that happen to be in the neighborhood of the sink send data to the mobile sink either through a single hop or multi-hop communication. Mobile sinks move to a new location when nodes near the mobile sink become low in energy. Consequently, nodes near the mobile sink change which results in increasing the network lifetime by having a balanced use of the sensor node's energy. Following is the review of literature for protocols using the mobile sink mechanism (Yang et al. 2010).

a) Algorithms with a pre-determined sink mobility path

In these protocols, the mobile sink will move along the pre-defined path and collect the data from the sensor nodes. To follow is the review of literature for algorithms with a pre-determined sink mobility path.

i) Joint mobility and routing for lifetime elongation in wireless sensor networks

In JMR (Jun and Hubaux 2005), a routing protocol is proposed to maximize the network lifetime by solving the load balancing problem using a min-max solution. It employs a mobile sink to gather data from the sensor nodes. It fixes the routing strategy to the shortest path algorithm and looks for an optimal movement strategy. Based on the optimal movement strategy, it looks for a routing strategy that can perform better than the shortest path algorithm. Through theoretical analysis it concludes that the optimal mobility strategy is the circular trajectory (see Figure 2.4 (a) redrawn from (Yang et al. 2010). The paper then proposes a routing strategy where the sink moves in a circle with the radius R_m where $R_m < R$ (*R* is the original radius of the circle). The network is divided into two parts (see Figure 2.4 (b) redrawn from (Yang et al. 2010)): the area of the circle with *Rm* , and the area of the circle between the inner circle and the periphery of the network (named annulus). The routing strategy in the inner circle is the shortest path and the routing strategy in the annulus is round routing until it reaches the *OB* which is the distance between the centre of the circle and shortest path to the sink.



(a) JMR network organization (b) JMR round routing

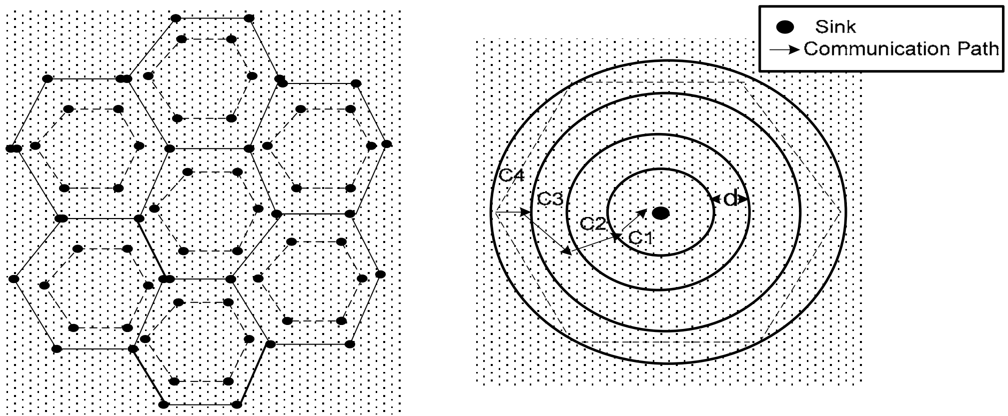
Figure 2.4 Joint mobility and routing strategy

ii) A data gathering algorithm for a mobile sink in large-scale sensor networks

Saad et al. (2008) proposed a movement strategy for a mobile sink in a hierarchical architecture. Sensor nodes are randomly deployed in the network. The network is divided into small groups (clusters) and each cluster has one leader node (cluster head). The cluster head collects data from its member nodes and forward it to the mobile sink when it comes into its neighborhood. The mobile sink starts at a specified point, moves in the network along the defined path, collects the data buffered at the cluster head, and then returns back to the starting position. The moving strategy is divided into two parts: mobile sink *path identification* and *path optimization*. In *path identification*, clusters are formed using clustering; the centroid of each cluster is calculated. Nearby CHs (maximum two hop) are grouped together, and more CHs are added to the group until it is maximized and no more CHs can be added further. The sink path is formed by the centroid points of each cluster. In *path optimization*, the mobile sink first moves along the path passing through the centroid points of each cluster. Thus, each CH can communicate with the sink at the maximum one hop. Bee's algorithm is then used for path optimization.

iii) Improved sensor network lifetime with multiple mobile sinks

Marta and Cardei (2009) proposed a data gathering protocol for WSNs using mobile sinks having a pre defined path. The WSN employed is heterogeneous, with limited energy sensor nodes and unlimited energy multiple mobile sinks implementing periodic data gathering. The sensors send their data to the nearest mobile sink using multi-hop communication. Their paper addresses the *sink mobility network lifetime increase (SMN-LI)* problem; the sink movement plan is proposed to increase network lifetime. The whole sensing area is divided into hexagon cells as shown in Figure 2.5(a), redrawn from (Marta and Cardei 2009), where the sink is located at the hexagon centre. The hexagon is divided into a corona of length *d,* where *d* is the sensor's communication range. The message is generated and forwarded by *C4* to *C3*, *C3* forwards it to *C2,* and so on until it reaches the sink (Figure 2.5(b), redrawn from (Marta and Cardei 2009)). The first case considered, is when the sink moves along the corners of the hexagon. It starts from one hexagon corner, stays there for time *T*, collects the data from the neighboring nodes, and then moves to the next corner of the hexagon till it finishes all corners and the step are repeated again. Nodes in the first corona consume more energy than other nodes in the hexagon as they need to do data forwarding to relay the data of the nodes in the outer corona. Thus, the protocol avoids the scenario where nodes are the first corona of two hexagons. To do so, the sink moves in a hexagon which lies inner to the original hexagon. Simulation results show that the sink moving to the hexagon corner achieves 3.48 times improvement over the static sink lying in the centre of the hexagon. In another experiment, the sink is allowed to take many positions at the hexagon parameter. Simulation parameters are used to control the number of stops of the sink; simulation results demonstrate that when 12-positions stop at the hexagon parameter, it achieves a 4.86 times improvement in the network lifetime compared to the static sink and 1.39 times improvement in the network lifetime compared to a 6-position case.



(a) Sinks movement trajectory (b) Corona division for a sink

Figure 2.5 Corona division and sink trajectories

iv) Controllably mobile infrastructure for low energy embedded networks

Somasundara et al. (2006) proposed an adaptive motion control (AMC) for the mobile node is proposed. The main objective of the paper is to plan a mobility path for the sink to increase the data delivery and traverse the path in the specified delay constraints. Congestion regions of the sensor nodes are found out by the mobile nodes. Based on these congestion regions, the mobile sink plans its mobility path, mobile sink speed and how long it needs to stay at a particular region in order to increase the network lifetime.

v) Energy efficient schemes for wireless sensor networks with multiple mobile base stations

Gandham et al. (2003) to find out new locations for the mobile sink, used an integer linear program along with the flow-based routing (FBR) protocol. At the beginning of each round, a new position where the BS will be moving is decided, and during that round these positions remain the same.

b) Algorithms with autonomous sink movement

Yanzhong et al. (2007) proposed an autonomous mobile sink movement strategy for mobile sink based WSNs. Sensor nodes discover their one hop neighbor by sending a hello message to its one hop neighbor. The sink nodes send messages to the sensor nodes informing them about their position. Since the sensor nodes and sink are all location aware, to report the data back to the sink, the nodes use location based routing. The sink decides on a new moving location based on the energy level of the sensor node collected during data gathering. The sink arrives to the new position before the start of the next data gathering cycle. When the sensor nodes have completed their communication, they can go into the sleep state and wakeup again when a sink comes into their neighborhood.

i) Half-quadrant based moving strategy (HUMS)

The autonomous sink moving strategy called half-quadrant based moving strategy (HUMS) (Ma et al. 2007) is proposed. To balance the energy usage among sensor nodes in the network, the mobile sink (referred as *energy mower*) moves to the nodes with higher residual energy. During data gathering, the mobile sink collects energy information of the sensor nodes including its residual energy as well as the location of the node with the highest and lowest energy level. Mobile sink movement is done in rounds, where in each round the mobile sink moves to one hop. After each round, the mobile sink re-calculates its next *moving destination (MovDest)* based on the new energy distribution. The mobile sink in HUMS makes a moving decision based on a coordinate system, considering its current location as the origin and then dividing the coordinate system into eight half-quadrants. A set of sensor nodes that have the lowest energy in the network in one period is found and termed as a *quasi- hot spot*. The mobile sink will define its trajectory based on the quasi hotpot's location and its destination location. Considering the distance of the mobile sink from the MovDest, there are two cases. *The first case* is when the mobile sink is far from the MovDest, the half quadrant based algorithm is used to avoid the mobile sink moving to the location of the quasi-hotspot. If no clean neighbor sectors exist on the left or right of the destination sector, then the minimum influence position selection algorithm (MIPS) is used. In the *second case*, when the sink is close to the destination, then, square hoping is used to find out the sojourn position near the destination. In the square hopping approach, four points are selected near the destination on a circle (the radius of the circle is equal to the communication range of the sensor node). The goal is it to move the sink one hop away from the destination in such a way that the high energy node (destination node) does the data forwarding on behalf of the other nodes.

ii) Adaptive location updates for mobile sinks in wireless sensor networks

Wang et al. (2009) proposed the Adaptive Local Update based Routing Protocol (ALURP), which is an extension of their previous work LURP. The sink broadcasts its location information to the whole network after deployment. The routing process consists of two phases. The first phase, involves forwarding the packets from the sensor node to the destination area. The second phase includes forwarding the packets in the destination area to the sink itself. To make other nodes aware of the position of the sink, when the sink moves in the destination area it will broadcast its location. When the sink moves outside the destination area, there may be some nodes far away from the sink which may not know the new location of the sink, as they do not hear the sink location message. In this case, these sensor nodes send the data packet to the known destination area from there it is forwarded to the new sink location.

iii) Improved sensor network lifetime with multiple mobile sinks

Marta and Cardei addressed the SM-NLI problem for autonomously moving the sink in such a way that 1) the sinks are always interconnected and 2) network lifetime is prolonged (Marta and Cardei 2009). Each round starts with the formation of a data collection tree using the clustering mechanism. Each sink acts as a cluster head and each node maintains information about the closest sink. After the cluster formation, data is collected along the shortest path established. During data forwarding, sensor nodes also piggy back their current energy level to the sink, which is later used in the sink movement. At the end of the round, the sink makes its movement decision based on the energy level of its first hop neighbors and moves to the zone having higher energy resources. Throughout this process, sinks remain interconnected all the time. To find the new location, the sink uses the incremental ring approach. The first preference of the sink is to find the location nearest to the old location. If no such location is found, then the sink increments the number of hops needed to increase its search domain. If no location is found, even after the whole cluster has been scanned, then the sink will not move at all. When a sink has a candidate location to move to, it will first check the backbone connectivity status. The sink has got two transceivers, one is for communication with the sensor nodes and the other is for communicating with the other sinks. To check connectivity, the sink exchanges *HELLO* messages with its one hop neighbor. To check connectivity the Breadth-First-Search algorithm (Cormen et al. 2001) is used, which uses one hop neighbor information as well as candidate sink location.

Furthermore, the coverage based SM-NLI problem is addressed where the sink autonomously moves and the sensor node alternates between sleep and wake states in such a way that: firstly, at any particular instance of time active nodes are providing full coverage of the area. Secondly, sinks in the network remain interconnected forming the backbone interconnection. Thirdly, the network lifetime is prolonged.

iv) Moving schemes for mobile sinks in wireless sensor networks

Yanzhong et al. (2007) proposed two autonomous movement strategies for mobile sinks: *one-step moving scheme (OSMS)* and multi-step moving scheme *(MSMS).* OSMOS is similar to the HUMS algorithm (Ma et al. 2007) with no half quadrant strategy used. Here, the sink can move to the destination region in one step, hence avoiding low energy sensor nodes on the moving path. In MSMS, two-hop neighboring nodes' information is used for making the sink movement decision.

2.4.3.2 Mechanisms using mobile sensors redeployment

In many WSN applications, sensors are randomly deployed in the sensing area, thus it is very difficult to control the initial deployment. In this case to minimize the energy hole and minimize other anomalies in the initial deployment, mobile sensors can prove very handy by relocating themselves to the required position. By doing so, with good deployment, energy is balanced and network lifetime is maximized. After this initial post deployment relocation phase, sensor nodes become static and begin their normal data sensing/transmission job.

In (Cardei et al. 2008; Yinying and Mihaela 2007) the sink is located in the centre of the sensing area and the sensor nodes send their data to the sink. The sensing area is virtually divided in to a corona, where C_1 is the inner most corona and C_n is the outer most corona. Messages generated from the outermost corona is forwarded to C_{i-1}, which transmits it to the C_{i-2} and so on until the data reaches C_1 from where it is forwarded to the sink. In this architecture, the nodes in the inner most corona will deplete their energy earlier as they are required to forward the data of all the outer corona. Thus, to have a balanced use of sensor node energy, the sensor density is readjusted where the number of nodes deployed in the inner coronas are kept higher than the nodes in the outer corona, i.e. as the inner most corona C_1 has a maximum node per unit area, corona C_2 has less than C_1 and so on with the outer most corona having the least number of sensor nodes. In this way, all the nodes in the network deplete their energy in a balanced way and ultimately results in increasing the network lifetime.

a) Movement-assisted sensor redeployment scheme

In (Yinying and Mihaela 2007) a flip based mobility model is proposed. Here, the sensor moving distance and number of moves is fixed. To achieve the desired sensor density, nodes are relocated after initial deployment by using a centralized mechanism. The objective is to maximize the overall network lifetime by having a minimum number of moves.

b) Non-uniform sensor deployment in mobile WSNs

In (Cardei et al. 2008), a more flexible solution to the mobility model is proposed. The number of sensor movements is not limited and the sensor nodes are relocated to achieve the required sensor density to maximize the network lifetime having minimum sensor moves. For a solution to this problem, three mechanisms are proposed: an *Integer Programming (IP) approach*, a *localized matching method*, and a *scan-based mechanism*. In the *Inter Programming mechanism*, the objective function is to decrease the total moving distance of a sensor. If the current number of sensors in a corona is greater than the desired number of sensors, then it is a source region. If the current number of sensors in a corona is less than, equal to or greater than the desired number of sensors then it is a hole region, neutral or source region, respectively. The objective is to make every region a neutral region, by attaining a sufficient number of sensors in each region. *The localized matching mechanism* operates in three steps. In the first step, a whole region broadcasts in its neighborhood, the request is for the sensor movement to achieve the required sensor density. In the second step, source regions respond to the whole region request by sending the number of nodes they can contribute to the whole region. In step 3, the whole region decides from which region to accept nodes based on the distance moved. It broadcasts its decision to the localized regions and the process ends with the actual node movement.

In the *scan based mechanism,* the sensing region is divided into a corona and sectors. Two types of scans are done: *corona scan* and *radius scan*. In the *corona scan*, the number of nodes per corona is balanced and after the end of this scan, regions in the same corona are balanced and will have the same number of nodes. In the *radius scan,* according to the desired sensor density, nodes are re-distributed in the sectors. It has got two sweeps. In the first sweep, the total nodes per sector are calculated. In the second sweep, the final number of sensors in each region and sensor movement is calculated. Simulation results presented in the paper showed that when it is evaluated against uniform deployment, the proposed three schemes are able to prolong the network lifetime.

2.4.3.3 Mechanisms using mobile relays

Mobile sensor nodes can be used to prolong the network lifetime by moving them to different regions and assigning them the duty of relay nodes. In that case, mobile nodes move to any potential energy hole or low energy region in the network and do the relaying on behalf of the co-located static sensor. Static nodes in such regions can go into sleep mode to save energy. In this way, mobile nodes can increase the network lifetime. Additionally, mobile nodes can be used as data collectors by carrying other sensors' data to the sink. When these mobile nodes get in communication range of the sink, they deliver the data to the sink. Hence, saving their energy which otherwise would have been wasted in multi-hop communication.

a) Using mobile relays to prolong the lifetime of WSNs

Ke et al. (2005) proposed a mobile relay based scheme to maximize the network lifetime. Assume that the whole network consists of two components: component *A* and component *B* with two sensor nodes *i* and *j* connecting these two components. These two nodes are responsible for sending all the data from component *A* to component *B*. It can significantly limit the network lifetime as they can deplete earlier because of excessive use, leaving the energy hole. Mobile relay nodes can be used here to do the data forwarding on behalf of node *i* and *j*, to prolong the network lifetime. One possible way of doing so is for the mobile relay node to alternatively move between node *i* and *j*. It does the data forwarding on behalf of node *i* if it is near node *i* and node *j* if it near node *j* (see Figure 2.6 redrawn from (Yang et al. 2010)).

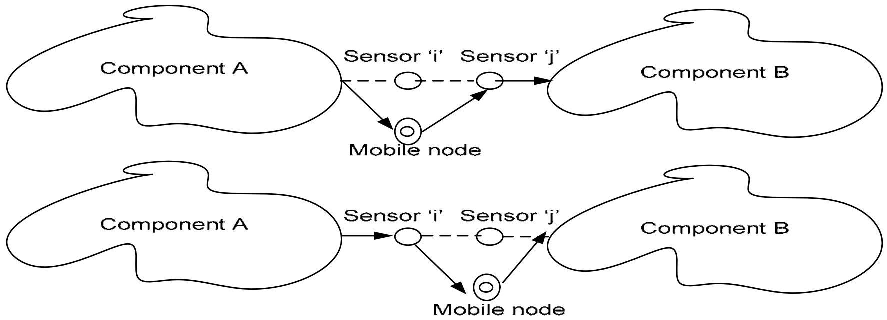


Figure 2.6 An example of using mobile relay nodes to prolong the network lifetime.

b) Data mules: modeling a three-tier architecture for sparse WSNs

Shah et al. (2003) proposed a three tier, mobile MULE based architecture. The mobile MULE exhibits symmetric random walk mobility. It collects data from the sensors which are in its neighborhood and delivers that data to the access point. The first tier includes the interconnection of the access points. The second tier includes the mobile MULE which acts as an intermediate between the upper and lower tiers. It collects data from the lower tier (sensor node) and sends the data to the upper tier (access point). They are resource rich and have no energy issues. The lower tier includes randomly deployed wireless sensor nodes. These nodes do the actual data sensing task, and forward the data to the nearest mobile MULE. Unlike on the top and intermediate tiers, network devices (sensor nodes) on the bottom are energy constrained, thus their use is minimized to prolonging the network lifetime. Although this protocol achieves energy efficiency, it increases the network delay. Thus, it is suitable for delay tolerant applications.

c) An extended localized algorithm for connected dominating set formation in ad hoc wireless networks

Dai and Wu uses controlled mobility to minimize the delay. Sensor nodes are randomly deployed and the network is divided into a grid (Dai and Wu 2004). Clustering is done among the static/mobile sensor nodes, such that, each cluster contains at least one mobile and one static node. Each cluster has four short links to connect it with its right, left, top and bottom clusters and it has *q* long links. A sensor *u* in $m \times m$ space through long links may have contacts with $5m \times 5m$ space, i.e. cluster *v*. To pick up and deliver data, mobile nodes move between *u* and *v* using the long link. To forward data, the sensor nodes in a grid will choose a short or long link whichever is closer to the destination.

Banerjee and Agrawal investigated the mobility of the cluster head in the clustered based WSN to prolong the network lifetime (Banerjee and Agrawal 2008). The paper considered WSNs with randomly deployed sensor nodes other then CHs (which are mobile nodes), all the other sensor nodes and sink are static. Static nodes form the cluster by selecting mobile nodes as the CH. Each sensor node joins one of the nearest mobile nodes acting as the CH. The CH moves in the cluster and communicates with other CHs, to minimize the cluster overlapping area. Each CH can directly communicate with the sink and during its movement the CH maintains a connected path with the sink. Each sensor node reports the data to the nearest CH using single or multi-hop communication and the CH then forwards the data to the sink. Furthermore, the paper proposed three mobility strategies for the mobile cluster node: *energy efficient mobility*, *event-oriented mobility*, and *hybrid*. In *energy efficient mobility*, based on the energy usage history of the sensor nodes in the cluster, the mobile cluster nodes predict the residual energy of the nodes in the cluster. Based on this residual energy information, the cluster moves to the point where the residual energy of the nodes is high. It results in a balanced use of sensor node energy and helps in prolonging the network lifetime. In *event oriented mobility*, the mobile cluster head moves to the region in the cluster, where more events are occurring and more data is being generated. It will improve the energy consumption of the sensor nodes around the event, by minimizing the costly multi-hop communication of data between sensor nodes and the mobile cluster head. In the *hybrid mobility strategy*, the nodes' residual energy as well as event location is considered. Using energy efficient mobility, a mobile cluster head may stay away from the event all the while using event oriented mobility, the mobile cluster head may deplete the energy of the node near the event. The hybrid strategy combines these two approaches, where the mobile cluster head first moves towards the event, and then exhibits the energy efficient mobility. Thereby, making sure that data forwarding is energy efficient and the mobile cluster head stays near to the event.

Through the extensive literature review, it is concluded that a number of papers can be found discussing energy efficient in WSN. However, to the best of our knowledge no work considers QoS based data gathering in hybrid (static, mobile sink) based clustered wireless sensor network s to maximize energy efficiency and minimize end-to-end delay using the QoS based prioritization approach.

In this thesis, QoS based energy-efficient routing in WSNs is considered. For this reason, the QoS based Energy Efficient Mobile Sink (QoSEM) based Routing Protocol for Clustered Wireless Sensor Network is proposed. The main objective of QoSEM is to provide QoS based energy efficient routing by using a combination of static and mobile sinks with prioritization of data based on QoS requirements (bandwidth, delay). Each in-coming packet is assigned some priority based on message type and content. Normal nodes associate themselves to one of the nearest CH nodes. The CH is associated with a super node. When a node has some data to send to a BS, it is forwarded to the CH based on its priority (first sending the packets with higher priority). Super nodes gather data from the CH in its vicinity and forward this data either to a static sink or a mobile sink based on the priority of the message.

The detailed operations of these QoSEM are elaborated later in the methodology section 3.3

2.5 Summary

This chapter presented an overview of WSNs, the parts/types of hardware configuration of typical sensor nodes, applications of WSNs, and challenges in protocol design for WSNs. The review of the background work summarized in this chapter has helped in developing the understanding of basic concepts and terminologies of WSNs. Furthermore, the review of the related work and issues in the existing QoS aware energy efficient approaches has been presented. Related work on QoS aware routing in WSNs was presented in the perspective of the clustering approach, sleep/wake scheduling, and mobile sink approach. The chapter discussed in detail the related work on the clustering approach, sleep/wake scheduling, and mobile sink approach. At the end of each of these three sub-sections, potential problem statements were briefly discussed and the proposed solutions were briefly overviewed.

Based on the literature review summarized in this chapter, three protocols are proposed for QoS aware Energy Efficient Routing in WSNs, namely: QoS aware Energy Efficient Clustering (QoSEC) for WSNs, QoS aware Energy Efficient Sleep/wake (QoSES) Scheduling for WSNs, and QoS aware Energy Efficient Mobile Sink (QoSEM) based Routing Protocol for Clustered WSNs. The next chapter will explain in detail the workings of the three proposed protocols.

CHAPTER 3

METHODOLOGY

In this thesis, QoS based energy efficient routing in WSNs is considered. To provide QoS and energy efficiency, three perspectives of typical data gathering protocols are considered, namely: Clustering approach, Sleep/wake scheduling approach and mobile sink based approach. This chapter will discuss the proposed protocols in detail, namely: QoS based Energy Efficient Clustering (QoSEC) for WSNs, QoS based Energy Efficient Sleep/wake (QoSES) Scheduling for WSNs and QoS based Energy Efficient Mobile Sink (QoSEM) based Routing Protocol for Clustered WSNs. Each section that explains these protocols (QoSEC, QoSES and QoSEM) gives a detailed description of the problem addressed, followed by a network model and assumptions of the proposed protocols. It then discusses the preliminaries (data structures used at sensor nodes) of the proposed protocol, and finally, a detailed description of workings of the proposed protocols is presented.

3.1 QoS based Energy Efficient Clustering (QoSEC) Protocol for WSN

Typically, sensor nodes have non-rechargeable batteries. This makes energy efficiency one of the critical routing design issues in a Wireless Sensor Network (WSN). Clustering has been identified as an energy efficient routing technique in the WSN. In this section, the QoS based Energy Efficient Clustering (QoSEC) Protocol for Wireless Sensor Networks is proposed, where energy efficiency is achieved by deploying some nodes with additional energy resources, termed as super-nodes, in a hierarchical manner. These super-nodes, together with the normal capability sensor nodes, are used to forward data to a Base Station (BS). Multi-hierarchy clustering is achieved by placing super-nodes at the top tier, the cluster head at the middle tier, and cluster members at the lowest tier. Once clustering is performed, the CH collects data from its cluster nodes and forwards it to the nearest super-node, which subsequently forwards it to the BS using multi-hop super-node communication. With QoSEC, *CH selection* and *sleep/wake scheduling* algorithms are executed to optimize the network and coverage lifetime. The efficiency of the proposed QoSEC is observed through simulation and its performance is compared against the contemporary CPCP, EEMC and LEACH protocols. The results demonstrate that QoSEC has prolonged the network and coverage lifetime as well as improved the other QoS routing parameters, such as delay, packet loss ratio, and throughput. Following sections will explain in detail how QoSEC works.

3.1.1 Problem Description and Overview of Proposed Solution

As stated earlier in section 3.1, clustering has been identified to be an energy-efficient routing technique in a WSN. With the clustering approach, data forwarding to the sink/BS is performed by using multi-hop communication, in which each CH sends data to its neighboring CH, and so on until the data reaches the BS (see Figure 3.1). Ultimately, nodes nearest to a BS send data directly to the BS. However, the potential disadvantage of this technique is that the CH nearest to the BS may over utilize their energy resulting in depletion of the sensor node's energy nearest to the BS. This problem is termed as a *hot spot* or *energy-hole problem*. The hot spot problem is an inherent phenomenon in a WSN and can significantly decrease the lifetime of the overall network. As in this scenario, though the nodes that are a farther distance away from the BS still have significant energy left, their energy cannot be utilized as the nodes nearer the BS have already died. Consequently, they cannot forward their data to the BS.

To solve the hot spot problem and to minimize its impact on WSN lifetime, the QoS based Energy Efficient Clustering (QoSEC) Protocol for Wireless Sensor Networks is proposed. With QoSEC, energy efficiency is achieved by minimizing the hot spots by avoiding using only the nodes closer to the BS to forward data to the BS. Rather, forwarding data to the BS in a heterogeneous environment is considered; there some nodes with additional energy resources, called *super-nodes*, are used. Like the other normal nodes, these super-nodes are also uniformly deployed across the sensing area. The idea is for the super-nodes to behave as local sinks. Normal nodes associate themselves to one of the super-nodes in their locality. Energy-efficient clustering is done among the normal nodes. When a CH receives the sensed data from all the nodes, it aggregates that data. The CH sends the data to the super-node to which it is associated. Super-nodes collect the data from all the CHs within its vicinity. Super-nodes either use multi-hop super-node communication to forward the data to the BS or they send the data directly to the BS. In this way, the hot spot problem is avoided because nodes do not send the data all the way to the BS. Instead, they just have to forward the data to the super-node (local sink) within their locality. The responsibility for forwarding data to the BS is distributed among the super-nodes, which are not energy constrained. Thus, traffic load at the nodes near the BS is minimized, and consequently hot spot problems do not develop.

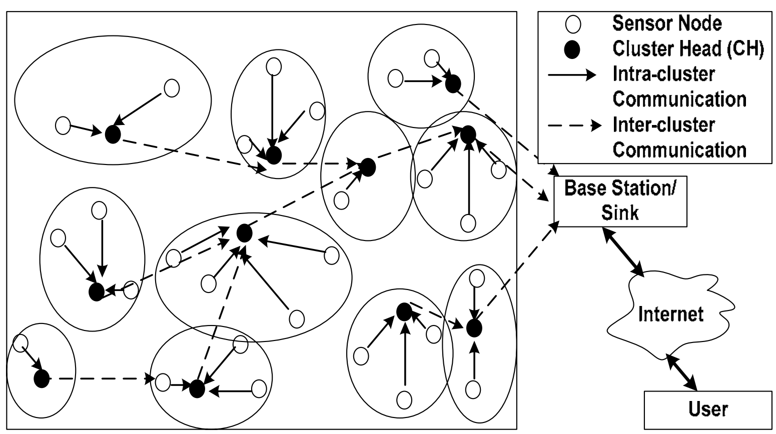


Figure 3.1 A typical Clustering based data gathering in a Wireless Sensor Network

Furthermore, to improve coverage and to achieve energy efficiency, a *cluster-head-selection* algorithm and a *sleep/wake scheduling* algorithm are proposed. Firstly, the cluster-head-selection algorithm tends to increase energy efficiency and optimize coverage lifetime by using heuristics, which gives less priority for CH selection to the nodes having lower energy and less overlapping coverage. The purpose is to optimize the network/coverage lifetime because the role of the CH is resource intensive, that is, a node in the CH role consumes more battery power than other nodes. Hence, if nodes with lower energy and less overlapping coverage are selected as the CH, they may deplete soon, resulting in an energy or coverage hole. Secondly, the proposed sleep/wake scheduling algorithm minimizes the wakeup interval for nodes having lower energy and less overlapping coverage. The notion is to optimize the network lifetime and coverage lifetime. If a higher energy node is available in a cluster, selecting the longer sleep interval of the sensor node with less energy will optimize network lifetime. Similarly, if overlapped coverage nodes are available in a node's sensing range, selecting the longer sleep interval of that node having overlapping coverage will improve coverage lifetime. In contrast to this, if nodes with lower energy and less overlapping coverage are selected for a longer wakeup interval, they can deplete sooner, resulting in an energy or coverage hole. This heuristics ensure balanced energy usage of the nodes and have an impact on the coverage of the entire network.

QoSEC, when run on a flat topology WSN, ends up with a multi-hierarchy topology in which sensors are organized into clusters. In each cluster, one node is promoted as the CH and all the other regular sensors connect to the closest CH. All these CHs are associated with the nearest super-nodes which act as local sinks. CHs also act as data relays since they route data received from peer CHs towards the super-node, either directly if the super-node is within their range, or indirectly through a neighbor CH. To control the hot spot problem, the routing task of the normal node is restricted to the super-node (local sink), so that it does not relay data directly to the BS, since it is less eligible than the super-node in terms of battery power. Hence, a super-node performs the energy-intensive data processing and routing tasks of forwarding data to the BS. Consequently, normal node energy usage in data forwarding is minimized at the expense of more energy being consumed at the super-nodes, which have extra energy resources.

3.1.2 Network Model and Assumptions

WSNs consist of a large number of sensor nodes randomly deployed in a certain area (sensing field). After deployment, all the sensor nodes in the WSN will form group clusters, and each cluster elects their CH in a distributed manner using local information. Each CH collects sensor data from other nodes in the cluster and transfers the aggregated data to the next super-node, which then sends it to the BS using multi-hop super-node communication. The CH performs extra duties since it collects, aggregates, and relays data from/to its member nodes. To achieve balanced energy consumption, the role and responsibility as a CH is rotated over time among the member nodes in the cluster. In the proposed QoSEC protocol, the following assumptions are considered.

a. It is assumed that a percentage of the population of the sensor nodes is equipped with additional energy resources. Consequently, WSNs contain two types of nodes: normal nodes and a few nodes equipped with extra energy resources, termed as *super-node*s. Super-nodes are a lot less in number than normal nodes. Both of these node types are randomly deployed in the sensing area (see Figure 3.2).
b. There is a BS located at the center of the network. The BS has a sufficient energy resource. The sensor nodes (both super-nodes and normal nodes) are static and have no mobility as is generally the case in most of WSN applications.
c. The battery (energy), computation, communication capabilities, and sensing range of the normal nodes are the same for all peer normal nodes and consume the same power to transmit one bit of data; the same is true for super-node peers. However, super-nodes have greater energy and communication resources than normal nodes. Each normal node has the same initial energy, while the initial energy of the super-nodes is significantly higher than the initial energy of normal nodes.
d. Normal sensor nodes have limited battery power and because of inaccessible and/or hazardous terrain, are left unattended after deployment. Thus, it is impossible to recharge the battery, for example, in a battlefield, sensor nodes are dispersed in a large target area where reaching and recharging them is extremely difficult and even dangerous. Super-nodes are not energy constrained while normal nodes are energy constrained; this motivates the need for an energy efficiency in communication protocol for normal nodes in order to extend the lifetime of the WSN. Thus, prolonging the network lifetime mainly depends on an energy-efficient protocol design. Energy-aware sensor network protocols are thus required for energy conservation.

e. All sensor nodes have various transmission power levels, and each node can dynamically vary its communication range by changing its transmission power level.
f. The Identificaion (ID) and the position of the sensor nodes are fixed after random deployment and all the sensor nodes know their location by using GPS or any other localization strategy; moreover, all the sensor nodes know the location of the sink.
g. Active sensors sense the event happening within their sensing range and store the data in a little buffer till its TDMA slots to communicate arrive. It then sends the data to the cluster head and deletes the buffer entries.
h. CHs consume more energy than regular sensors have to perform additional data-related tasks like aggregating data, data filtering, and forwarding/relaying data.

It is important to note that in our model, no assumptions are made about 1) the homogeneity of node dispersion in the field, 2) the network density or diameter, 3) the distribution of energy consumption among sensor nodes, 4) the proximity of querying observers, 5) the ability to communicate with the BS, or 6) each node having a fixed communication range.

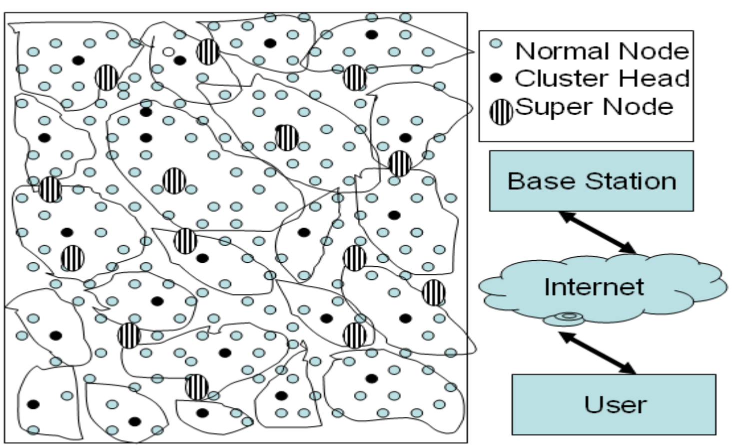


Figure 3.2 A typical QoSEC based data gathering in Wireless Sensor Network

3.1.3 Preliminaries for QoSEC

Following is the brief discussion of the outlines of data structures maintained at a sensor node in QoSEC.

3.1.3.1 Data structure at a sensor node

To assist the routing process in QoSEC, each node maintains some data structure, the usage of these data structures; will come in later section. However, a brief description of these data structures is as follows.

- *Node ID.* Each sensor node is assigned a unique ID. It is used to identify the sender/receiver during normal node communication.

- *Super-node ID.* Each super-node is assigned a unique ID. It is used to identify the sender/receiver during super-node communication.

- *Cluster ID.* A unique ID is assigned to each cluster. It is used to identify which cluster a node belongs to and helps in streamlining the communication between the CH and other cluster member nodes.

- *Next hop to super-node.* Each normal node stores information about the next CH, which acts as the next hop to the super-node. It consists of tuples in the form (CH ID, hop-count, prev-node), where the CH ID is the CH node ID, hop-count is the number of hops leading to this CH, and prev-node is the node ID of a 1-hop neighbor node that can lead to this CH node using the minimum number of hops (part of the shortest path).

- N*ext hop to sink.* Each super-node stores information about the next super-node, which can act as the next hop to the BS/sink. It consists of tuples in the form (SN-ID, hop-count, prev-node), where SN-ID is the super-node ID, hop-count is the number of hops leading to this super-node, and prev-node is the super node ID of a 1-hop neighbor super node that can lead to this super node using the minimum number of hops (part of the shortest path).

3.1.4 Description of the QoSEC Proposed Protocol

In the proposed work, QoSEC has considered energy efficiency as the core design issue for clustering in WSNs. With QoSEC, clustering can be easily explained by dividing it into two main phases, that is, the Setup-Phase and Steady-Phase (see Figure 3.3). Generally, the Steady-Phase takes a longer time than the Setup-Phase.

The following is the detailed explanation of the proposed QoSEC protocol.

3.1.4.1 Setup Phase

During the setup phase, all the sensor nodes in the whole WSN are divided into a number of small groups known as clusters. It is further divided in to five sub-phases (see Figure 3.4): initialization, route update, cluster head selection and cluster formation, sensor activation, and TDMA scheduling (see Figure 3.5).

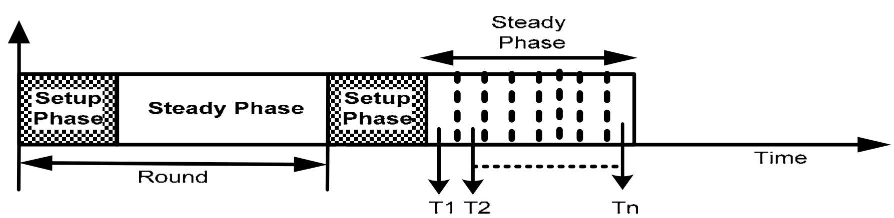


Figure 3.3 Round of sensor network operations in QoSEC protocol

a) Setup Phase: Initialization

After random deployment, each node determines its residual energy (i.e. battery power left) and the location in which it lies. The information is later used in decision making, such as for cluster-head selection and sleep/wake scheduling. Residual energy and the distance to a BS are used as cost calculation in cluster formation.

a) Setup phase: Route Update

The route update phase maintains the route information, which will be used in later phases for inter-clustering communication. The route update consists of two sub-phases: route update to the super-node and route update to the BS.

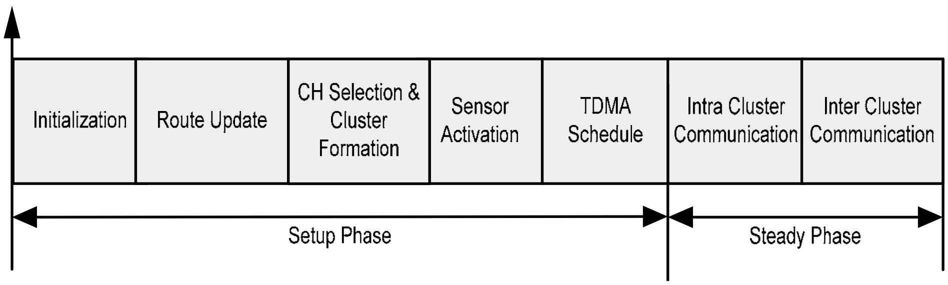


Figure 3.4 QoSEC operations

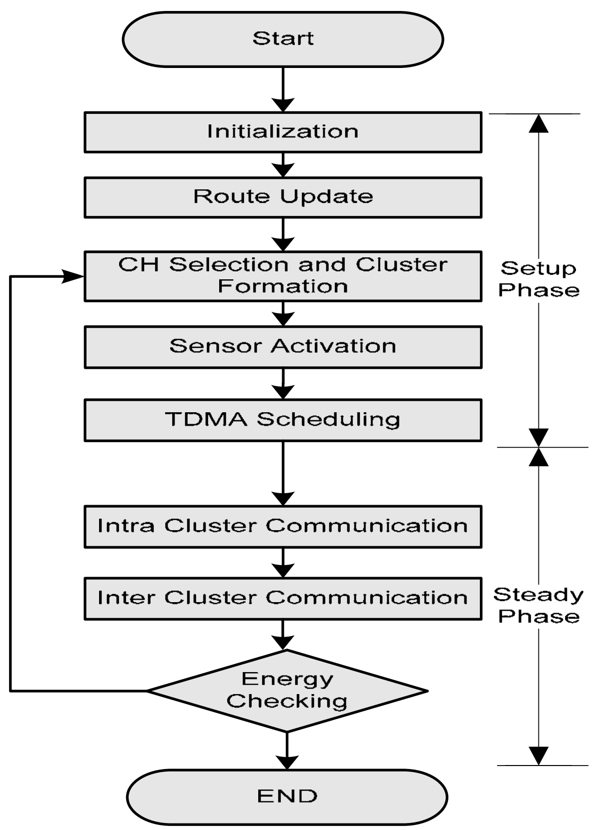


Figure 3.5 Flow chart of major phases and their interaction in QoSEC protocol

i) Route Update to Super-node

To forward data to the super-node, CH nodes use multi-hop CH communication. To develop these routes, each super-node generates a route discovery message with a hop count of 0. It is broadcast within the locality of these super-nodes. Normal nodes are the recipients of this message as these nodes (which could later become a CH) will carry the received data to the super-nodes. Upon receiving this broadcast message, a normal node updates its hop count value, i.e. changes its value to the new value, if the received hop count value is less than the previous hop count value; otherwise, it retains the previous value. Before forwarding the route discovery message, each normal node increments the hop count and then broadcasts the message to nodes in its communication range. In this way, a message arrives at each node following the desired minimum cost path. As a result, each normal node has a minimum hop count path to their respective super-node. Once the hop count value is determined, this route information is later used in inter-clustering communication.

ii) Route Update to BS

In the route update phase, the BS generates a route discovery message with a hop count 0. It is broadcast throughout the network, where super-nodes are the intended recipients of these messages as they are responsible for carrying data to a BS. Upon receiving this broadcast message, a super-node updates its hop count value, that is, changes its value to a new value, if the received hop count value is less than the previous hop count value, otherwise, it retains the previous value. Before forwarding the route discovery message, each super-node increments the hop count and then broadcasts the message to nodes in its communication range. In this way, a message arrives at each super-node following the desired minimum cost path. As a result, each super-node had obtained a minimum hop count path to the BS. Once the hop count value is determined, this route information is later used in forwarding the data to the BS.

After the route update phase, each normal node has a minimum hop count path to the super-node, and each super-node has a minimum hop count path to the BS.

d) Setup Phase: Cluster Head Selection

For cluster-head-selection, the algorithms found in the literature have considered residual energy, number of linked nodes, distance from the BS, and coverage as CH selection parameters. However, generally these parameters are considered singularly. Optimizing one parameter would lead to performance improvement in that parameter but would have to compromise on the performance in other parameters. For instance, considering residual energy as a CH selection parameter could increase the network lifetime, but it could trade-off the coverage because the algorithm will always try to optimize energy usage without considering its impact on the converge, and vice versa. To overcome this problem, both residual energy and coverage are used as CH selection parameters in QoSEC. Consequently, avoiding this drawback of compromising the coverage can be achieved by finding an overlapping coverage area of all the nodes. According to the overlapping coverage area, the nodes are divided into two types: normal nodes and critical nodes (see Figure 3.6). Nodes which have sensing areas overlapped by other neighboring nodes are termed as normal nodes. Nodes having sensing areas not being overlapped by any other node are termed as critical nodes.

During CH selection, nodes that are critical nodes are avoided from being selected as the CH. Hence, when a node has more battery power, but is a critical node in terms of coverage of sensing area (it is the only node covering a particular part of sensing area), it will not be selected as a CH. In this case, to avoid a coverage hole, it is better to select a node which has less battery power but is less critical in terms of coverage (it has one or more other sensor nodes, which are overlapping covering the same part of the sensing area as this node is covering). In this case, even if this less critical node (in terms of coverage) dies, other nodes can cover the same sensing area. Thus, the coverage of the whole sensor network is still intact. Very little works can be found in the literature that consider the coverage of a sensor node as a parameter when selecting a sensor node as the CH (Soro and Heinzelman 2009). With QoSEC, to improve lifetime and coverage, a hybrid function of the parameters, such as battery power and coverage, is used. To achieve this, an optimum node for the CH in terms of energy and coverage, is selected. Algorithm 1 explains how a CH is selected in each cluster.

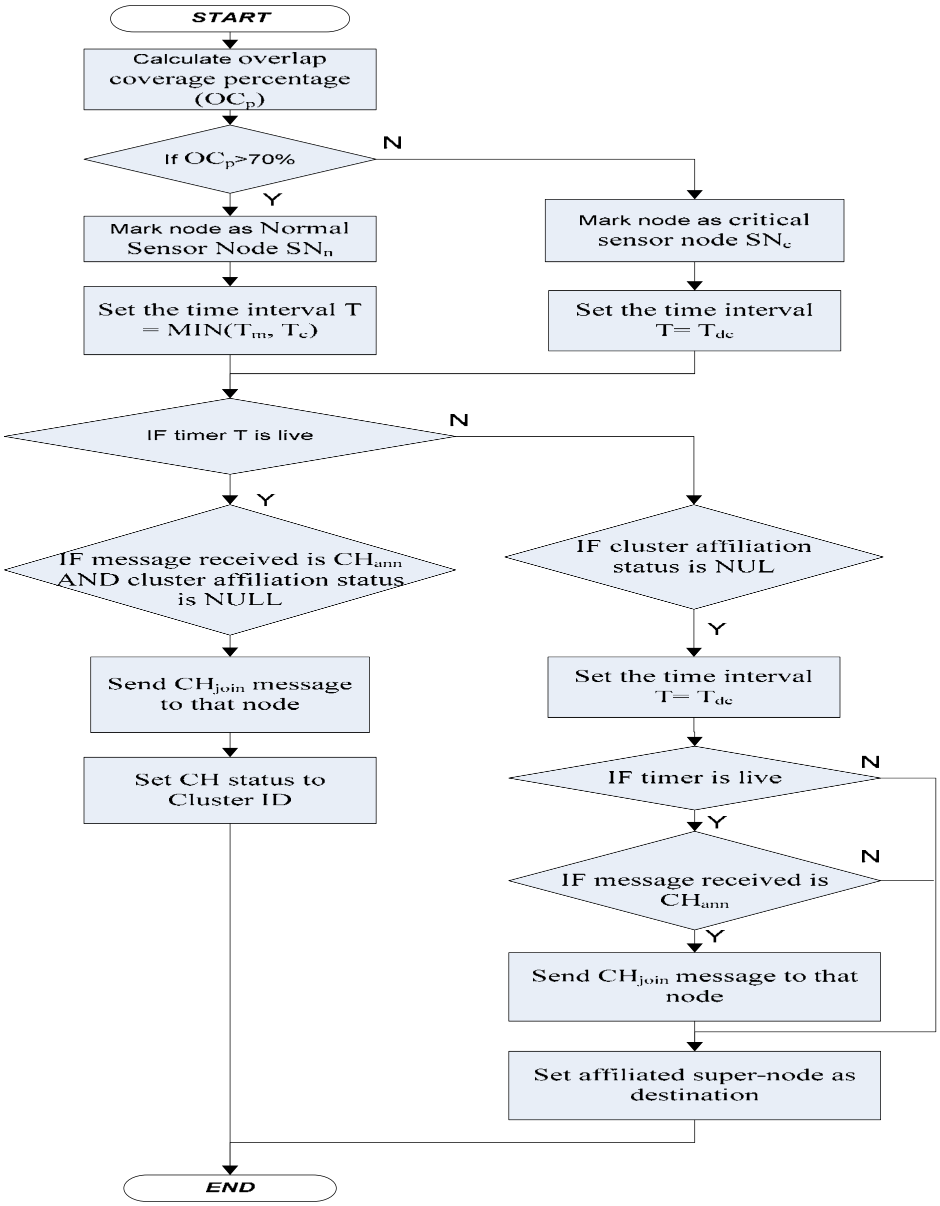


Figure 3.6 Detailed flow chart of cluster head selection algorithm in QoSEC protocol

i) Message types

The following is a brief description of message types, which are exchanged by sensor nodes during the CH selection process.

1. *Cluster Head Announcement (CH_{ann}):* It is a message sent by the node when its timer to declare itself as the CH expires and it is supposed to declare itself as the CH. It sends this message within communication range R_{comm} to let the other sensor nodes know about its new status. This message contains packet ID, node ID, cluster ID, Message type, and Time Sent T_{send}.

2. *Join Cluster Head (CH_{join}):* When any sensor node receives the CH_{ann} message from some other CH, that node sends this request message (CH_{join}) back to that CH as confirmation to associate itself with that CH. This message contains packet ID, node ID, cluster ID, Message type, and Time Sent T_{send}.

3. *Time period based on sensing range (T_s):* It is a self-message which is sent when the wait-time T_s expires. T_s is a time period based on the sensing range that is, half of the transmission range.

4. *Time period based on cluster head selection metric (T_m):* It is a self-message which is sent when wait-time T_m expires. T_m is a time period calculated based onthe cluster head selection metric.

5. *Time period based on communication delay (T_c):* It is a self-message which is sent when wait-time T_c expires. T_c is a time period based on the communication delay.

6. *Time period based on double communication delay (T_{dc}):* It is a self-message which is sent when wait-time T_{dc} expires. T_{dc} is a time period based o a double communication delay.

ii) **Algorithm 1:** Cluster Head Selection algorithm

Input: Sensor node coverage and position data

Output: Cluster Head (CH)

T_s	:	*Time period based on double communication delay (T_{dc}):*
T_m	:	*Time period based on cluster head selection metric*
T_c	:	*Time period based on communication delay*
T_{dc}	:	*Time period based on double communication delay*

CH	:	*Cluster Head*
CH_{ann}	:	*Cluster Head Announcement*
CH_{join}	:	*Join Cluster Head*
SN_n	:	*Sensor Node Normal*
SN_c	:	*Sensor Node Critical*
OC_p	:	*overlap coverage percentage*

begin

calculate overlap coverage percentage (OC_p) for each node with the help of equation (3.1) (Youssef et al. 2006) (see Figure 3.7).

$$OC_p = \frac{4\pi Cos^{-1}\left(\frac{d}{2r}\right) - Sin^{-1}\left(\frac{d}{2r}\right)}{\pi} \times 100 \quad (3.1)$$

where *d* is the distance between node A and B which can be calculated using the following equation (3.2)

$$d = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2} \quad (3.2)$$

if $OC_p > 70\%$ **then**

mark node as Normal Sensor Node SN_n

aet the time interval $T = MIN\ (T_m, T_c)$

else

mark node as Critical Sensor Node SN_n

set the time interval $T = T_{dc}$

end

if T is alive **then**
if *message received is* CH_{ann} *AND cluster affiliation status is NULL* **then**
send the CH_{join} message to that node
set the CH_{status} to the *cluster ID*
endif
else
if *cluster affiliation status* is *NULL* **then**
set the time interval $T = T_{dc}$
if *timer* is *alive* **then**
if message received is CH_{ann} **then**
send CH_{join} message to that node
else
set affiliated super node as destination
endif
endif
endif
end
end

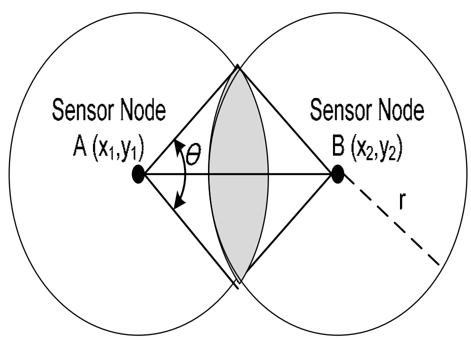


Figure 3.7 Overlapping coverage percentage of two sensor nodes

e) Setup Phase: Sensor Activation

Once a CH is selected for a specific cluster, all other nodes in the cluster are made known to the CH. Then, based on energy efficiency and coverage, the CH defines the sleep/wake schedule for the nodes in each round. The CH selects a subset of sensor nodes to perform the sensing task for the upcoming round, while the rest of the sensor nodes go to sleep. The selected active nodes should provide full coverage over the sensing area of the whole cluster during this communication round. For that purpose, the sleep/wake scheduling algorithm is proposed, by which when sensor nodes are found redundantly covering the same area, they will be alternatively made to sleep and wakeup. The following Algorithm 2 explains how each node finds its overlapping coverage and then performs the sleep/wake scheduling algorithm.

Algorithm 2: Sleep/Wake-Scheduling Algorithm

Input: Sensor Node overlapping coverage status

Output: sleep/wakeup schedule

SN_n : *Sensor Node Normal*

SN_c : *Sensor Node Critical*

OC_p : *Overlap coverage percentage*

begin

1.
2. Each node sends a message to the CH whether it is SN_n Sensor Node Normal or SN_c Sensor Node Critical as decided in Algorithm 1. The CH receives the coverage status (normal or critical node) from all the cluster member nodes. Subsequently, the CH considers this information during sleep/wake scheduling.
3. The CH considers the full coverage of the cluster sensing area and selects the subset of nodes as active nodes, which provide full coverage over the cluster sensing area.
4. For that, the CH sends the sleep message to the critical node SN_c and only considers normal nodes SN_n for active nodes. This is done to eliminate the redundant activation of sensor nodes in the clusters, which may happen when the activation of nodes is done irrespective of their coverage redundancy.
5. The CH sends the broadcast ACTIVATION message to inform its cluster

members about its sleep/wake schedule for the upcoming round.

6. Nodes on receipt of the ACTIVATION message, as per the sleep/wake schedule, go into sleep or active state for the next round.

end

3.1.4.2 Steady Phase

In the Steady-Phase, nodes perform the actual work of clustering, which is data gathering. The Steady-Phase is further divided into two sub-phases: intra-cluster communication (forwarding to the CH) and inter-cluster communication (forwarding to the sink).

a) Steady-Phase: TDMA Schedule.

When a CH has per-formed the sleep/wake scheduling, then the CH assigns time slots to all nodes in the cluster: these are, the slot times when the member cluster nodes can send the sensed data to the CH. The following Algorithm 3 explains how the TDMA scheduling is done.

Algorithm 3: TDMA Scheduling Algorithm

Input: ID's of the cluster member nodes

Output: TDMA schedule

TDMA : *Time division multiple access*

CH : *Cluster Head*

begin

1. The *CH* consults the sleep/wake schedule and arranges the *TDMA* schedule for the active nodes.
2. The *CH* sends the *TDMA* schedule (mentioning the slots in which each node can forward data to the *CH*) to every node in the cluster. The *CH* allocates equal time slots to all the sensor nodes in the cluster.

3. Upon receiving the *TDMA* schedule, every node goes into the sleep state and wakes up when its time slot to send data to the *CH* has been reached.
4. When the time slot to send data to the *CH* has been reached, cluster member nodes wakeup and send their data to the *CH* and sleep again till the next scheduled time slot.
5. The *CH* receives data from all senor nodes in its cluster in their assigned *TDMA* time slots. Data is aggregated and sent to the nearby *CH* and so on, until it reaches the super-node.

end

b) Steady-Phase: Intra-cluster Communication (Forwarding to CH)

This phase, involves how a sensor node should send sensed data to its CH. For intra-cluster communication, there exist two types of techniques: single-hop or multi-hop communications. In QoSEC, single hop communication is used. The CH receives data from all sensor nodes in its cluster in their assigned TDMA time slots using single-hop communication. The CH then aggregates the received data and sends the data to the nearby CH and so on, until it reaches the super-node.

c) Steady Phase: Inter-Cluster Communication (Forwarding to Sink)

For forwarding data to a sink or BS, protocols in the literature have used a very simple technique, that is, each CH sends data to the next CH, and so on. Ultimately, nodes nearest to the BS send data directly to the BS. However, the potential disadvantage of this technique is that the CHs nearest to the BS will over utilize their energy and will be depleted quickly. Therefore, an energy-hole (hot spot problem) is formed near the BS, and other nodes cannot forward their data to the BS. In WSNs, hot spot problems can significantly reduce the lifetime of the WSN. Consequently, with the existence of hot spots near the BS, nodes away from the BS cannot send data to the BS irrespective of their own high energy level.

With QoSEC, a WSN operates with two types of nodes: normal nodes and super nodes. Normal nodes perform the normal task of sensing, whilst super nodes act as local sinks since they have extra energy and communication capability. Data from normal nodes is collected by these super-nodes. Super-nodes send the collected data to the BS using either the multi-hop communication of the super-nodes or directly to the BS (see Figure 3.8).

i) Communication between CH and super-node

As described in the route update phase, after random node deployment, super-nodes with some random delays send a join message to all the nodes in its communication range. Nodes in the communication range maintain the route to the nearest super-node by a minimum hop count. As the CH has aggregated the data and it has information of the minimum hop path to the super-node, it sends data to the super-node using the minimum hop path.

ii) Communication between super-nodes

Finally, in inter-cluster communication, super-nodes that have the sensed data should now forward the data to the BS. In the route update phase, a super-node already has a minimum hop route to the BS. It sends data to the next hop as per the minimum hop entry or it will directly send the sensed data to the BS. In this way, through multi-hop super-node communication, the transmitted data will be able to reach the intended BS.

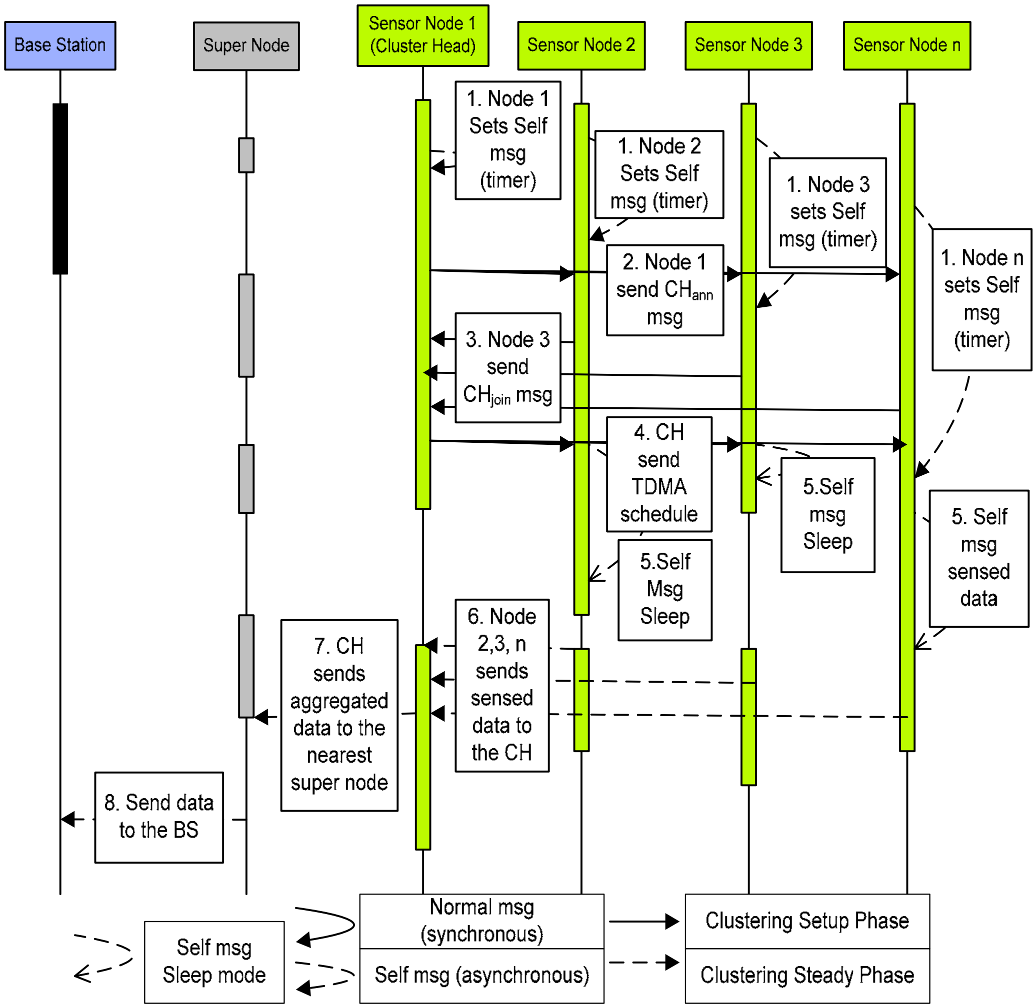


Figure 3.8 An event diagram for the interaction between sensor nodes, CH and BS during simulation

3.1.5 QoSEC: Summary

In this section, an energy-efficient clustering-based data gathering and routing technique for WSNs is proposed, called the QoS based Energy Efficient Clustering (QoSEC) protocol for Wireless Sensor Networks. With QoSEC, to achieve energy efficiency that prolongs network and coverage lifetime, some nodes with additional energy resources, termed as super-nodes, in addition to normal capability nodes, are deployed. Multi-hierarchy clustering is then implemented by placing super-nodes (acting as local sinks) at the top tier, cluster head (normal node) at the middle tier, and cluster member (normal node) at the lowest tier in the hierarchy. Clustering with normal sensor nodes is done by optimizing the network/coverage lifetime. Specifically, the improvement on energy efficiency and coverage lifetime is achieved through the implementation of the *cluster-head-selection* algorithm and *sleep/wake scheduling* algorithm. In summary, with QoSEC, the hot spot problem in WSNs is resolved, and most importantly the network/coverage lifetime of the WSN is prolonged.

3.2 QoS based Energy Efficient Sleep/wake (QoSES) Scheduling Protocol for WSNs

Many sleep/wake scheduling protocols found in literature use the same generic periodic sleep/wake interval for all the nodes in the network. In such sleep/wake scheduling protocols, variations in traffic load are not considered while assigning the sleep/wake schedule, rather the same generic periodic sleep/wake interval is assigned for all the nodes in the network. It results in accumulating delay at heavily loaded nodes. This increased delay is unacceptable for many emerging delay-sensitive WSN applications which require a fast response and real-time control. In this section, the delay-minimization problem in sleep/wake scheduling for event-driven sensor networks in delay-sensitive applications is considered. This work differs from previous approaches as it concentrates on minimizing end-to-end delay, while at the same time maintaining reasonable energy savings. In this section, the QoS based Energy Efficient Sleep/wake (QoSES) Scheduling Protocol for WSNs is proposed. To reduce delay, QoSES identifies nodes for different sleep/wake intervals according to their traffic load at three levels: a) Nodes with different sleep/wake schedule requirements according to their variable traffic loads based on their distance from the BS. b) Nodes with different sleep/wake schedule requirements according to their different traffic loads based on their topological importance in the network. c) Nodes with different sleep/wake schedules based on handling burst traffic in the proximity of an event occurrence node. It then assigns different active intervals to the nodes, according to variable traffic load requirements as defined by the node position in the network, by node topological importance and by handling burst traffic in the proximity of an event occurrence node. Using these heuristics, QoSES reduces end-to-end delay and maximizes the throughput by minimizing the congestion at nodes having heavy traffic loads. Simulations are carried out to evaluate the performance of the proposed protocol, by comparing its performance with S-MAC and anycast protocols. Simulation results demonstrate that the proposed protocol has significantly reduced the end-to-end delay, as well as improved the other QoS parameters, like average energy per packet, average delay, packet loss ratio, and throughput and coverage lifetime.

Following sections will explain in detail how QoSEC works.

3.2.1 Problem Description and Overview of Proposed Solution

Sleep/wake scheduling has been used to save energy and extend the network lifetime. Energy efficiency has an inherent tradeoff with delay, thus, generally in such sleep/wake scheduling strategies, maximization in network lifetime is achieved at the expense of an increase in delay. In many delay sensitive applications, where real time response is required, such delays could not be tolerated. In this section, the delay minimization problem in delay sensitive applications is addressed. The following is the description of the problem statement and how this problem is addressed.

In most sleep/wake schemes found in literature, all nodes have same generic sleep/wake schedule and each node makes autonomous wake up decisions without considering their neighbors in order to save energy. However, as WSNs use multi-hop communication, every node has one designated next-hop forwarding node in the neighborhood and to do the transmission, the sender node has to wait for the arrival of the wakeup time of the next hop forwarding node. Similarly, the next relay nodes have to wait for the wakeup interval of its next hop and so on, until the message finally reaches the BS. Consequently, due to the different wakeup intervals, additional delays are added at each hop, as the packet is forwarded along the path to the sink, because each node has to wait for its next-hop node to wake up before it can transmit the packet. All these delays at each hop, contribute to the final end-to-end delay of the packet. The increase in end-to-end delay is equal to the product of the number of intermediate forwarders times the length of the wakeup interval (Wei et al. 2002). This delay is unacceptable for many delay-sensitive applications, which require the event reporting delay to be small, such as in military surveillance, tsunami warning systems, seismic detection, biomedical health monitoring, hazardous environment sensing, and fire detection. The previous studies in sleep/wake scheduling are able to achieve energy efficiency, but do not consider these delays, which occur during sleep/wake scheduling.

In this section, the delay minimization problem is investigated and a sleep/wake scheduling scheme is proposed to optimally choose the variable wakeup interval of the nodes based on the nodes' dissimilar traffic loads. The first challenge for minimizing the expected end-to-end delay is to identify the different areas, where extra delay can be minimized. To achieve this, delay minimization at three levels is analyzed and addressed: the delay occurs because of traffic load at the nodes near the BS, delay occurs due to traffic load at the connectivity critical node, and delay occurs when dealing with burst traffic on occurrence of an event. It is indentified that delay is incurred because of following three reasons. Firstly, during sleep scheduling the wakeup requirement of nodes according to their position with respect to the BS is not considered. For instance, the node near the BS should have a greater wake period as it is doing the data forwarding jobs on behalf of other nodes and vice versa. Secondly, nodes have the same awake interval throughout the network. However, wakeup requirements in terms of forwarding traffic load are different for nodes than according to their topological importance in connectivity. For instance, nodes which are cut vertex nodes and happen to connect two distant parts of network, are expected to handle a much greater traffic load as compared to a normal load normal node. Thirdly, sleep scheduling is not adaptive to handle burst traffic in the occurrence of an event. In the occurrence of an event, the wake interval of the node and its neighborhood remain the same; however, in an event occurrence, the node is expected to receive more traffic.

In this section, these challenges are addressed and delay is minimized at these three levels. To reduce the delay, heuristics are used, which maximize the wakeup-up time in a scheduling period at three levels. The following is the brief description of how this problem is addressed.

Firstly, in a typical WSN architecture, the sink is located away from the sensor nodes or in the middle of the WSN. All the nodes use multi-hop communication to send their data to the BS. In such architecture, nodes near to the BS do a greater amount of data relaying than the nodes farther away from the system, thus the nodes near to the BS have to handle more traffic. But in sleep/wake scheduling, all nodes have the same generic sleep/wake schedule irrespective of their traffic load. This results in unnecessary delay at each hop because nodes have to wait for the wakeup interval of the next hop node, while data is routed along the multi-hop path to the BS. The problem gets worse when a packet approaches the nodes near to the BS and the whole network traffic has been routed through these nodes. This section proposes that this delay can be minimized by considering the fact that forwarding requirements of the nodes are different according to their distance from the BS. The forwarding job is directly related to the sleep/wake schedule because the more wake intervals there are the more efficient forwarding can be done with minimum delay by minimizing schedule misses. Nodes near the BS have to do more forwarding work as compared to the nodes farther away from the BS. Thus, to minimize delay in the proposed protocol, different sleep/wake intervals are proposed for the nodes relative to their distance from the BS. The nodes' wakeup time in the sleep/wake schedule is increased as the distance of the nodes from the BS decreases. Sensor nodes that are near the BS, are put into the sleep state with a lesser probability and sensor nodes that are farther away from the BS, are put into sleep state with a greater probability. Consequently, the wake intervals of the nodes increase as the nodes come closer to the BS in order to handle the extra delay. The notion behind this is that, nodes near the BS have to do a lot more data forwarding as compared to the nodes farther away from the BS; therefore, they should have higher wake intervals. This is to ensure that just because there is a high amount of traffic at the nodes near the BS, the packet does not suffer delay in forwarding.

Secondly, because of the multi-hop communication nature of WSNs, the nodes' role in routing is important. According to topology importance (traffic load), different nodes have different significance in the network. For instance, because of their topology position some nodes handle heavy traffic, while others handle less traffic. But generally, this aspect is not considered while devising sleep/wake schedules and all nodes are the given the same generic sleep/wake schedule, which can introduce unnecessary extra delays. The heavily loaded nodes are not available for data forwarding, because of having the same sleep/awake schedule as of other nodes. It is observed that these heavily loaded nodes can be determined by cut vertex, as the nodes which connect two segments of a network are expected to handle more traffic as they are the only nodes connecting two parts of the network. Thus, delay can be minimized by allocating sleep/wake schedules to the nodes according to their traffic load which is determined by the nodes' importance in connectivity. Giving higher wake intervals to heavily loaded nodes (connectivity critical nodes) ensures their availability when they are needed and giving lower wake intervals to lightly loaded nodes (less connectivity critical nodes) saves their energy.

Thirdly, when an event occurs at any particular area in a WSN, generic sleep/wake cycles of the nodes remain the same no matter what the frequency of the event detection is. It does not adapt according to the event occurrence in terms of changing their sleep/wake intervals. However, in the case of an event occurrence at a node, data forwarding requirements (traffic load) of the node itself and neighboring nodes are increased, as generally, event occurrences generate bursts of traffic. This data cannot be communicated in a timely manner with the generic sleep/wake intervals of the node. The reason is that, data has been generated at the event occurrence area, but neighborhood nodes did not adapt to the occasion to increase their sleep schedule to ensure a timely delivery of the event. For this problem, simple ideas of temporal and spatial dependency are used. Temporal dependency in this context refers that, when an event occurs in a sensing area of the node in one time slot, it is likely to occur in the proceeding time slots. Thus, if the nodes can adapt and change their sleep cycle, they can reduce the delay. Similarly, local dependency refers to the fact that, if an event occurrence is reported by a sensor node, there is the likelihood of the event occurring in its neighborhood nodes. Thus, nodes in the neighborhood of that node should adapt to that traffic burst and change their sleep cycles. Thus, based on temporal dependency the wake intervals of nodes where event occurrences take place are increased in the next time slots to handle expected traffic bursts in a timely manner. Similarly, based on the local dependency of the occurrence of the event in a node, its neighborhood nodes' wake intervals are also increased to handle expected traffic bursts in a timely manner. These measures can significantly reduce the delay.

3.2.2 Network Model and Assumptions

The WSN considered, consists of a large number of sensor nodes randomly deployed in a certain area (sensing field). After deployment, all the sensor nodes in the WSN sense the environment for a desired event. Sensor nodes wake up periodically to process the sensed data and propagate their own sensed data. As relay nodes, they also receive packet and forward data of other wireless sensor nodes. Sensed data is sent to the BS using multi-hop communication by doing data aggregation at each node. Thus, each node, in addition to the event sensing tasks it performs, has to collect, aggregate and relay data.

In the proposed protocol, the following assumptions are considered:

a. A sufficient number of sensor nodes are randomly deployed in a sensing area in such a way that the nodes can alternatively go into the sleep mode.
b. There is a BS located in the centre of a square sensing field and it has a sufficient energy resource.
c. The sensor nodes are static and have no mobility as is generally the case in most WSN applications.
d. The sensing, computation and communication capabilities of the nodes are the same and consume the same power to transmit one bit of data. Moreover, all sensor nodes have the same initial energy.
e. Links are bidirectional and symmetric, i.e., two nodes v1 and v2 can communicate with each other using the same transmission power level.
f. The Identificaion (ID) and the position of each sensor node are fixed after random deployment and all the sensor nodes know their location by using GPS or any other localization strategy. All the sensor nodes know the location of the BS.

g. Active sensors sense the event happening within their sensing range and then send the data to the next hop node and delete the buffer entries. Buffer size is small and generally data is communicated in the next available wakeup period.

It is important to note that in the proposed model no assumptions are made about 1) the heterogeneity of node dispersion in the field, 2) the network density or diameter, 3) the distribution of energy consumption among sensor nodes, 4) the proximity of querying observers, 5) the ability to communicate with the BS, or6) each node having a fixed communication range.

3.2.3 Preliminaries for QoSES

Following is the brief discussion ofthe overview of QoSES and outlines of data structures maintained at a sensor node in QoSES.

3.2.3.1 Data Structures at sensor nodes

In the proposed protocol, to assist the sleep scheduling/routing process each node maintains some data structures; use of these data structures will come in a later section. A brief description of these data structures is below:

- *Node ID.* Each sensor node is assigned a unique ID. It is used to identify the sender/receiver during the communication.

- *Next hop to sink:* Each node stores information about the next neighbor node, which can act as the next hop to the BS/sink. It consists of tuples in the form (SN-ID, hop-count, prev-node). Where SN-ID is the sensor node ID, hop-count is the number of hops leading to the BS and prev-node is the node ID of a 1-hop neighbor node that can lead to the BS using a minimum number of hops (part of the shortest path).

- *Region ID:* Each sensor node maintains the region ID by receiving messages from the BS. It is used to identify the node region number and is used in decision making in sleep/wake scheduling.

3.2.4 Description of the Proposed QoSES Protocol

The lifetime of an event driven WSN can be viewed, as consisting of two phases (see Figure 3.9): the setup phase and the operation phase. Generally, the operation phase is longer than the steady phase, and needs to be executed only once.

Following is the detailed explanation of the problem statement and the proposed QoSES protocol (see Figure 3.10).

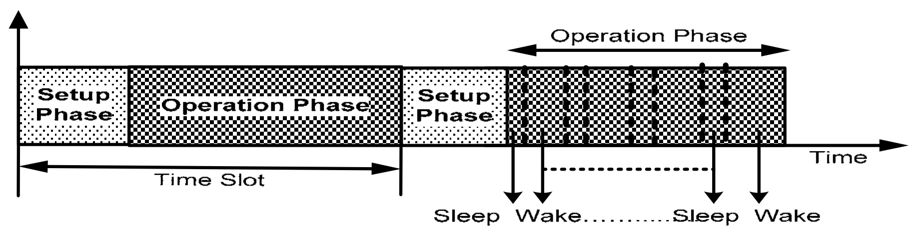


Figure 3.9 Life cycle of sensor network operations in the QoSES protocol

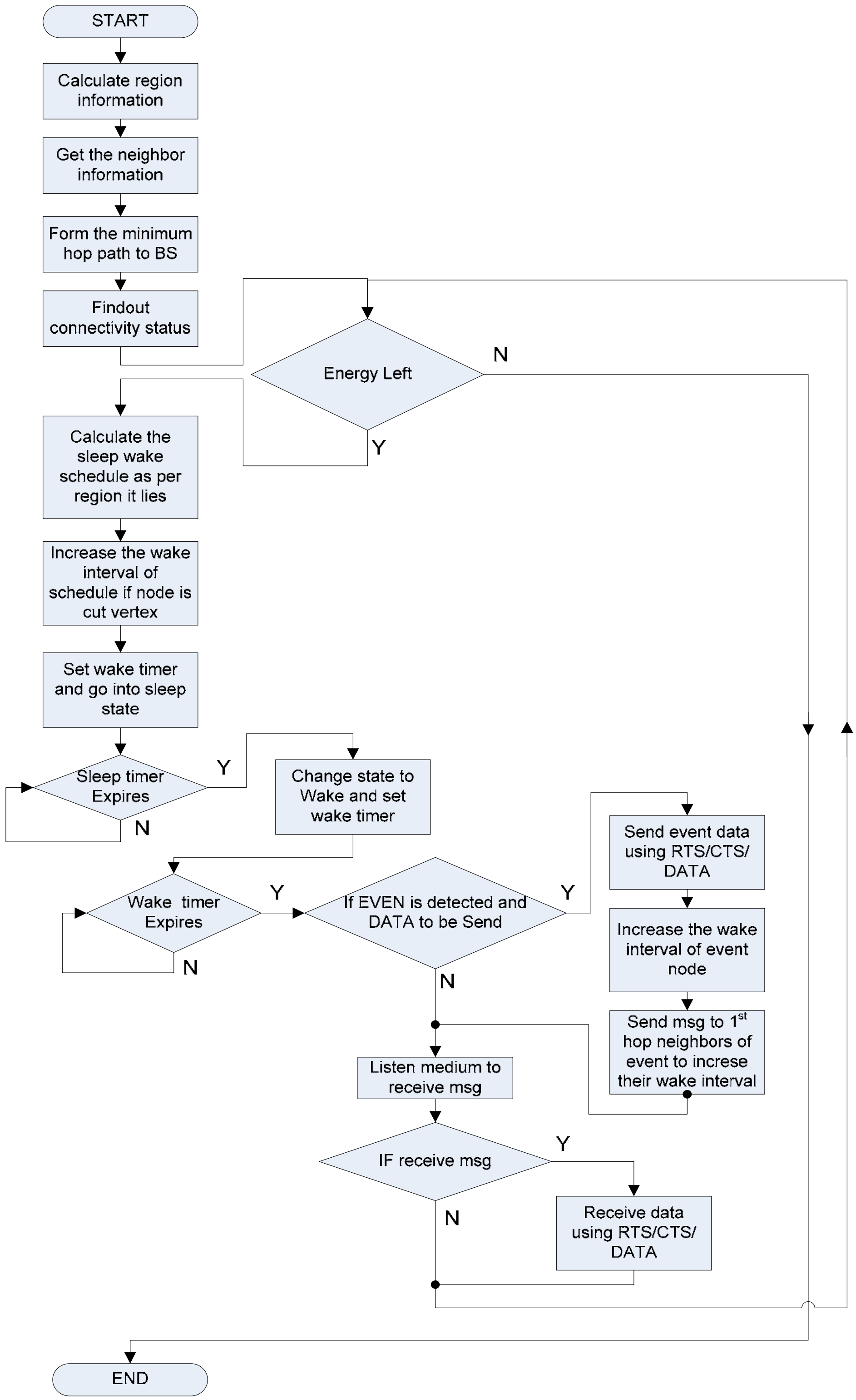


Figure 3.10 Detailed flow chart of the QoSES protocol.

3.2.4.1 Setup Phase:

During this phase all the sensor nodes in the whole WSN share control messages to know their neighbor nodes. It is further divided into two sub-phases (see Figure 3.11): initialization and route update (see Figure 3.12).

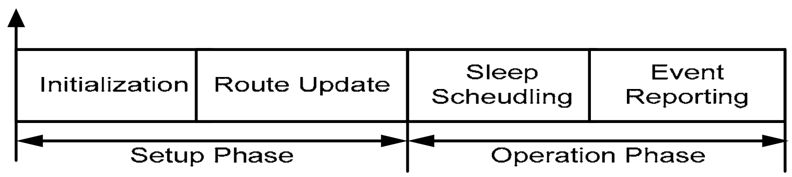


Figure 3.11 QoSES operations

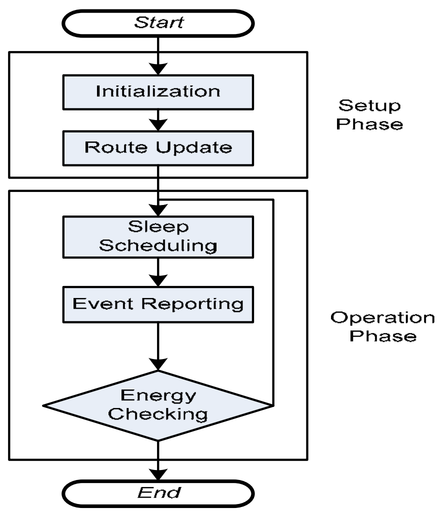


Figure 3.12 Flow chart of major phases and their interaction in the QoSES protocol

a) Setup Phase: Initialization

Each node determines its battery power and the location in which it lies. This is very important information which is used in the proceeding phases for important decision making like sleep/wake, route update and event report. Furthermore, the BS divides the network into different regions. The BS sends messages to all the nodes in the network using three different transmission powers:(TP) where $TP_1<TP_2<TP_3$. TP_1 determines region#1, TP_2 determines region#2 and TP3 determines region#3 (see Figure 3.13).

The BS first propagates a beacon message with the transmission power TP_1; the node receiving this message will mark its region status as region#1 and go into the sleep state. The BS propagates a control message with the transmission power TP_2. As nodes in region#1 are in the sleep state, they will not receive this message. All other nodes receiving this message will mark its location status as region#2 and go into the sleep state. Similarly, the rest of the nodes mark its region status as region#3 upon receiving message TP_3.

The region information is retained by the nodes and is used in succeeding phases. Afterwards, each node sends control messages to maintain their first hop neighbor information. Once the node has its neighbor's information, then it decides whether it is a connectivity critical node or not. This information is later used in making sleep scheduling decisions. For finding connectivity status, the cut vertex method is used as adopted in (Abbasi et al. 2007). Thus, a node finds out whether it is a cut vertex or not. If it is a cut vertex, it will mark itself as a connectivity critical node; otherwise, it will mark itself as normal node.

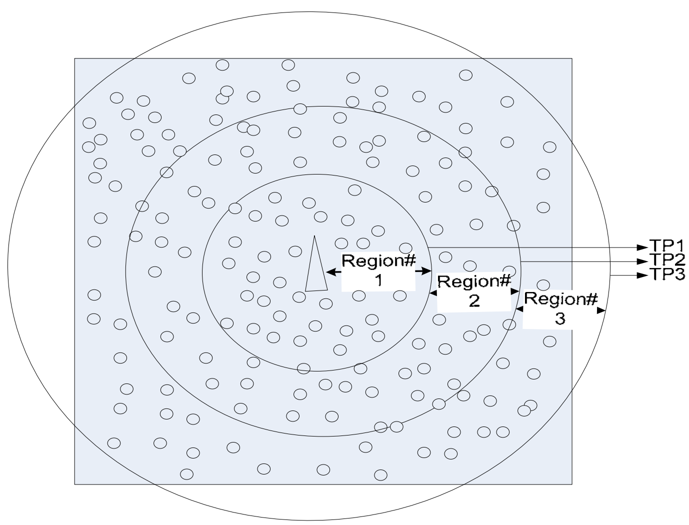


Figure 3.13 Dividing a WSN in to three regions.

b) Setup Phase: Route Update

In the route update phase, nodes form a multi-hop path of sensor nodes to forward data to the BS. In sensor nodes to update these routes, the BS first generates a route discovery message that is broadcast throughout the network. Upon receiving the broadcast message, each sensor node introduces a hop delay proportional to its cost before it forwards the route discovery message to nodes in its communication range. In this way, a message arrives at each node along the desired minimum cost path and each node has minimum hop count path to the BS.

In the route update phase, the BS generates a route discovery message with a hop count 0, which is broadcast throughout the network. Sensor nodes are the intended recipients for this message as these nodes will carry data to the BS using multi-hop communication. Each node upon receiving this broadcast message, updates its hop count value, i.e., change its value to the new value if the received hop count value is less than the previous hop count value; otherwise, retain the previous value. Before forwarding the route discovery message, each node increments the hop count and then broadcasts the message to nodes in its communication range. In this way, a message arrives at each node along the desired minimum cost path. In this way, after this phase, each node has a minimum hop count path to the BS. The hop count value is determined and the route information is later used in event reporting.

3.2.4.2 Operation Phase

In the operation phase, nodes do the sleep scheduling and perform the event reporting. The route information is determined and maintained in this phase to be used in later phases for event reporting to the BS.

a) Operation phase: Sleep/ wake Scheduling

In this phase, sleep/wake scheduling is done which is either based on the region of the sensor node or its connectivity importance. Based on these factors, each node determines its different sleep wake pattern. The sleep/wake scheduling algorithm addresses the delay minimization problem. It attempts to give longer sleep intervals to nodes, as per their traffic requirement. The traffic load of nodes differs according to the region they lie in, their connectivity importance and their proximity to the event occurrence. Based on these factors each node determines its sleep/wake pattern and accordingly switches between sleep/wake states (see Figure 3.14). In each sleep/wake cycle, nodes wakeup at the set time intervals, wait for events to occur, listen to the medium and sense/receive data.

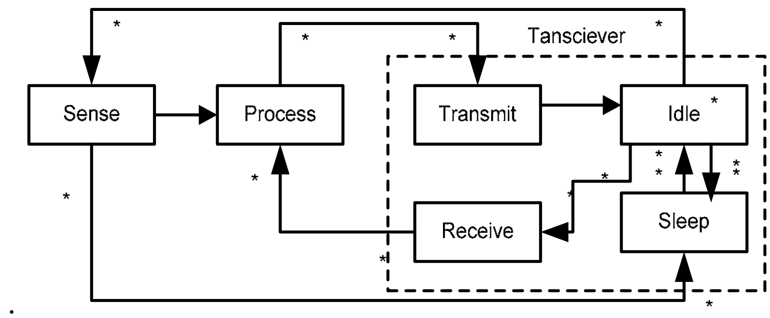


Figure 3.14 State transition in QoSES operations

Firstly, for sleep/wake scheduling, each node considers its region status in which it lies as determined in the initialization phase. The algorithm gives different wake intervals to different nodes according their region status. The algorithm gives three different wake intervals (WT) WT_1, WT_2 and WT_3 to nodes in region#1, region#2 and region#3, respectively where $WT_1<WT_2<WT_3$. Wake intervals of the node are inversely proportional to the distance from the BS, i.e., shorter the distance longer the wake interval (see Figure 3.15). It is done to make the wake interval adaptive to the traffic load of the nodes, as nodes near to the BS have a greater traffic load as compared to the nodes farther away from the BS. Thus, to accommodate this heavy traffic appropriately, nodes nearer to the BS should have greater wake intervals to avoid high end-to-end delay (see Figure 3.16). As at each hop, the packet has to wait for some time for the arrival of the wake interval of the next hop node and this wait time increases as the packet moves nearer to the BS. Because the traffic is coming from different hops to the same nodes near to the BS. Ultimately all this contributes in increasing the end-to-end delay. Such delays are very crucial in delay sensitive applications. Hence, assignment of wake intervals according to the traffic requirements of the nodes is proposed. It can decrease the end-to-end delay and help in the delay minimization problem.

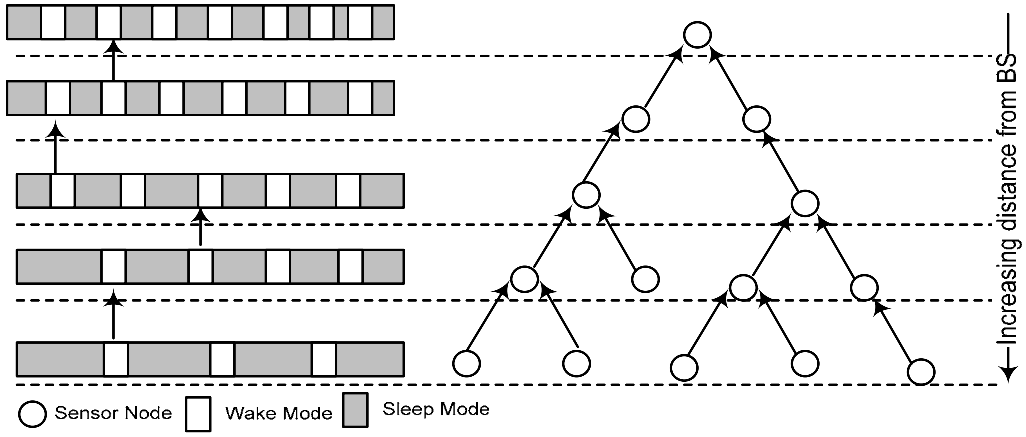


Figure 3.15 Assignment of different sleep/wake schedules to nodes according to their distance from the BS.

Secondly, as in the route update phase, each node knows about the connectivity status as determined in the route update phase. Connectivity status refers to the importance of node in maintaining the network topology. Thus, nodes which happen to be connectivity critical nodes should also have longer wake up intervals. Connectivity critical nodes are expected to have more traffic as these nodes are connecting two distant networks. If these nodes have the same wake interval as other nodes, then, because of the expected heavy traffic, the packets have to wait longer to be forwarded through these nodes. That is why the CH node assigns longer wake intervals to connectivity critical nodes.

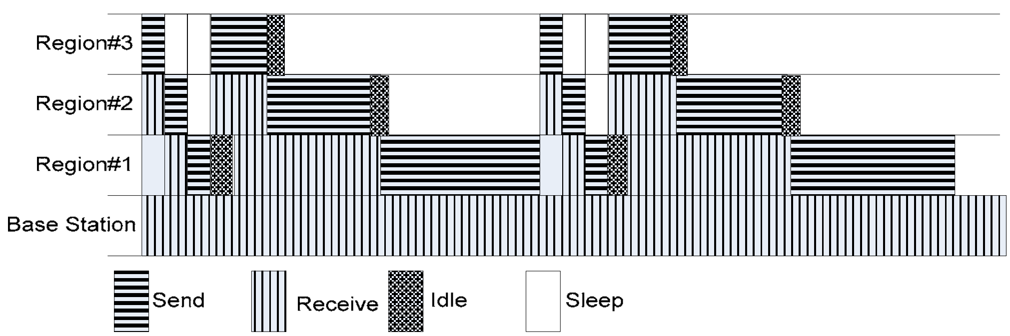


Figure 3.16 The path wakeup pattern of nodes in three regions.

At the end of this phase, each sensor node has different sleep/wake interval to the cluster member nodes. It assigns longer wake interval to nodes nearer to BS and it assign longer wake interval to the connectivity critical nodes to make sure the end-to-end delay is minimized.

The Following Algorithm 1 explains how each node does the sleep/wake scheduling.

Algorithm 1: sleep/wakeup scheduling algorithm

Input: WI_{normal} the Wake Interval Normal

Output: sleep/wakeup schedule

WI_{normal} : *Wake Interval Normal*

begin

check region status by each node

if *region status is region 1* **then**

  set wake interval to $3 \times WI_{normal}$

else

  if *region* is *region 2* **then**

    set wake interval to $2\times WI_{normal}$

  else

    set wake interval to WI_{normal}

  end

endif

check topology status by each node

if *topology status is critical* **then**

  set wake interval to $2 \times WI_{normal}$

else

  set wake interval to WI_{normal}

endif

follow up, by each sensor node, the determined sleep/wake schedule and sense the environment for the occurrence of the event for the specified interval defend by the sleep/wake schedule.

if event occurs **then**

set wake interval of event occurrence node to $3 \times WI_{normal}$

set wake interval first hop neighbors of event occurrence node to $2 \times WI_{normal}$ and send the message about the changed wake interval to the first hop neighbors

change of the sleep/wake interval by the first hop neighbors upon receipt of the updated sleep/wake schedule

wait of the event occurrence node for the arrival of the next scheduled slot determined by self timer before sending the sensed event data

send the data to the next hop neighbor using three way communication *RTS*, *CTS* and *DATA*.

endif

end

Consequently after this phase, each node has a minimum hop count path to the BS.

f) Operation phase: Event Reporting

The event reporting phase is responsible for forwarding data to the BS upon the occurrence of an event. In this phase, data is gathered from the sensor nodes and sent to the BS. When an event occurs near any node, that node will increase its wake interval for the proceeding time slot. Furthermore, it sends a message to its neighboring nodes to increase their wake intervals to handle the expected traffic burst. This is because of the fact that, when an event occurs near the node, there is a probability that the event will occur in the future as well (temporal dependency). Similarly, if an event occurs near a node there is likelihood of the occurrence of event in its neighborhood (spatial dependency). Both these cases will result in expected bursts traffic. Thus, wake intervals of node where the event occurs and nodes in its neighborhood are increased, to ensure the least delay in handling expected burst traffic (see Figure 3.17).

Algorithm 2: sleep/wakeup scheduling algorithm

Input: WI_{normal} the Wake Interval Normal

Output: sleep/wakeup schedule

begin

follow the determined sleep/wake schedule and sense the environment for the occurrence of the event for the specified interval defend by sleep/wake schedule and employed by the self timer

if event occurs **then**

> **set** wake interval of event occurrence node to 3 $*WI_{normal}$
>
> **set** wake interval of the first hop neighbors of event occurrence node to 2 $*WI_{normal}$ and send the message about the changed wake interval to the first hop neighbors
>
> **change** of the sleep/wake interval by the first hop neighbor upon receipt of the updated sleep/wake schedule.
>
> **wait** of the event occurrence node for the arrival of the next scheduled slot determined by the self timer before sending the sensed event data
>
> **send** the data to the next hop neighbor using three way communication RTS, CTS and DATA.
>
> **go** into sleep mode and wakes up in the next schedule interval: a node will wakeup in the next schedule slot determined by updated sleep/wake schedule.

endif

end

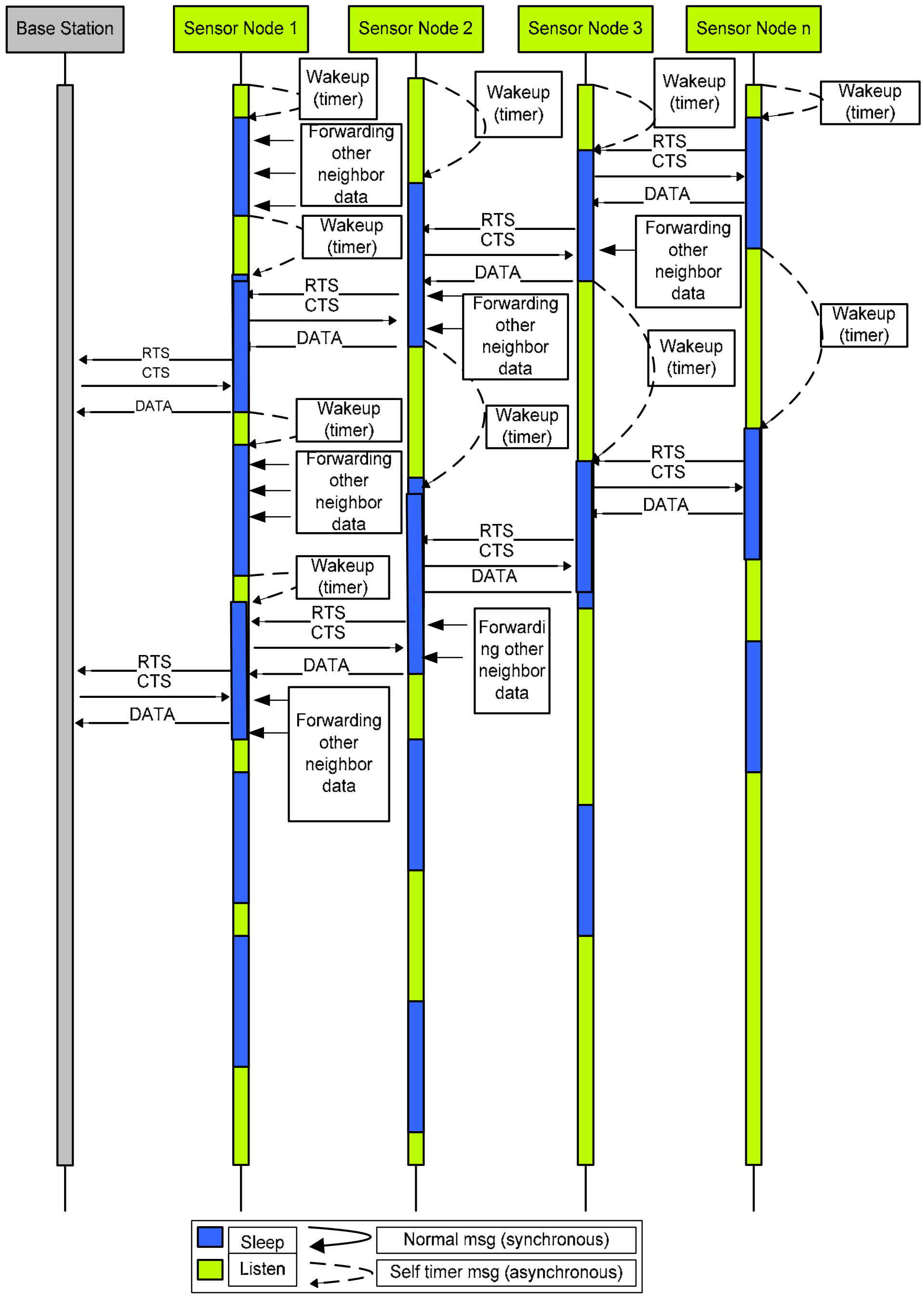


Figure 3.17 An event diagram for the interaction between sensor node and base station during simulation.

3.2.5 QoSES: Summary

In this section, the QoS based Energy Efficient Sleep/wake (QoSES) Scheduling Protocol for Wireless Sensor Networks is proposed. The proposed scheme is designed to have variable active duration of nodes according to their variable traffic load. The variable active durations are assigned to the nodes based on node distance from the BS, node topological importance, and occurrence of event in its vicinity. It will enable the nodes to handle the traffic appropriately, as nodes are dynamically assigned active durations according to their expected traffic load. It minimizes delay at the nodes nearer to the BS, nodes having critical topological positions, and nodes in the vicinity of the event occurrence. This ensures rapid dissemination of data to the BS, and hence reduces the end-to-end delay.

3.3 QoS based Energy Efficient Mobile Sink (QoSEM) based Routing Protocol for Clustered WSNs

In a WSN, sensor nodes near the static sink have to relay the data of the nodes away from the sink and as a result nodes near to the BS drain their energy very quickly. It results in network partitioning, as the nodes near the BS unable to relay the data to the BS, the problem is termed a hot spot problem. It can significantly limit the network lifetime. Contrary to the static sink, in recent years, the mobile sink approach has been used to address the hot spot problem but it increases end-to-end delay which is not acceptable for delay sensitive applications. Additionally, recently with an increase in multimedia WSN applications, QoS in data gathering has emerged as a critical issue. Sensor nodes generate different types of traffic which have got differentiated forwarding requirements based on their bandwidth and delay. To solve these three issues, namely: *hot spot problem*, *delay minimization* and *QoS assurance*, in this section, the QoS based Energy Efficient Mobile Sink (QoSEM) based Routing Protocol for Clustered WSNs is proposed. To address energy efficiency (hot spot problem) and high end-to-end delay problem, a combination of mobile and static sink is used for data gathering. Delay sensitive messages are sent through the static sink and delay tolerant message are sent through the mobile sink. Furthermore, to minimize delay in mobile sink data gathering, movement of the mobile sink is associated with the priority messages nodes need to send, i.e., in each cycle the mobile sink move across the whole network visiting different nodes based on the priority of the packets to be forwarded. To ensure QoS for different traffic types, prioritization of data is done based on message type and content. Data forwarding decisions are based on the priority of each packet, to ensure that message is fairly treated as per their distinct QoS requirements. In this way, the proposed protocol incurs less end-to-end delay, is energy efficient as well as able to ensure QoS. To evaluate the performance of the proposed strategy intensive simulations are carried out using OMNet++. Performance of the proposed strategy is compared with the contemporary static sink and mobile sink strategies. The simulation results demonstrate that QoSEM has prolonged the network and coverage lifetime, as well as improved the other QoS routing parameters, such as delay, packet loss ratio, and throughput.

The following sections will explain in detail the workings of the proposed QoSEC protocol.

3.3.1 Problem Description and Proposed Solution Overview

In the proposed protocol, three issues, namely: *hot spot problem*, *delay minimization* and *QoS assurance* are addressed. Following is the discussion of these issues and the overview of how the proposed mobile sink based protocol addresses these issues.

Firstly, collecting data from source sensor nodes to sink is a common and challenging task in most of the WSN applications. In a typical WSN architecture, the sink is static and sensor nodes use multi-hop communication to forward their data to the sink. Each sensor node sends data to neighboring sensor nodes, and so on until the data reaches the BS. Ultimately, nodes nearest to a BS send data directly to the BS. However, the potential disadvantage of this technique is that the sensors nearest to the BS may over utilize their energy, resulting in quick depletion of the energy of the sensor nodes nearest to the BS. This problem is termed as a *hot spot* or *energy-hole problem*. Hot spot problems refer to the phenomenon when the nodes near the sink quickly drain their energy on account of relaying the data of the nodes farther from the sink. As a result, although the nodes farther from the sink have significant energy, but their energy cannot be utilized as the nodes near the sink have depleted their energy. Consequently, data cannot be sent to the sink across hot spot or energy hole near the sink. The hot spot problem is an inherent phenomenon in WSNs with static sinks and can significantly decrease the lifetime of the overall network. The static nature of the sink is the main reason behind the hot spot problem, as the same nodes near the sink have to forward the data all the time. While this is a mjor drawback, the static sink approach comes with an obvious advantage; it involves a minimum end-to-end delay, as the nodes can send data at anytime using multi-hop communication.

Contrary to the static sink, in recent years, the mobile sink approach has been used to address the hot spot problem. It has attracted much research interest because of an increase in its potential in WSN applications and its potential to improve network performance. As stated earlier, mobile sinks have been identified to provide energy-efficient data gathering in WSNs. With the mobile sink approach, data forwarding to the mobile sink/BS is performed by moving the mobile sink across the whole WSN. During mobile sink movement, nodes send their data to the mobile sink when it comes in their vicinity either directly or indirectly using multi-hop communication. The mobile sink approach although by virtue of sink mobility gives considerable energy savings, the potential disadvantage of this technique is that as sensor nodes have to wait for the mobile sink to come in to their vicinity to do the communication, it results in an increasing end-to-end delay. This increase in end-to-end delay is not acceptable for delay sensitive applications. Notwithstanding the drawback of increased end-to-end delay, the mobile sink approach comes with an obvious advantage; it minimizes the hot spot problem and considerably improves the energy efficiency.

To address energy efficiency (hot spot problem) in the static sink approach and the high end-to-end delay in mobile sink approach, the proposed protocol uses a combination of the mobile and static sink for data gathering. Delay sensitive messages are sent through the static sink and delay tolerant messages are sent through the mobile sink. Furthermore, to minimize delay in mobile sink data gathering, movement of the mobile sink is associated with the priority messages of the nodes, i.e., in each cycle, the mobile sink moves across the whole network visiting different nodes based on priority of the packets to be forwarded. In this way, the proposed protocol incurs less end-to-end delay by virtue of the static sink and has got energy efficient data gathering by virtue of the mobile sink.

Secondly, with a recent increase in multimedia WSN applications, QoS based data gathering is emerging as a critical issue. Sensor nodes generate different types of traffic which have got their own delay and bandwidth requirements. They need some packet prioritization mechanism to ensure those messages are treated fairly as per their distinct QoS requirements. But generally, in packet forwarding protocols found in literature, all the packets have the same level of priority. It results in forwarding the potentially low priority message type (which can tolerate delay) first while making real time traffic (which cannot tolerate delay) wait, and all because the low priority message was generated earlier by the sensor node. To address this problem in the proposed protocol, incoming traffic is divided into different types, based on the different delay and bandwidth requirements of the traffic classes. The distinct classes are given different priorities where the real time traffic has highest priority and best effort text data has the lowest priority. The decision of packet forwarding is based on the priorities assigned to the packets. To be fair with all types of traffic, the priorities of all the packets increase with an increase in their queue wait; this ensure that low priority packets do not have to wait indefinitely

Thirdly, different messages generated by a sensor node have different levels of importance. Sending all messages generated by the sensor node all the way to the sink, involves usage of the scarce energy resources of the sensor node(s). This approach exhausts the energy resources of the WSNs, in propagating unimportant messages, while other more important messages wait for their turn to be forwarded. This situation arises simply because there is no priority mechanism involved in the packet forwarding phenomenon which can prioritize different types of message based on the importance of the message content. Consequently, with the proposed protocol, while forwarding the data, preference is given to the high priority critical data over low priority routine packets. Hence, efficiently utilizing the in hand energy resources of the sensor nodes for forwarding important data.

To solve these three issues, namely: hot spot problem, delay minimization and QoS assurance, the QoS based Energy Efficient Mobile Sink (QoSEM) based Routing Protocol for Clustered Wireless Sensor Networks is proposed. With QoSEM, energy efficiency is achieved by minimizing the hot spot by using mobile sinks that move across the network to gather data. As during the course of mobile sink movement, nodes near to the mobile sink change with each movement, hence the hot spot problem is minimized. Secondly, to deal with the extra delay which is generally encountered with the mobile sink approach, as nodes have to wait for the mobile sink to come into their neighborhood to send the data, a static sink is used in addition to the mobile sink. Delay sensitive data (high priority data) is forwarded to the static sink while the delay tolerant data (low priority data) is sent through the mobile sink. Thirdly, to address the QoS issue, incoming traffic is divided into different classes based the on message type and content. Prioritizations are assigned to the different types of traffic based on their QoS requirements (bandwidth, delay) determined by the message content and type. Message content refers to the importance of the data sent in a message (e.g. route messages, critical messages) and message type refers to the traffic type of the message (e.g. real time traffic, non-real time traffic). The distinct classes are given different priorities where the important messages and real time traffic has highest priority, whereas, routine messages and best effort data have lowest priority. These classes are assigned different priorities as per the delay and bandwidth requirements of each class. A priority queue is maintained to handle different priority packets and the highest priority packet is forwarded when the sensor node has its turn to send the data. Decisions about packet forwarding are based on the priorities assigned to the packets.

In this section, two approaches, static sink and mobile sink, are combined to achieve energy efficiency (hot spot problem) and improve QoS in clustered WSNs. In this section, the QoS based Energy Efficient Mobile Sink (QoSEM) based Routing Protocol for Clustered Wireless Sensor Networks is proposed. In addition to the usage of a combination of mobile and static sink approach, for ensuring QoS, in QoSEM, data is prioritized based on message content and type according to its delay and bandwidth requirements. Delay sensitive data is sent to the static sink using multi-hop communication, and for collection of delay tolerant data, mobile sinks move in the clustered WSN to collect sensed data from the super-nodes. From normal sensor nodes, data is collected by the CH, and super-nodes gather data from the CH in their vicinity. At a particular instance of time, super-nodes in the neighborhood of the mobile sink forward their data to the mobile sink. The rest of the nodes in the network having delay tolerant data, wait for their turn to become the mobile sink neighborhood. In this way, during sink movement, all the nodes in the network forward their delay tolerant data to the mobile sink, when the mobile sink comes in their neighborhood. Similarly, delay sensitive data is forwarded to the static sink by using multi-hop super-node communication. During data gathering, the mobile sink also maintains information about the queue weight (QW) (sum of priorities of the packets in the queue) of the super- nodes. In deciding a movement schedule, a mobile sink consults its maintained QW information about the priorities as well as residual energy of the super-nodes and searches for zones with higher QW and richer energy super-nodes. Consequently, mobile sinks based on the residual energy and queue weight of the super-nodes, devise the movement schedule by first visiting the super-nodes having a higher QW and residual energy. Mobile sink neighboring super-node nodes forward their data to the mobile sink using multi-hop or single hop super node communication. In this way, due to the mobility of the mobile sink, nodes near to the sink change after some time. Hence, no hot spot is created because during sink movement, responsibility of relaying the data is alternatively shared by different high energy super-nodes near the sink. It results in a balanced use of WSN energy and improves network lifetime. In addition, movement of the mobile sink near the highest energy nodes also avoids mini hot spots at the location where the mobile sink stays. Since in this way, the highest energy nodes become the first hop neighbor of the mobile sink and they will relay the data of the other nodes to the sink. Hence, avoiding lower energy nodes to become mobile sink neighbors and depleting their energy which results in an energy hole.

These measures ensure balanced use of energy among the nodes and can improve the network lifetime. Consequently, hot spots are not formed around the sink as the sink moves in the network; therefore, the nodes in the network are able to use their energy in balanced way.

3.3.2 Network Model and Assumptions

WSNs consist of a large number of sensor nodes randomly deployed in a certain area (sensing field). After deployment, all the sensor nodes in the WSN will form group clusters, and each cluster elects their CH in a distributed manner using local information. Each CH collects sensed data from other nodes in its cluster and transfers the aggregated data to the associated super- node, which sends it to the static/mobile sink using multi-hop super-node communication. The CH performs extra duties since it collects, aggregates, and relays data from/to its member nodes. To achieve balanced energy consumption, the role and responsibility as a CH is rotated over time among the member nodes in the cluster. In the proposed QoSEM protocol, the assumptions considered are same as described in section 3.1.2, additionally the following assumption is considered.

a. There is a static sink located at the center of the network. In addition, to the static sink there is one mobile sink which can move across the whole WSN. Both the static and mobile sinks have a sufficient energy resource. The sensor nodes (both super-nodes and normal nodes) are static and have no mobility as is generally the case in most WSN applications. Only mobile sinks have the ability to move (see Figure 3.18).

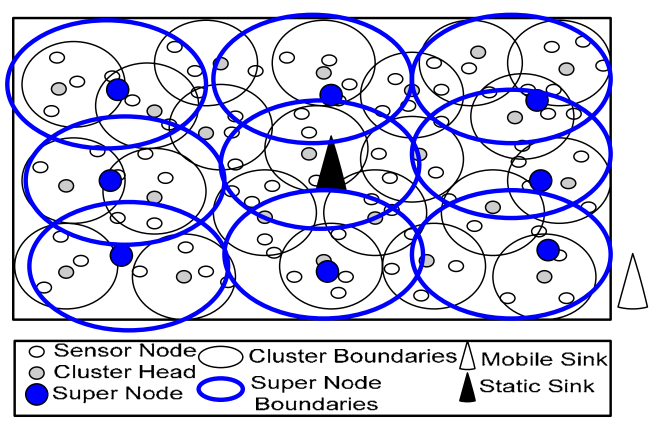


Figure 3.18 A typical QoSEM based data gathering in Wireless Sensor Network

3.3.3 Preliminaries of QoSEM

Following is the brief discussion of the overview of QoSEM and outlines of data structures maintained at a sensor node in QoSEM.

3.3.3.1 Data structure at a sensor node

To assist the routing process in QoSEM, each node maintains some data structure, the details of the data structures are same as described in section. 3.1.3.1.

3.3.4 Description of the Proposed QoSEM Protocol

In the proposed work, QoSEM has considered QoS and energy efficiency as the core design issues for data gathering in WSNs. With QoSEM, the data gathering process can be easily explained by dividing the protocol into two main phases: Setup Phase and Steady Phase (see Figure 3.19). Generally, Steady Phase takes a longer time than the Setup Phase. An explanation of QoSEM is given using these phases. The following is the detailed explanation of the proposed QoSEM protocol.

3.3.4.1 Setup Phase

During this phase, each node determines its residual energy (i.e., battery power left) and the location in which it lies. The information is later used in decision making, such as cluster-head selection, and sink movement. Values like residual energy, and distance to the BS are used as cost calculation in cluster formation. Clustering is done and the route to the static sink is also formed in this phase. The setup phase is further divided in to three sub-phases: Initialization, Clustering and Route update (see Figure 3.20).

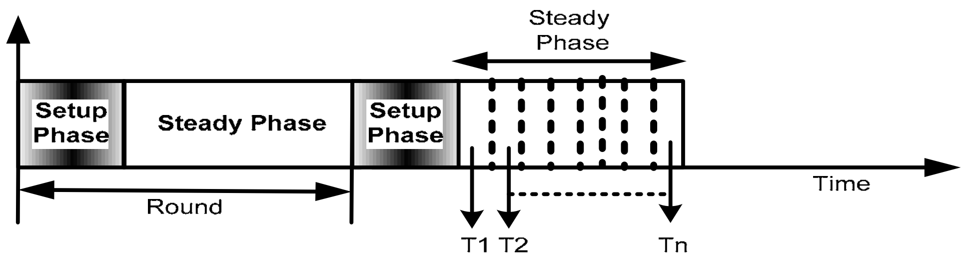


Figure 3.19 Round of sensor network operations in QoSEM protocol

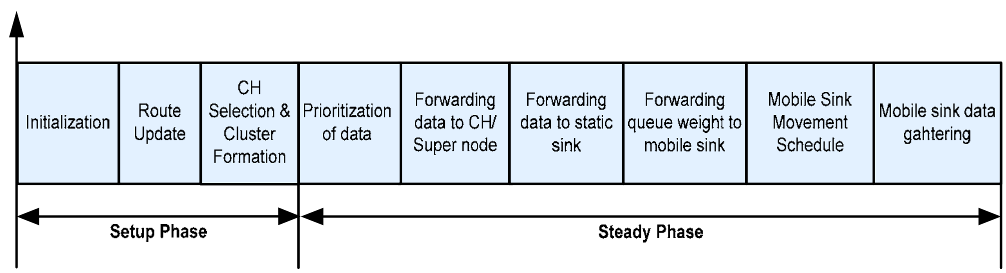


Figure 3.20 QoSEM operations

3.3.4.2 Setup Phase

During this phase, each node determines its residual energy (i.e., battery power left) and the location in which it lies. The information is later used in decision making, such as cluster-head selection, and sink movement. Values like residual energy, and distance to the BS are used as cost calculation in cluster formation. Clustering is done and the route to the static sink is also formed in this phase. The setup phase is further divided in to three sub-phases: Initialization, Clustering and Route update.

a) Setup Phase: Initialization

In this phase, after random deployment, each node determines its residual energy (i.e., battery power left) and the location in which it lies. The information is later used in decision making, and values like residual energy and distance to a BS are used as cost calculation in cluster formation and sink movement (see Figure 4.21).

b) Setup Phase: Clustering

In this phase, clustering (Hasbullah and Nazir 2010; Nazir and Hasbullah 2010) is done. All the nodes in a particular cluster choose a head node, known as a cluster head (CH). The sensor nodes must associate themselves with some of these CHs and become members of a local group (cluster). Details of how clustering is done is discussed in detail in section 3.1, interested readers may refer to the work in (Hasbullah and Nazir 2010; Nazir and Hasbullah 2010). Each group of nodes sends their data to the CH and then to the static/mobile sink.

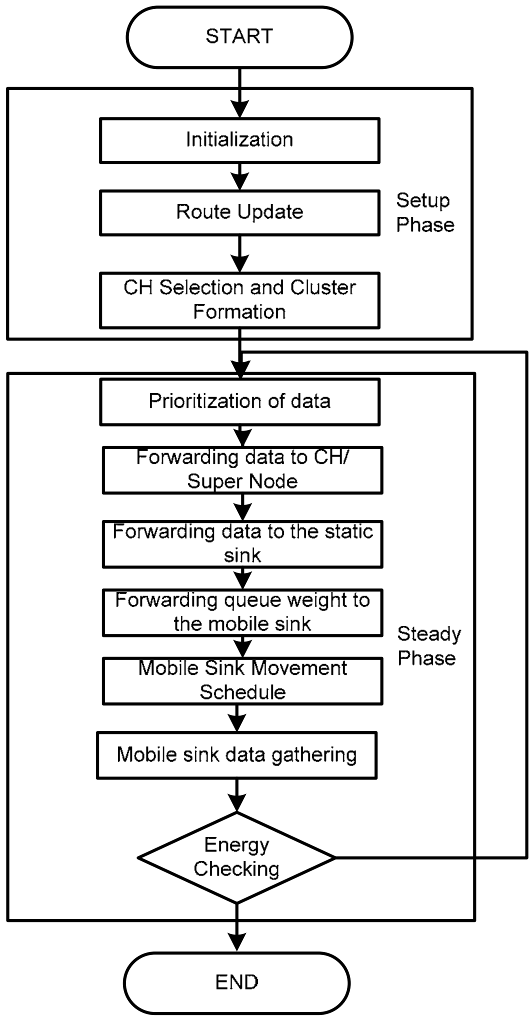


Figure 3.21 Flow chart of major phases and their interaction in the QoSEM protocol

c) Setup Phase: Route Update

One hierarchy above the CH, lies the super-nodes which act as local sinks and gather data from the CH. In the route update phase, CH nodes form a multi-hop path of CHs to forward data to the super-nodes. Afterwards, all the super-nodes form a multi-hop path of super-nodes to forward data to a static sink. For sensor super-node route formation, super nodes first generate a route discovery message that is broadcast within the locality of the super-node. Upon receiving the broadcast message, each sensor node computes a hop delay that is proportional to its cost before it forwards the route discovery message to nodes that are in the communication range. In this way, a message arrives at each node following the desired path of minimum cost. Similarly, for the super-node static sink route formation, the static sink generates a route discovery message (intended for super-nodes) that is broadcast- throughout the whole network. Upon receiving the broadcast message, each super-node computes a delay that is proportional to its cost before it forwards the route discovery message to nodes in the communication range. In this way, a message arrives at each super-node following the desired minimum cost path. As a result, each normal node and super-node can find their path to the super-node and static sink, respectively. Following are the details of how the route is maintained for the super-node and the static sink. Route update consists of two sub-phases: route update to super-node and route update to static sink**.**

i) Route Update to Super node

To forward data to super-nodes, The CH nodes use multi-hop CH communication. To develop these routes, each super-node generates a route discovery message with a hop count of 0. It is broadcast within the locality of these super-nodes. Normal nodes are the recipients of this message, as these nodes (which could later become CHs) will carry the received data to the super-node. Upon receiving this broadcast message, a normal node updates its hop count value, that is, changes its value to a new value, if the received hop counts value is less than the previous hop count value, otherwise, it retains the previous value. Before forwarding the route discovery message, each normal node increments the hop count and broadcasts the message to nodes in its communication range. In this way, a message arrives at each node following the desired minimum cost path. As a result, each normal node has a minimum hop count path to their respective super-node. Once the hop count value is determined, this route information is later used in inter-clustering communication.

ii) Route Update to static sink

In the route update phase, the static sink generates a route discovery message with a hop count of 0. It is broadcast throughout the network, where super-nodes are the intended recipients of these messages as they are responsible for carrying data to a static sink. Upon receiving this broadcast message, a super-node updates its hop count value, that is, changes its value to a new value, if the received hop count value is less than the previous hop count value, otherwise, it retains the previous value. Before forwarding the route discovery message, each super-node increments the hop count and broadcasts the message to nodes in its communication range. In this way, a message arrives at each super-node following the desired minimum cost path. As a result, each super-node has obtained a minimum hop count path to the static sink. Once the hop count value is determined, this route information is will later be used in forwarding data to the static sink.

After the route update phase, each normal node has a minimum hop count path to the super-node, and each super-node has a minimum hop count path to the static sink. The route information maintained will be used in later phases for inter-clustering communication

3.3.4.3 Steady Phase

In the steady phase, the proposed protocol performs the actual data gathering task. The steady phase can be further divided into the following sub-phases:

a) Steady Phase: Prioritization of data

In this phase, incoming traffic is prioritized based on message type and message content. Priorities assigned to the different messages are later used in the data forwarding decision, to maintain a reasonable QoS for each traffic class. Prioritization of data is done at two levels:

i) Prioritization based on message type

Increasing interest in multimedia applications in WSNs has made the QoS (quality of service) support an unavoidable task. A sensor node may have different types of sensors which gather different kinds of data. Based on the message type, the gathered sensed data can be divided into high priority real time traffic and low priority non-real time traffic. Both these types of data have different forwarding requirements. Real time traffic requires low latency and high reliability so that immediate action can be taken when required. Conversely, non real time traffic can generally tolerate delay. Needs for both of these types of traffic should be considered separately in making data forwarding decisions. For that, incoming traffic is prioritized based on its delay and bandwidth requirements and is divided into different traffic classes. Moreover each traffic class depicts a distinct traffic type with a specific delay and bandwidth requirements. Therefore, traffic classes are assigned different priorities. In QoSEM, to provide QoS, the incoming traffic is divided into three traffic classes: real time traffic, intermediate and non-real time traffic. Each traffic class is assigned different priorities with real time traffic highest priority, non-real time lowest priority and intermediate class priority lies between the two. According to their QoS requirements, different priorities are assigned to these traffic classes and the phenomenon of packet forwarding is dictated by these priorities.

ii) Prioritization based on message content

The data sensed by a sensor node can have different levels of importance. It is desirable for sensor network resources to be spent in disseminating packets carrying more important information. The information content is determined through the importance of the potential event needing to be reported. The rule used to determine information importance may vary depending upon the different applications. Once the importance of the message has been defined, different priorities are assigned to different messages based on the content of the packet. For instance, a high priority message might relate to a very high temperature in forest fire detection, which would require an urgent action. Contrary to this, a low message priority relates to a low importance packet (for example a packet carrying a routine normal temperature). Intermediate packets refer to messages whose information content lies between the high priority and low priority messages. Accordingly based on the importance of the message content, packets are divided into three different message classes (MC). MC 1 is the highest priority message, MC 2 is the intermediate priority message and MC 3 is the lowest priority message.

To ensure fairness and to prevent the packet from the low priority traffic class suffering a long starvation as the case may be when the higher priority process keeps on coming in the queue. The priority of the packet waiting in the queue increases by a factor of 1 at the start of every new time slot which helps to eliminate the obvious problem of starvation of the packets belonging to best effort and other low priority traffic. Hence, fairness is ensured to the low priority packets in terms of offering reasonable short delays.

Thus, at each sensor node different priorities are assigned to different types of traffic as mentioned above. A priority queue is maintained at each node, where packets are forwarded according to their priorities.

g) Steady Phase: Forwarding data to CH/super node

When data is sensed by the sensor node, it needs to be propagated to the sink. It is assigned the priority based on its importance according to its message type and content defined in previous phase. To handle these messages having different priorities, a priority queue is maintained at each of normal/CH/super nodes.

Whenever, data forwarding decisions are made at normal/CH/super nodes, the highest priority packet in the head of the priority queue is forwarded first.

As a priority queue has already been maintained at each sensor node, during this phase, each sensor node sends the sensed data to its respective elected CH according to the message priority. To do so, a normal node checks its priority queue for the highest priority packet and forwards the highest priority packet to the CH. The process of forwarding the highest priority packets at sensor nodes continues until the queue is empty. Similarly, a priority queue is also maintained at each CH. Each CH sends collected sensed data to its respective super-node according to its message priority. The CH checks its priority queue for the highest priority packet and forwards the highest priority packet to the super-node. The process of forwarding the highest priority packets at the CH continues until the queue becomes empty. To avoid long delays, super-nodes increment the priority of the packet waiting in the super-node queue by 1 at the end of every time slot. As data has already been accumulated by the super-node from its CH, the super node will calculate its Queue Weight (QW). This is the calculation of the weight of the queue, which is the sum of the priority of all the packets in the queue. The QW is used in a later phase for deciding mobile sink movement.

h) Steady Phase: Forwarding data to the static sink

The main shortcoming of the mobile sink approach is that, it maximizes end-to-end delay; this is because each sensor has to wait for a mobile sink to come into its vicinity to forward data. To avoid this drawback, QoSEM, in addition to the mobile sink, uses a static sink. In this phase, important data as dictated by its priority is sent to the static sink to minimize its end-to-end delay. Whereas, delay tolerant data is sent through the mobile sink, when it comes into its vicinity

For forwarding data to the sink, protocols found in the literature have used a very simple technique, that is, each CH sends data to the next CH, and so on. Ultimately, nodes nearest to the sink send data directly to the sink. Whereas, with QoSEM, the WSN operates with two types of nodes: normal nodes and super-nodes. Normal nodes perform the normal task of sensing, whilst super-nodes act as local sinks since they have extra energy and communication capability. Data from normal nodes is collected by these super-nodes. Super-nodes send the collected data to the sink using either multi-hop communication of the super-node or directly to the sink.

The movement of a mobile sink involves a delay which is not tolerable by the delay sensitive packets. Thus, to avoid this drawback of the mobile sink, in QoSEM, the super-node checks the priority queue of the node and based on the priority queue makes the decision whether to wait for the mobile sink or send the data to the static sink.

Following is a description of how super-nodes make decisions on whether to wait for the mobile sink or forward the data on to the static sink. It includes a description of the decision variables involved and details of the actual forwarding decision.

i) Decision Variables

The description of these decision variables used for determining whether to send data to the static sink or wait for a mobile sink is as presented below:

Queue Weight (QW): It is the calculation of the weight of the queue, which is the sum of the priorities of all the packets in the queue.

Max Queue Weight (MQW): Since the arrival rate of packets belonging to different traffic classes can be random, hence making the decision on the basis of queue length can affect QoS of packets belonging to different traffic classes. To ensure QoS in forwarding decisions (to send data to static sink or mobile sink), MQW is used.

ii) Forwarding Decision

Decisions on whether to send the data to the static sink or wait for the mobile sink are made when the following conditions become true (see Figure 3.22):

$$\textit{Queue Weight of the super-node} > \textit{Maximum Queue Weight (MQW)} \qquad (3.3)$$

i.e., when QW becomes greater than MQW, data is sent to the static sink to minimize delay.

Once the decision of sending the data to the static sink is made, the data is sent to the static sink by using the multi-hop communication of the super-nodes. In the route update phase, super-nodes already have a minimum hop route to the static sink. Thus, super-nodes send data to the next hop as per minimum hop entry or it will directly send the sensed data to the static sink. In this way, through the multi-hop super-node communication, the sensed data is forwarded to the static sink.

i) Steady Phase: Forwarding queue weight to the mobile sink

At the end of first and every subsequent cycle of mobile sink movement across the whole WSN, the mobile sink sends the request for queue weight across the WSN. Super-nodes are the intended recipients of this message. Each super-node, upon the receipt of this message from the mobile sink, calculates its queue weight. It is the calculation of the weight of the queue, which is the sum of the priority of all the packets in the queue. It sends the required information to the mobile sink, which will later devise its schedule based on this queue weight.

a) Steady Phase: Mobile Sink Movement Schedule

Queue weight information of all the super-nodes from the whole network has already been received by the mobile sink in the previous phase. In this phase, the mobile sink will prepare its schedule to move across the whole WSN to collect data from the available super-nodes. Movement of the mobile sink in the first movement cycle is according to a predefined position. But, to provide fair QoS and ensure balanced use of energy, the movement of the mobile sink in second and subsequent cycles is dictated by the residual energy and priority status (queue weight) of super nodes in the network.

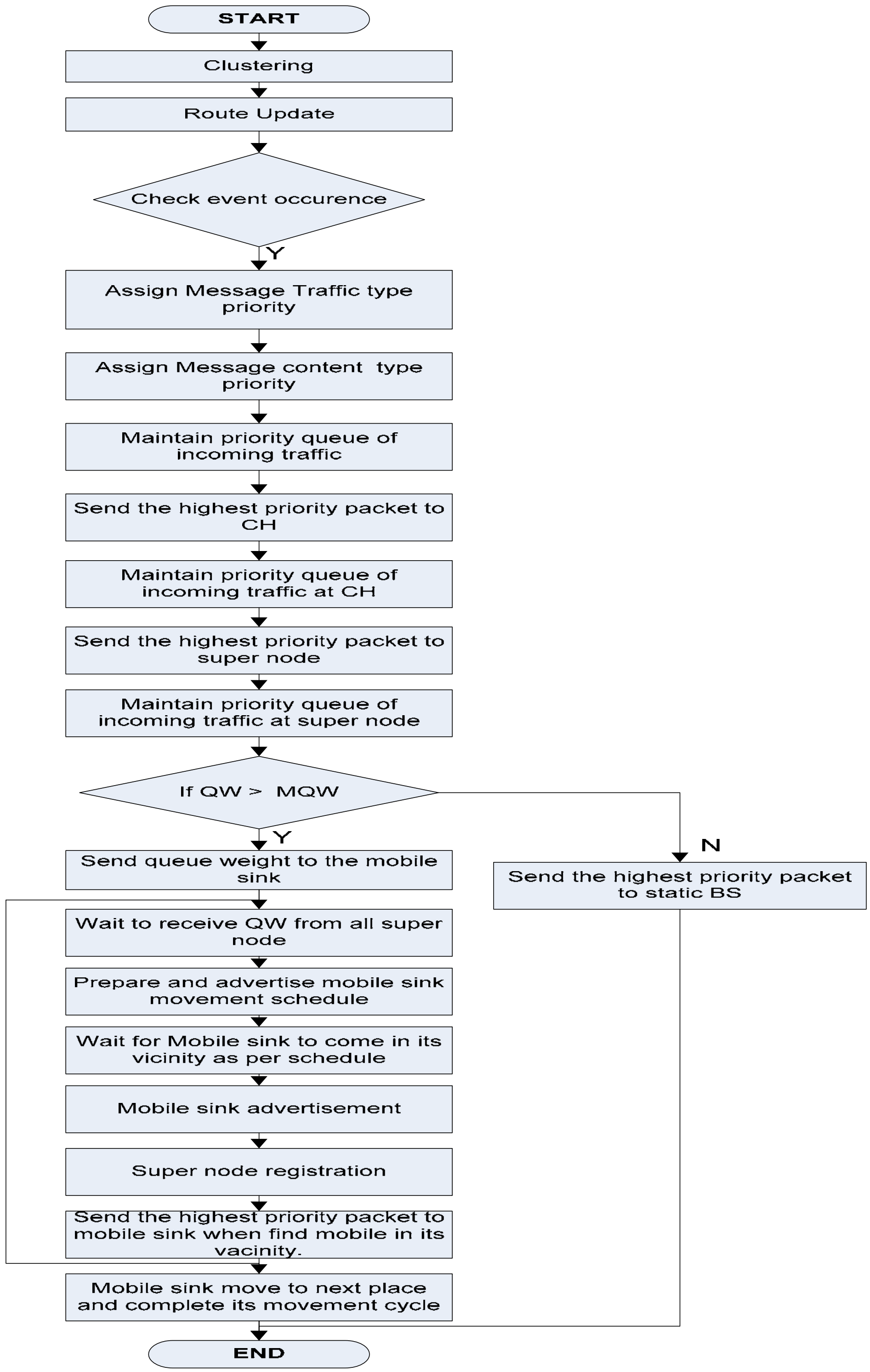


Figure 3.22 Detailed flow chart of cluster head selection algorithm in the QoSEM protocol

i) Devising movement schedule

In devising a movement schedule, to achieve energy efficiency and QoS, the mobile sink favors the super-node with a higher energy level and a higher queue weight, and therefore a scan is made through the network. The mobile sink arranges the movement plan and visits each super-node based on their queue weight and residual from the highest to the lowest. It disseminates the movement plan to all the nodes in the WSN. It helps to achieve QoS and prolongs the network lifetime, as super-nodes in the mobile sink vicinity are always the high energy nodes. As these nodes have to relay the data of the nodes farther away from the mobile sink. Furthermore, with regular sink movement, nodes near to the sink change with each movement. Thus, chances of formation of hop spot problem or energy hole is minimized. Algorithm 3 states the sink movement algorithm.

Algorithm 1: Sink Movement Algorithm

Input: Queue Weight and residual energy of super nodes.

Output: Sink movement schedule.

QW : *Queue Weight*

begin

if movement cycle is first **then**

  move in the network according to the predefined locations.

else

  sort the QW and residual energy of different super-nodes in descending order.

  arrange the movement schedule for each super-node based on its queue weight and residual energy from the highest to the lowest.

  broadcast the movement schedule across the whole WSN.

endif

end

ii) Advertising mobile sink movement schedule

Once a decision has been made, the mobile sink advertises the movement schedule across the whole WSN. Super-nodes are the intended recipients of this advertisement message. Upon the receipt of the mobile sink movement advertisement message, the super-node arranges the sleep/wake schedule accordingly, to make sure that, at the time of the visit of the mobile sink, the super-node will be in the wake state.

At the end of each cycle of movement across the whole WSN, the mobile sink sends the request for queue weight the super-nodes are the intended recipients of this message. Each super-node, upon the receipt of this message from the mobile sink, calculates its queue weight, which is the sum of the priority of all the packets in the queue. It then sends the required information to the mobile sink.

b) Steady Phase: Mobile sink data gathering

In this phase, the mobile sink moves across the whole WSN to gather the sensed data from the super-nodes. The schedule on how the mobile sink visits all the super-nodes has already been defined in the previous phase. As per the movement schedule, the mobile sink visits each super-node, sends the advertisement, defines the TDMA schedule for the super-nodes, and gathers the data from super-nodes. Following is a description of the steps involved in this phase.

i) Mobile Sink Advertisement and Super Node Registration

When the mobile sinks reaches any new place during its mobility, it must inform the super-nodes in its new neighborhood about its presence there; the nodes in its neighborhood can then send the sensed data to the sink. To do this, upon reaching any new destination, the mobile sink broadcasts a beacon advertisement message which contains the location information of the mobile sink and information of its moving velocity V. All the super-nodes which receive the mobile sink's beacon message, respond by sending a registration message to the mobile sink. During this phase, when the mobile sink enters the valid dissemination range of the super-nodes, the super-nodes intercept the beacon message of the mobile sink. By doing so, the super-nodes come to know that the mobile sink is in their communication range and can send their data to the sink. If the super-nodes have already sent their data to the static sink or the mobile sink in some of its previous movements in the current cycle, the super-nodes will ignore this message. Otherwise, the super-nodes respond to the mobile sink by sending a registration request to the mobile, they will send the $SN_{register}$ message. The mobile sink waits for a time interval (communication delay) to receive the SH_{ack} from the mobile sink.

Following is Algorithm 2 which is used for the mobile sink advertisement and super-node registration.

Algorithm 2: Mobile sink advertisement and super node registration Algorithm

---

Input: Reception of $SN_{register}$ messages from the super nodes.

Output: Registered super-node list.

T_i	:	*Time period based on communication delay*
$SN_{register}$	:	*Super node Announcement*
SN_{ack}	:	*Super node Acknowledgment.*

begin

Mobile sink sends the beacon message and sets the time interval ‘T_i’ based on the communication delay

Super-node receives the beacon message from the mobile sink

if super node has data to send **then**

  send $SN_{register}$ to the mobile sink

  set the time interval ‘T_i’ based on the communication delay and wait for SN_{ack}

end if

if mobile sink receives message from super-node within the time period 'T_i' **then**

if message received is $SN_{register}$ **then**

send SN_{ack} message to that super node

include the node in the registered super-node list.

endif

else

move the mobile sink to the next place, based on the sink movement schedule defined by algorithm 1.

endif

end

ii) TDMA Scheduling

Once the mobile sink has registered all the super-nodes in its current neighborhood, it carries out its responsibility as the mobile sink. It assigns the time slots to all the registered super-nodes, i.e., slots when the registered super-nodes can send the sensed data to the mobile sink. Consequently, in this phase, the mobile sink devises and sends the TDMA schedule to the registered super-nodes using the following Algorithm 3.

Algorithm 3: TDMA schedule Algorithm

Input: Registered super-node list.

Output: Super-node TDMA schedule.

TDMA : *Time division multiple access*

begin

1. Mobile sink checks the registered super-node list defined by algorithm 2 and arranges the time slot for the registered super-nodes accordingly.
2. **send** the super-node *TDMA* schedule T*DMA* to the registered super-nodes.
3. **wait** for the sensed data from the registered super-nodes
4. **receive** the sensed data from the super-nodes in the assigned time slots.

end

iii) Forwarding to mobile Sink

In this phase, each super node uses single hop or multi-hop communication to forward the sensed data to the mobile sink (see Figure 3.23). It involves how super-nodes send sensed data to the mobile sink in their allocated time slot. Thus, for the agreed upon TDMA time slot (discussed in previous section), the super-node sends the sensed data to the mobile sink using single or multi hop communication. For forwarding the packets, the super-node checks its priority queue and forwards the highest priority packets to the mobile sink. This process of forwarding the highest priority packets continues when any of the conditions mentioned in the above “Sink Movement Decision” phase becomes true. When any of switching condition becomes true, the mobile sink moves to the next destination and repeats the same process of forwarding the packet and continues at the next position. To avoid long delays, super-nodes increment the priority of the packet waiting in the super-node queue by 1 at the end of every time slot. With sensed data, super-nodes also piggy back the residual energy information to the mobile sink. At the mobile sink, this residual energy information is maintained in the Super-Node Residual Energy Table (SNRET). Residual energy information is used later in the next phase in making sink movement decisions.

iv) Sink Next Movement

The schedule of how to visit different super-nodes in the network has already been defined in the phase “Sink Movement Decision”. In this phase, the mobile sink makes the decision when to switch/move to the next super-node.

Following is the description of the decision variables involved in devising the mobile sink’s next movement decision and details of how the actual movement/switching decision is made.

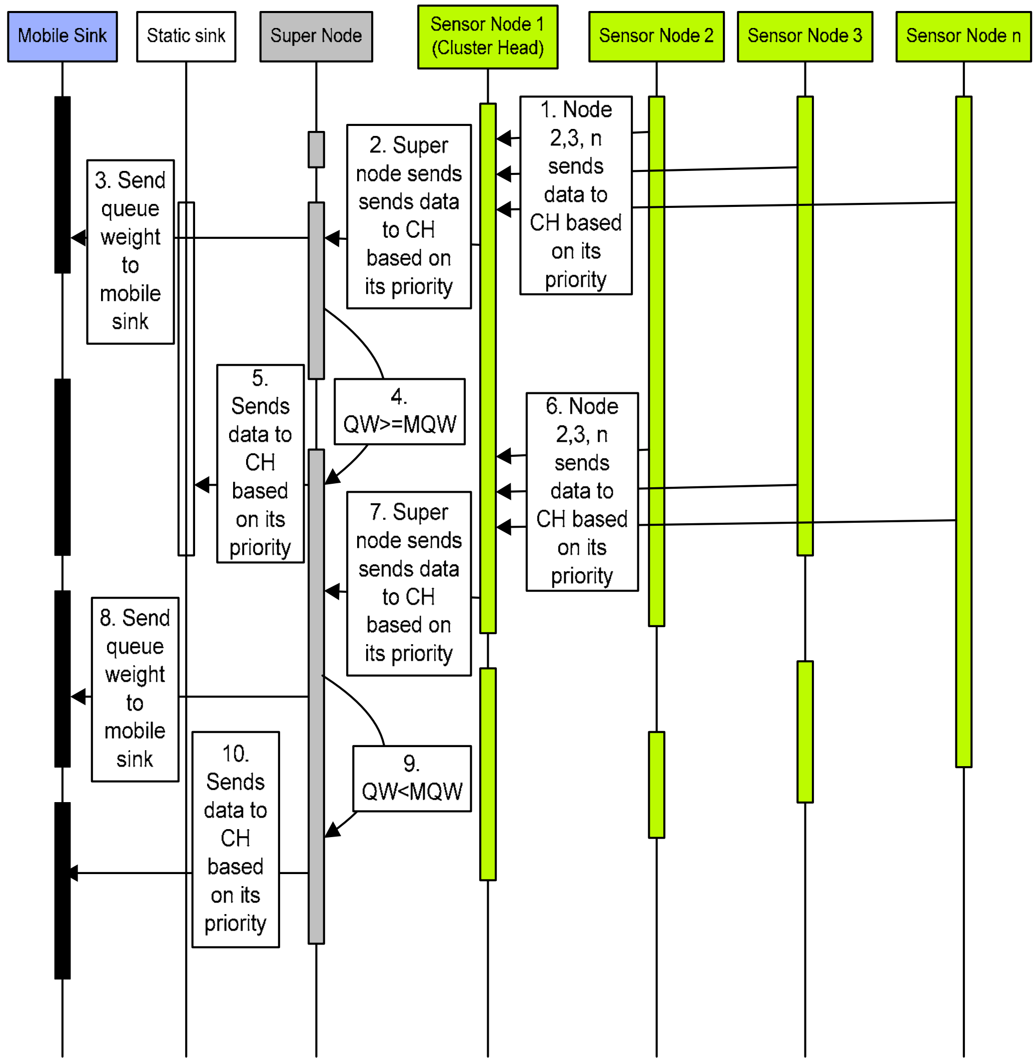


Figure 3.23 An event diagram for the interaction between sensor nodes, CH and BS during simulation

1) Decision Variables

Decision variables are used for determining the movement schedule of the mobile sink and the time instances when the mobile sink should switch from one super-node to another during its movement. The description of these decision variables is as presented below:

Queue Weight (QW): It is the calculation of the weight of the queue, which is the sum of the priority of all the packets in the queue.

Max Queue Weight (MQW): Since the arrival rate of the packets belonging to different traffic categories can be random and making decisions on the basis of queue length can affect the QoS of packets belonging to different traffic classes; to ensure QoS in switching from one super-node to another, the Maximum Queue Weight (MQW) is used. When QW becomes greater than MQW the mobile sink must try to switch from the current super-node to the next super-node.

Predictability Factor (P): This factor is used in calculating the Time Commitment (TC). It is calculated by estimating the average packet length of the traffic in the queue of a super-node.

Time commitment (TC): This variable sets the minimum interval that the mobile sink will spend outside any super-node. It is calculated on the basis of the queue length and Predictability factor (P) of the other super-node, i.e., it provides an estimate of the time to exhaust the outgoing queue. It limits the mobile sink from addressing a current super-node indefinitely and switches to the next super-node when its TC has expired.

Max time-share (MTS): This is the maximum time a mobile sink can spend with a super-node. It is the time limit set so that, the mobile sink does not address a current super-node indefinitely, and switches to another super-node when the MTS expires.

2) Switching Decision

The decision of when to switch the mobile sink to the next super-node is based on the following conditions using the above mentioned decision parameter. Consider a mobile sink with $X_1, X_2, \ldots, X_n$ as its super-node points, where it stays across the network, X_1 is the first super-node to be visited and X_n is last to be visited. Suppose X_1 and X_2 are any consecutive super-node points, the mobile sink is at X_1 and X_2 is the next point to be visited. According to QoSEM, the mobile sink will move from X_1 to X_2 when any of the following conditions become true:

1. Time in X_1 has become greater than the Maximum Time Share (MTS).

2. Time Commitment (TC) for super-node X_2 becomes 0.

3. Queue Length for super-node X_1 becomes 0.

4. Queue Weight of super-node X_2 is greater than the Maximum Queue Weight (MQW).

If any of the above conditions become true, then the mobile sink before moving to the X_2 calculates its TC for X_1 which will be:

$$Time\ commitment\ (TC) = \sum_{i=1}^{n}(Queue\ Length\ of\ super\ node\ X_i \times Predictability\ Function\ (P)\ of\ super\ node\ Xi\ ) \tag{3.4}$$

During its movement across the network when the mobile sink reaches the last movement place, it sends the cycle completion message across the whole network. All the super-nodes which receive this message respond back by sending their respective queue weight. Afterwards, steps from phase IV “Forwarding the queue weight to the mobile sink” till the end (explained above), are repeated for the next and preceding mobile sink movements for each movement cycle.

3.3.5 QoSEM: Summary

In this section, an energy-efficient and QoS based routing technique for WSNs has been proposed, called QoS based Energy Efficient Mobile Sink (QoSEM) based Routing Protocol for Clustered Wireless Sensor Networks. With QoSEM, to ensure QoS for different traffic types, prioritization of data is performed based on message type and content. Data forwarding decisions are based on the priority of each packet to ensure that messages are fairly treated as per their distinct QoS requirement. To address energy efficiency (hot spot problem) and high end-to-end delay problem, a combination of mobile and static sinks is used for data gathering. Delay sensitive messages are sent through the static sink and delay tolerant messages are sent through the mobile sink. Furthermore, to minimize delay in mobile sink data gathering, movement of the mobile sink is associated with the priority messages that nodes need to send, i.e., in each cycle, the mobile sink moves across the whole network visiting different nodes based on priority of the packets to be forwarded. In this way, the proposed protocol incurs less end-to-end delay, is energy efficient and able to ensure QoS. In summary, with QoSEM, the hot spot problems in WSNs are resolved with the increased in network lifetime and most importantly delay and QoS is improved.

3.4 Summary

This chapter has discussed the proposed protocols in detail, namely: QoS based Energy Efficient Clustering (QoSEC) for WSNs, QoS based Energy Efficient Sleep/wake (QoSES) Scheduling for WSNs, and QoS based Energy Efficient Mobile Sink (QoSEM) based Routing Protocol for Clustered WSNs. An explanation of each protocol (QoSEC, QoSES, and QoSEM) with a detailed statement of problem, network model, proposed protocol overview, data structures used at the sensor node, and finally a detailed description of the proposed protocol was presented.

CHAPTER 4

SIMULATION RESULTS, ANALYSIS AND DISCUSSION

This chapter starts with the types of network simulators (discrete event and continuous time). The ext section will present important points to be considered for selection criteria of WSN simulator. An overview of OMNeT++, important features of its framework and WSN simulation designs are discussed. The rest of the chapter will discuss the simulation results and critical analysis of the three proposed protocols, QoSEC, QoSES and QoSEM, in terms of performance metrics like average energy per packet, average delay per packet, packet loss ratio, throughput, network lifetime and coverage lifetime.

4.1 Network Simulator

Network simulation is a technique which involves programs that model the behavior of a real work network by modeling and simulating communication between different network entities. There are two types of simulators: the discrete event simulator and the time continuous simulator. To follow is the brief discussion of these two simulators.

4.1.1 Discrete Event Simulator

A discrete event simulator changes its state at discrete time instances. Change of state is referred to as an event and it is assumed that nothing interesting (no state change) happened between the two consecutive events. Computer networks are treated as discrete event systems where the start and end of a packet transmission can be viewed as an event happening in discrete intervals, and nothing interesting happens between the start of a packet transmission and the end of the packet transmission. Recording the interesting events varies from one simulation to another and depends upon the purpose of the simulation. The time when the event occurs is referred to as the time stamp and can be recorded to see the chronological order of the occurrence of the events.

4.1.2 Continuous Time Simulator

It refers to a computer modeling of a physical system where continuous monitoring of the phenomenon is required. In contrast to discrete event simulators, here, the simulator's state changes are continuous over time. It is used in simulations of systems in which a continuous change is observed in the phenomena and needs to be monitored, like a system monitoring liquid moving through a pipeline, continuous tracking of objects and so on.

In computer networks, discrete event simulators are generally used to simulate and model the communication system. The section to follow gives the brief discussion of some popular options available for doing WSN simulations.

4.2 WSN Simulators/Emulator/Testbed

A testbed's (Curren 2006) ultimate aim in any WSN deployment is to achieve the objective of the application in place. This objective can only be achieved by ensuring effective communication between WSN components. Consequently, overall performance and efficiency of communication between WSN components is a critical issue to which the entire WSN performance is tightly coupled. Thus, for the efficient working of the communication protocol, exhaustive performance evaluation is really crucial. To find the best available option for the performance evaluation of the communication protocol in WSNs, an extensive survey (Akyildiz et al. 2006; Chandresh Pratap et al. 2008; Egea-Lopez et al. 2005. ; Lessmann et al. 2008; Marko et al. 2009; Mehdi and Bernard 2008) of the available choices is done, to explore the options available to do the performance evaluation of the communication protocol in the WSN. For performance evaluation of communication protocols, Mehdi and

Bernard divided the available WSN simulators into two classes (Mehdi and Bernard 2008): General Purpose Simulation Packages and Specific Sensor Webs Simulation Tools.

To follow is the brief survey of the different software packages available to evaluate the performance of WSN communication protocols in General Purpose Simulation Packages and Specific Sensor Webs Simulation Tools.

4.3 Selection of WSN Simulator

Egea-Lopez et al. (2005) discussd four key features for the selection of an appropriate simulation environment. It includes 1) reusability and availability, 2) performance and scalability, 3) support for rich-semantics scripting languages to define experiments and process results and 4) graphical, debug and trace support. In this section, we focus on the impact of each feature in the context of the WSN.

4.3.1 Reusability and availability

The research community generally uses simulation tools for the performance evaluation of their novel protocols. Thus, two aspects should be considered for the choice of a simulation tool: 1) Are common framework/models implemented in the simulation software? It is mainly concerned with the age of the framework and its user base. Early and well received frameworks are more likely to have a broad base of models implemented. 2) Level of ease to glue new model with exiting one? It deals with how well simulation software is designed; simulators having a modular structure with well defined interfaces are easy to use and enhance. In this regard, all general purpose packages have a broad range of implemented models, whereas in special-purpose packages only specific models are available.

WSNs a use hierarchical structure, thus, simulation software should provide facilities to build multi-tier models having reusable components. WSN simulators should have a feature to provide open interfaces to create and process input and output files with commonly available software tools and allow embedding simulation models into larger applications (Xiaodong et al. 2008).

4.3.2 Performance and Scalability

The second feature to be considered during the choice of WSN simulation tools is performance and scalability. Performance is concerned with programming language effectiveness and scalability is related to the computing resource (processor, memory, storage) requirements. Toolkits that operate on real-time environments like emulators and time driven simulations do not take long, thus have no high performance and scalability concerns. Contrary to this, simulation toolkits have performance and scalability concerns because they operate on models containing complex interactions between radio propagation, mobility and energy consumption. To add to this challenge is the large scale of WSNs involving hundreds of thousands of nodes.

4.3.3 Support for rich-semantics scripting languages to define experiments and process results

A typical WSN simulation experiment requires a vast amount of variables which requires specific input scripting language having high level semantics. Similarly, a WSN simulation experiment generates volumes of data which require output scripting language to obtain results from the experiment. Thus, availability of an input and output scripting language to handle input/output data should be considered while choosing a WSN simulator.

4.3.4 Graphical, debug and trace support

As a typical WSN contains a large number of sensor nodes, sometimes, spanning to hundreds or thousands, a good WSN simulator should provide GUI support with error tracing and debugging features. There are three aspects of GUI in a simulator: 1) Debugger, to help in quickly detecting and tracing the errors. 2) Visual modeling and composition tool, to speed up the design of the experiments. 3) Result plotter, for visualizing results.

Xiaodong et al. (2008) stated some requirements to be fulfilled when choosing a WSN simulation. It includes: 1) As a typical WSN contains a large number of sensor nodes, sometimes, spanning to hundreds or thousands, a good WSN simulator should provide error tracing and debugging features. 2) WSNs use a hierarchical structure, thus, the simulator software should provide facilities to build a multi-tier model having reusable components. 3) The WSN should have a feature to provide open interfaces to create and process input and output files with commonly available software tools and allow embedding simulation models into larger applications.

Based on these features, OMNet++ is used for the performance evaluation of the proposed protocols in this thesis. The next section will explain the modeling simulation of the proposed protocols, which is done using OMNeT++.

4.4 Modeling and Simulation using OMNeT++

The following sections will explain how modeling and simulation of the proposed WSN protocols are done using OMNeT++.

4.4.1 OMNeT ++ overview

OMNeT++ (Andras 2008) is an open source, C++ based, object oriented, modular, discrete event network simulator with extensible architecture. In a typical OMNeT++ simulation, a network model is formed of reusable components called *modules*. Modules connect to each other using a port, referred to as *a gate,* to form a compound module. Modules can have unlimited nesting. Modules communicate with each other using message passing; the messages can have any data structure. A *message* can be communicated between modules using a connection formed through gates or direct communication. Modules have parameters that can be changed to customize a model topology and module behavior. *Simple modules* lie at the lowest level of the hierarchy and they are programmed using C++ to define the behavior of the simple module.

OMNeT++ provides various user interfaces to run the simulation which includes a graphical interface which provides demonstration and debugging features, and command-line user interfaces which provides an execution environment for batch execution. Simulators work on all common operating systems (Linux, Mac OS/X, and Windows). Using MPI and named pipes, OMNeT++ also supports parallel distributed simulation and it does not need any special customization to run in parallel, rather it only needs a simple configuration. OMNeT++ is open source and free for academic and non-profit purposes. For commercial purposes, it provides OMNEST for which users need to obtain an OMNEST license from Simulcraft Inc. OMNET++ is quite simple and has a clean design, which makes it suitable for using it as a prototyping platform for WSN modeling.

4.4.2 OMNeT++ Framework

OMNeT++ (Andras 2008) simulation model is formed from different modules which are referred to as simple modules (written in C++), that communicates with each other through message passing. The simple modules lie at the lowest level of the module hierarchy. The simple module is where the user implements algorithms to define the behavior of the communication entity. Algorithms are written in C++, using the OMNeT++ simulation class library. Simple modules can be assembled together to form compound modules and so on with unlimited depth of hierarchy. A complete simulation model is referred to as a network which in itself is a compound module. Message passing between the simple modules takes place either through direct connection or through a link formed by gates. A typical interconnection between simple and compound modules to form a network is shown in Figure 4.1; the shaded boxes show simple modules and un-shaded boxes represent compound/network modules. The arrows show the connection between modules through black boxes representing gates.

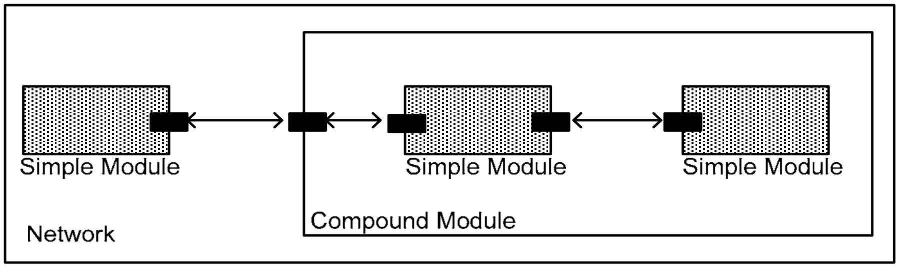


Figure 4.1 Simple and compound modules

Simple modules send data to each other directly or through links formed by the interconnection of the gates. Gates are input/output interfaces of the modules, where data is received at the input gate and data is sent at the output gate. Connection between the modules can be created by forming a connection between an input gate of one module with the output gate of another module, but only connection between modules at the same hierarchy level, modules within a compound module, corresponding gates of sub-modules or one sub-module and a compound module are allowed. Models reusing connections between the models of different hierarchy levels are not allowed. Due to the hierarchal structure of the model, starting from the sender simple module till arriving at the receiver simple module, the message needs to travel along many connections. A compound module on its own, is not associated with any behavior (C++ implementation), it just acts as “card board” (Varga and Hornig 2008), simply relaying the messages between the connections. Forming connection parameters including data rate, error rate, delay and other connection properties can be associated with one connection type (referred to as a channel) and then be reused. Modules have parameters which are used to pass configuration data to simple modules, act as the source of random numbers, prompt the user for values and help in defining the topology of the model. The model structure is described in OMNeT++'s NED language.

4.4.2.1 Messages, gates, links

Modules communicate with each other by exchanging messages. Messages in simulation can correspond to packets, frames, cells, bits or signals travelling in a network and so on. In real world communication, messages can contain any complex data structures. Simple modules can send data directly or through well defined path using gates and connections. When a message is received by the module, the local simulation time is progressed. Messages can be received from other modules or from the same module (referred to as self message, used for implementing timers).

Gates act as the input interface and output interface of modules through which a message is received and sent, respectively.

A *connection* acts as a link between two modules, which is formed by linking the input and output gates of two modules. Connections can be made between two modules lying on the same hierarchy, i.e., having the same compound module. A connection can be made between the gates of two sub-modules or between a sub-module and compound module. When a message is sent by one simple module to another simple module, due to the hierarchal structure of the modules, it needs to pass through many connections referred to as routes.

4.4.2.2 Modeling of packet transmissions

Connections can have three parameters: propagation delay, bit error rate, and data rate. Propagation delay refers to the delay incurred by messages when travelling through the channel. Bit error rate refers to the probability of incorrect transmission of a bit. Data rate is bits per second used for calculation of a packet's transmission time.

4.4.2.3 Programming the algorithms

Behavior of the simple module can be customized by writing C++ functions supported by the OMNeT++ simulation class library. C++ classes are used to represent different simulation objects, i.e., modules, gates, connections, parameters, messages, container classes (e.g. queue, array), data collection classes, statistic and distribution estimation classes (histograms, P2 algorithm for calculating quantities), transient detection, and result accuracy detection classes. They can be used to work together to create a simulation programming framework.

4.4.2.4 Building and running simulations

Following is the brief overview of building and running simulations in OMNeT++.

a) Building simulations

An A typical OMNeT++ simulation model consists of the following parts.

NED Language topology description (.ned files): It contains the description of the module in terms of parameters and gates. It can be specified in text editor as well as GUI.

Message Definition (.msg files): It contains the definitions of various message types which are translated into C++ classes by OMNet.

Simple module source: It refers to the C++ files, with the extension .h/.cc. It contains the algorithms that define the actual behavior of the simple module.

To customize the behavior of the modules, parameters can be assigned to the modules using the configuration file (omnet.ini) or NED file. Structure of the model is defined in the NED language description (Network Description).

Furthermore, the following components are provided by the simulation system.

1. *Simulation Kernel*: It has the code written in C++ and is compiled into a shared/static library, that supervises the overall simulation process and manages the class library

2. *User Interface*: It contains different user interfaces (CUI, GUI) written in C++ and compiled into libraries, which provide debugging, demonstration or batch execution of simulations.

The following components constitute a typical OMNeT++ simulation model. 1) .msg files which are translated into C++ code by opp_msgc. program, 2) All C++ source files which are compiled/linked with simulation kernel, 3) User Interface library that creates executable and shared library, and 4) NED files which are dynamically loaded in text form when simulation is started.

b) Running the simulations and analyzing the results

The simulation program can be run in two ways: 1) As an executable standalone program that can run on other computers having no OMNeT++ and 2) As a shared library, requiring OMNeT++ shared libraries present on that system.

When the simulation program is run it reads all NED files (having a topology model), then reads all configuration files (omnetpp.ini file experiment configuration, module parameters).

The simulation output is written on data files: Output vector file, output scalar file and may be some user define files. Matlab, Octave, spreadsheets (like OpenOffice Calc, Gnumeric or MS Excel), sed, awk or perl can be used for statistical analysis and visualization to process the output file generated by the simulation program.

4.4.3 Simulation Design

This section will discuss the architecture and overall design of the sensor network in the simulation environment. Topology of the sensor network is comprised of simple and compound modules in the OMNet++ framework, where a layer acts as a simple module and sensor node as a compound module (see Figure 4.2 redrawn from (Mallanda et al. 2005)). Simple modules are connected in a layered structure to form a sensor node compound module. Layers communicate with each other using gates. The hardware model of the sensor node is composed of a radio, CPU and battery module.

To follow is the description of the function performed by these layers (Mallanda et al. 2005).

4.4.3.1 Coordinator Module

It acts as an interlayer and coordinates the activities of the hardware and software modules of the sensor node. It has a reference of other layers and can access/update the properties of the other layers. It is connected with other layers through gates and communication is done using messages. It controls and coordinates all activities of the sensor node, e.g., when physical a module sends/receives packets or when a battery module updates energy. It registers the sensor node and deactivates it when its energy becomes zero.

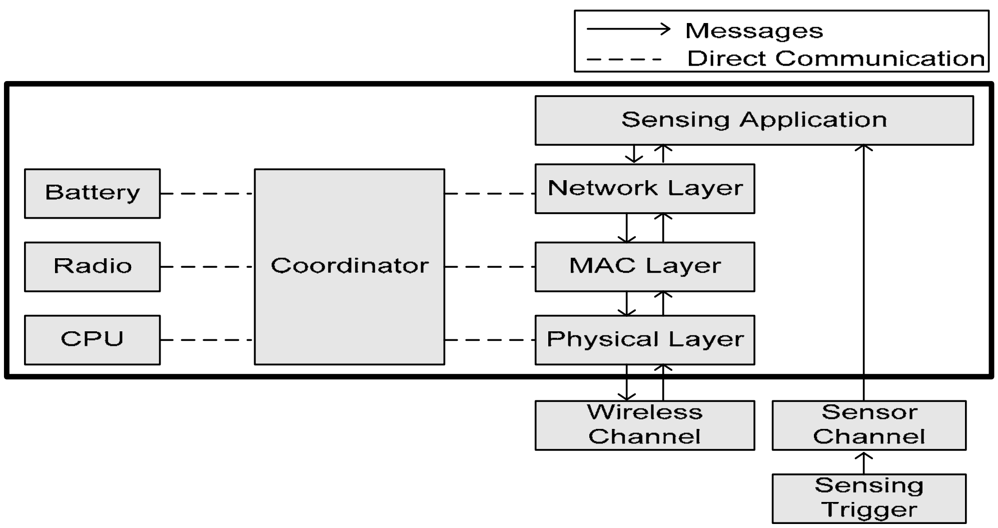


Figure 4.2 Sensor node's simulation model

4.4.3.2 Hardware Model

The hardware model of a sensor node includes modeling of its three major components including battery model, CPU model and radio model. Following is the brief discussion of these three models (Mallanda et al. 2005).

a) Battery Model

It is responsible for supplying the battery for operation of the CPU, radio and sensor modules. It has some initial energy which is deducted when the CPU, radio and sensor module operations are performed. Deduction is done using an energy model, i.e., linear battery model or discharge rate dependent model. Lifetime of the battery T in hours can be estimated as follows.

$$T = \frac{C}{I} \tag{4.1}$$

where

C remaining battery (Ampre-hours)

I total current drawn (Amperes)

The remaining battery using the linear discharge model can be calculated as:

$$C = C_{in} - \int_{\Delta t} I(t) \Delta t \tag{4.2}$$

where

C_{in} is the initial battery capacity

$I(t)$ is the current drawn in time Δt

This model works with assumptions including no self discharges and no degradation of battery when getting old.

b) CPU Model

Sensor nodes have low processors and generally power consumed for CPU operation is very low. Furthermore, energy consumption for idle, sleep and active states are different. All these energies are considered in this model.

c) Radio Model

It is concerned with power consumed for transceiver operations, i.e., sending and receiving. Furthermore, energy consumption for idle, sleep and active states are different. All these energies are considered in this model.

4.4.3.3 Wireless Channel Model

It controls all the connections between sensor nodes which enable sensor nodes to communicate with each other. Messages are sent from one node to the other node with delay D being:

$$D = \frac{d}{c} \tag{4.3}$$

where

d is the distance between sensor node.

c is the speed of light.

D is the distance between sensor node.

To predict the strength of the signal several propagation models are used, including Free Space Propagation Model and Two ray ground reflection model.

a) Free Space Propagation model

It assumes ideal communication conditions with a clear line of sight between sender and receiver. The following equation is used to calculate the received signal power in free space at distance d.

$$P_r = (P_t * G_t * G_r * \lambda^2) / (4\pi)^2 * \mathrm{d}^2 * \mathrm{L}^2 \tag{4.4}$$

where

P_r is the power of the signal transmitted

P_t is the power of the signal transmitted

G_t, G_r are the antenna's transmitter and the receiver gains, respectively.

L is the system loss, and λ is the wavelength.

b) Two-ray ground reflection model

It considers direct path as well as ground reflection path communication and has more accurate prediction than the free space model. The following equation is used to calculate the received signal power in the two ray ground reflection model at distance *d*.

$$P_r = (P_t * G_t * G_r * {h_t}^2 * {h_r}^2) / \mathrm{d}^4 * \mathrm{L} \quad (4.5)$$

where h_t and h_r are the heights of the transmitter antennas and the receiver, respectively.

4.4.3.4 Sensor Node Stack

The sensor node stack has AppLayerSimple at the highest level of its hierarchy and relates to the behavior of the application layer. It communicates with the lower Network layer module through gates; it simulates sending/receiving of the packet and attaches the network address with the packet, and sends the packet to the MAC layer simple module. The MAC layer is responsible for accessing media as well as acting as an interface between the routing layer and the lower Physical layer by sending the packet from routing to the Physical layer.

4.5 Results and Analysis

Following sections will present the simulation results and their analysis, explaining the performance evaluation of the proposed protocols QoSEC, QoSES and QoSEM.

4.5.1 QoS based Energy Efficient Clustering (QoSEC) Protocol for Wireless Sensor Networks: Result and Analysis

Based on the developed system model, simulations are carried out using OMNet++ (Andras 2008) to evaluate the performance of the proposed QoSEC protocol. The performance of the proposed QoSEC protocol is compared against the three contemporary energy-efficient protocols: CPCP (Soro and Heinzelman 2009), EEMC (Jin et al. 2008) and LEACH (Heinzelman et al. 2000). The following are the details of the simulation setup, energy model, and discussion of the results for QoSEC.

4.5.1.1 Simulation Setup

Simulations were conducted in the sensing area of 200 × 200 m^2 and the number of sensor nodes varied from 20 to 240 for different experiments. Sensor nodes were randomly deployed, and the random deployment is achieved by choosing (x, y) locations based on a uniform distribution. The BS is fixed and located at the center of the network. The simulations were conducted with a communication range which is equal to double of their sensing range. The simulation parameters are given in Table 4.1.

Table 4.1. Simulation parameters for QoSEC

Simulation Parameter	**Notation**	**Value**
1. NETWORK		
Network area/size	S	200×200 m^2
BS location	(x, y)	(100, 100)
Number of nodes	N	20-240 nodes
ID’s of sensor nodke	ID	0-240
Data Rate	$D_r/10s$	5 TDMA frame
Node radio range	R_{sense}	15 m
Communication range	R_{comm}	30 m
Node distribution		Random
Number of frame per round	N_{fpr}	1
Simulation Time	S_{time}	900 sec
Number of trails	N_t	20
Period of each cycle	P_{cycle}	5 sec
Number of transmission levels	T_{level}	4

Simulation Parameter	**Notation**	**Value**
2. BATTERY		
Initial Energy(J)	E_I	1J
Dead nodes(J)	E_D	< 0.1 J
3. RADIO		
Energy spent in transmitter/receive	E_{elec}	50 nJ/b
Constant for free space propagation	ϵ_{fs}	10 nJ/b
Constant for multi path propagation	ϵ_{mp}	0.003pJ/bit/m^4
Data aggregation/fusion energy	E_{DA}	5 nJ/bit/signal
4. APPLICATION		
Data Packet Size	P_{dsize}	500 bytes
Control Packet Size	P_{csize}	25 bytes
Packet rate	P_{rate}	1 packet/s
Transmission range Broadcast	R_{bc}	70 m
Period of each round	P_r	5 sec
Data rate (kbps)	D_{rate}	19.2kbps
Synchronization Interval (s)	T_s	60 sec
Number of Synchronization message	N_s	2
Transmission Period	*T(s)*	60 sec
Channel Bandwidth	B_c	1 Mbps
Sensor DATA-timer	S_{DT}	2 sec
CH Energy threshold	E_{th}	10^{-4} J
5. MESSAGE EXCHANGES		
HELLO	*H-packet*	25 bytes
INVITE	*I-packet*	25 bytes
SCHEDULE	*S-packet*	25 bytes
CONFIRM	*C-packet*	255 bytes

4.5.1.2 Event Summary for QoSEC

The following is the event summary of the main events as used in the QoSEC simulation (see Table 4.2).

Table 4.2. Events summary for QoSEC

Sno.	Event Name	Description
1.	*Initialization()*	This event is executed in the start, when the node is initialized. It involves initialization of the node parameters, some of which includes node ID, initial residual energy, random location coordinates etc.
2.	*CH_ANN*	This is activated when a node's timer expires and needs to declare itself as a CH by sending this event.
3.	*CH_JOIN*	This event is activated when a node receives a CH announcement event. This event is the confirmation by a node to be part of that CH.
4.	*Cluster Status*	This event is activated when either the timer for *CH_ANN* expires or the node receives *CH_ANN* from some other node. It involves either the change of CH status from *NULL* to new CH or *NULL* to make itself CH.
5.	*End Timer*	An event triggered when the WAIT timer expires. It can be at the completion of any of timer T_m, T_s, T_c and T_{dc} (mentioned in Algorithm 1). This event indicates that a node should terminate the timer related tasks and perform the next relevant operation.

4.5.1.3 Energy Model

It is assumed that the sensor nodes have the ability to adjust their transmission power according to the distance of the receiving node. The energy model presented in (Muruganathan et al. 2005) is adopted here. The amount of energy consumed for transmission E_{TX} of an l-bit message over a distance is given by:

$$E_{Tx} = \begin{cases} l.E_{elect} + l.\varepsilon_{fs}.d^2 & for \quad 0 \le d \le d_{crossover} \\ l.E_{elect} + l.\varepsilon_{mp}.d^4 & for \quad 0 \ge d_{crossover} \end{cases} \quad (4.6)$$

where E_{elect} is the amount of energy consumed in electronics, ε_{fs} is constant for free space propagation and is the energy consumed in an amplifier when transmitting at distance shorter then $d_{crossover}$, and ε_{mp} is constant for multi-path propagation and is the amplifier energy consumed in an amplifier when transmission is at a distance greater the $d_{crossover}$. The energy expended in receiving a 1-bit message is given as:

$$E_{Rx} = l\, E_{elect} \quad (4.7)$$

4.5.1.4 Results and Discussion

The performance of QoSEC is compared against the LEACH, CPCP and EEMC protocols. Experimental parameters, such as energy per packet, network lifetime, average delay per packet, average packet loss, throughput, and coverage lifetime are used to measure the QoSEC performance.

a) Average Energy per Packet

Average energy per packet is a measure of energy spent in forwarding a packet to a BS. It is an indicator of the lifetime that can be achieved by the protocols. In Figure 4.3, the average energy per packet is plotted on the y-axis, with a varying number of sensor nodes (from 20 to 240) on the x-axis. It can be observed that the average energy consumption per packet for the proposed QoSEC protocol is less than the other three protocols, indicating that the proposed protocol has improved the lifetime of the WSN. The reason for lifetime improvement is that in QoSEC, by virtue of multi-hierarchy, each node just needs to send its data to the local sink (super-node) in its neighborhood. With LEACH, CPCP and EEMC, however, each node has to relay the data all the way to the BS, which involves energy consumption of all the nodes (hops) connecting to the BS.

Furthermore, energy efficiency in QoSEC is complemented by the two proposed algorithms: *cluster-head-selection* and *sleep/wake scheduling* algorithm. These two algorithms also contribute to energy savings by delegating resource intensive CH jobs to higher energy nodes, and by giving longer active intervals to higher energy nodes. As a result, all the nodes in the network deplete their energy in a balanced manner. Therefore, using the notion of super-node, energy-efficient cluster-head selection algorithm and energy-efficient sleep-awake scheduling scheme makes QoSEC a more effective protocol for conserving energy than LEACH, CPCP and EEMC.

.

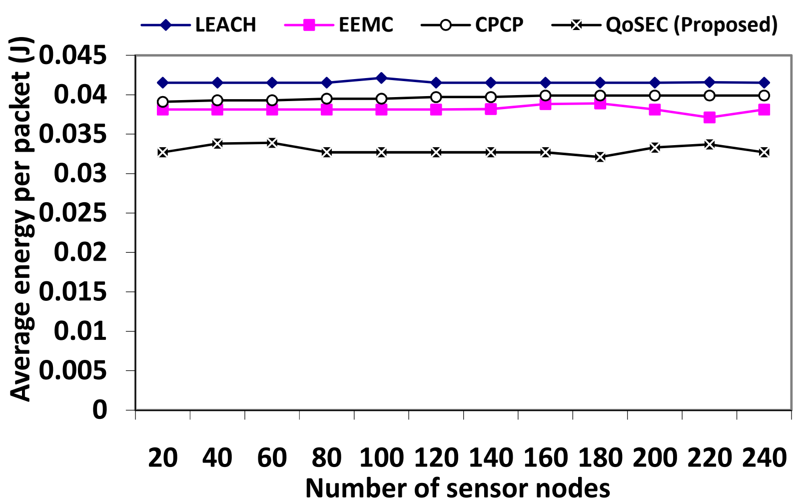


Figure 4.3 Average energy per packet for different number of sensor nodes

b) Network Lifetime

Network lifetime is how long the percentage of nodes remain alive. It is represented by $T_{n\%}$ where n is the number of alive nodes. Figure 4.4 shows the network lifetime corresponding to four protocols. It can be observed that the proposed protocol has the longest $T_{100\%}$ followed by EEMC, CPCP and LEACH. Indicating that the proposed protocol improves the lifetime of the WSN as compared to EEMC, CPCP and LEACH. The reason for the lifetime improvement is that in QoSEC, by virtue of multi-hierarchy, each node just needs to send its data to the local sink (super-node) in its neighborhood. With LEACH, CPCP and EEMC, however, each node has to relay the data all the way to the BS through the same nodes which are near the sink. Over time, these nodes expire, forming a hot spot/energy hole near the sink which significantly limits the network lifetime. Furthermore, the two proposed algorithms: cluster-head-selection and sleep/wake scheduling algorithm also contribute to maximizing network lifetime. In these two algorithms, the resource intensive CH role is given to higher energy nodes, and longer active intervals are given to higher energy nodes. As a result, all the nodes in the network deplete their energy in a balanced manner. Therefore, by using the notion of super-node, the algorithm for energy-efficient cluster-head selection and energy-efficient sleep-awake scheduling make QoSEC more effective in prolonging the network lifetime than EEMC, CPCP and LEACH.

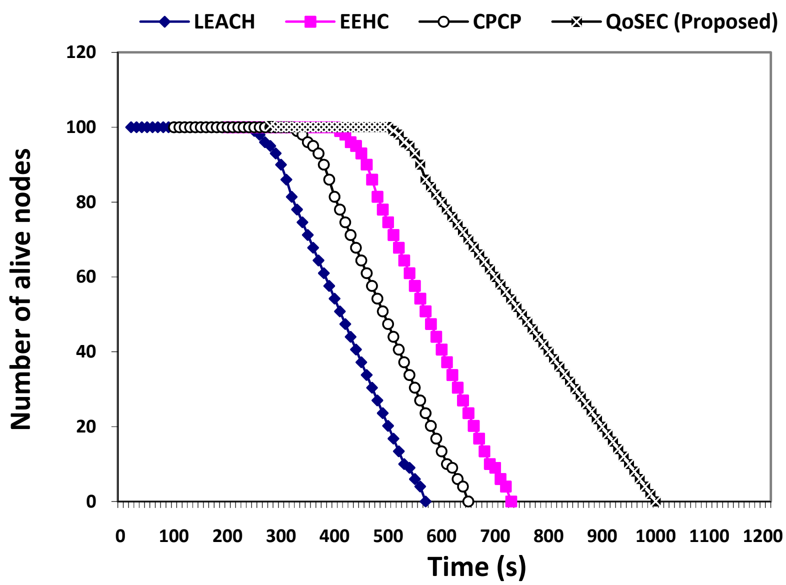


Figure 4.4 Network Lifetime

c) Average Delay per Packet

Delay is referred to as the time span between the packet sent from a sensor node and the packet received at a BS. Delay values are measured by changing the number of sensor nodes from 20 to 240. As shown in Figure 4.5, the average delay experienced by the proposed QoSEC protocol is the least, while CPCP and EEMC being the second and third, respectively, and LEACH has the worst delay time. In QoSEC, every node is supposed to send their data to the local sink (super-node) in their vicinity, which means that each packet does not wait long when it is propagated to the BS. As a result, average delay per packet in QoSEC is less than CPCP, EEMC and LEACH. In CPCP, EEMC and LEACH protocols, all nodes have to relay their data all the way to BS using multi-hop communication, and it involves many relay nodes to reach the BS. Considering the local traffic at each node, sensed data has to wait for some time at each node to get attention, as well as the time required to forward the data to the next node or BS. It makes LEACH, CPCP and EEMC prone to a longer delay. Furthermore, as the number of nodes increases the difference between QoSEC and the other three strategies increases as well. For QoSEC, the performance remains the same for the increased number of nodes, since some super-nodes are always there to be used as local sinks, whatever may be the size of the network. In this way, increasing the node number has no effect on QoSEC. Therefore, it suggests that the proposed QoSEC protocol is more scalable than CPCP, EEMC and LEACH protocols.

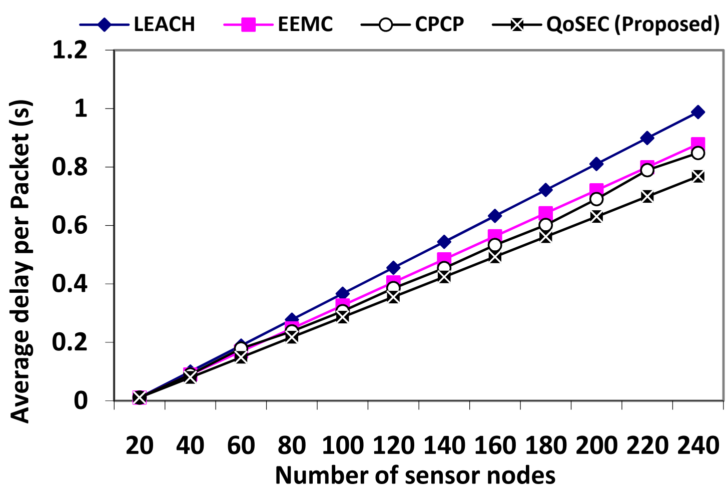


Figure 4.5 Average delay per packet for different number of sensor nodes

d) Packet Loss Ratio (%)

Packet lost ratio refers to the percentage of the packets that could not reach the BS, i.e.,

$$Packet\ Loss\ Ratio = \frac{t}{T} \tag{4.8}$$

where

t is total number of packets not received at the BS

T is total number of packet sent by all the sensor nodes.

Figure 4.6 shows the measurement of packet loss ratio for the four protocols with varying numbers of sensor nodes (20-240). It is clear that the proposed QoSEC protocol has a far less packet loss ratio as compared to CPCP, EEMC and LEACH. This is because in CPCP, EEMC and LEACH, a node needs to send data all the way to an intended BS using multi-hop communication. It makes CPCP, EEMC and LEACH more vulnerable to packet loss, as many nodes communicating at the same time result in an increasing number of collisions, thus reducing transmission reliability. With QoSEC, nodes are required to forward their data to a local sink (a super-node). The super-node will then send the data to the BS via multi-hop super-node communications, or direct to the BS. Hence, QoSEC involves far less relay nodes, which makes its less prone to packet loss. As a result, the QoSEC protocol outperformed the other three protocols in terms of packet loss ratio.

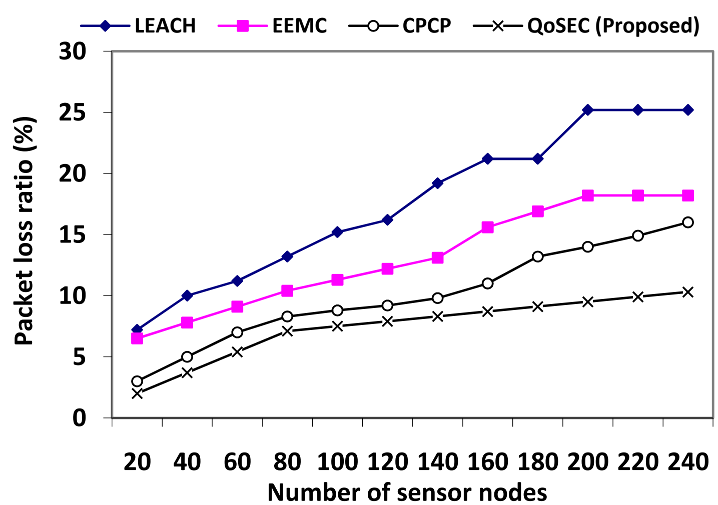


Figure 4.6 Packet loss ratio for different number of sensor nodes

e) Throughput (Packets per second)

Throughput is measured by the number of packets received per second at the BS. In this experiment, the numbers of nodes were varied from 20 to 240 and the throughput is measured at the BS. It can be seen from Figure 4.7 that by increasing the number of sensor nodes, the throughput for QoSEC, CPCP, EEMC and LEACH increases. Initially, for 20 sensor nodes, QoSEC has a more or less same throughput as that of CPCP and EEMC. For the larger network size, later, QoSEC has achieved throughput greater than all three protocols, i.e., CPCP, EEMC and LEACH. It shows that the QoSEC protocol is scalable and can perform better as the size of the WSN becomes larger. Simply put, out of the four evaluated protocols, QoSEC has the best throughput, while EEMC is second, CPCP being the third and LEACH has the minimum throughput.

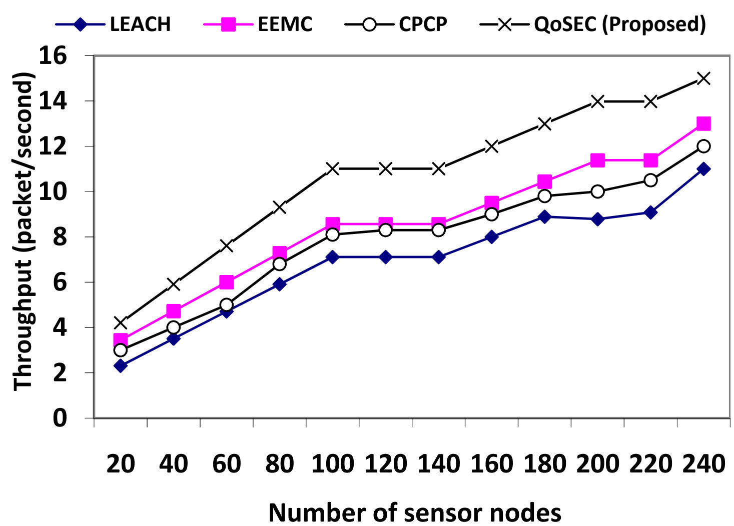


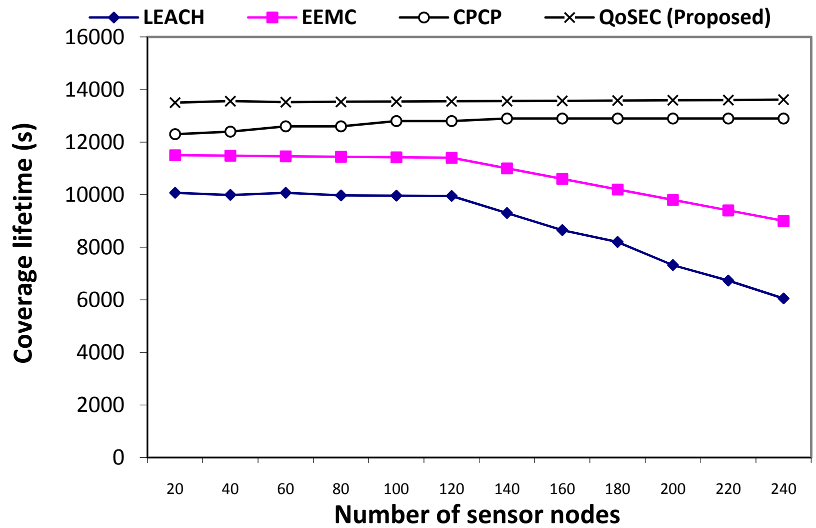
Figure 4.7 Throughput (packet per second) for different number of sensor nodes

f) Coverage Lifetime

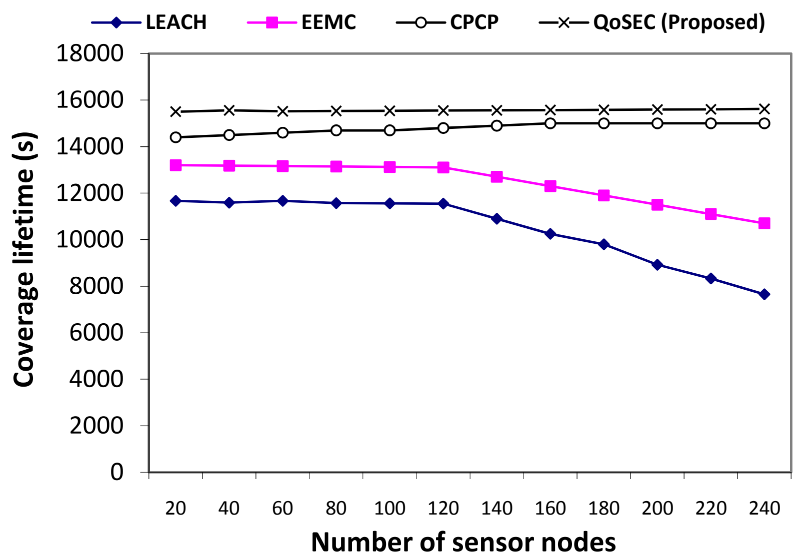
Coverage Coverage lifetime is referred to as the time the network is able to preserve 100% or over 90% coverage of the whole sensing area. As a generalization, coverage of less than this percentage is not tolerable and can be regarded as a failure of the whole network. In this experiment, the number of nodes is varied from 20 to 240, and coverage lifetime (100%, 90%) is measured. Figure 45.8(a) provides coverage lifetime for 100% coverage area, while Figure 4.8(b) depicts coverage lifetime for 90% coverage area.

It can be observed that QoSEC outperforms CPCP, EEMC and LEACH in terms of coverage lifetime in both cases of 100% and 90% coverage. The reason is that, in QoSEC, in addition to the efforts for optimizing energy efficiency, QoSEC also considers coverage as a CH selection parameter. As a result, QoSEC provides better coverage for a longer period of time. Though CPCP also considers the coverage preservation in cluster head selection, it still has less coverage lifetime than QoSEC. It is because of the fact that QoSEC by virtue of multi-hierarchy clustering is able to prolong the network lifetime.

It ultimately results in increasing coverage lifetime as more nodes are alive for longer periods of time, whereas, this feature is missing in CPCP. While for LEACH and EEMC, the CH selection mechanism is merely based on energy efficiency and has no consideration for preserving the coverage lifetime. Hence, these two protocols always select nodes that can provide energy efficiency, but this can adversely affect the coverage lifetime of the network. Quite often, coverage lifetime of LEACH and EEMC is compromised at the expense of energy efficiency. This is not the case with QoSEC, where it not only considers energy efficiency, but also coverage lifetime as parameters in CH selection. Another feature that complements the coverage lifetime is the sleep/wake scheduling algorithm, where it also considers energy efficiency and coverage lifetime in its scheduling heuristic. In this way, the QoSEC coverage lifetime is better than the other three protocols (EEMC, CPCP and LEACH).



(a) Coverage Lifetime considering 100% coverage



(b) Coverage Lifetime considering 90% coverage

Figure 4.8 Coverage lifetime for different numbers of sensor nodes: (a) 100% coverage and (b) 90% coverage

4.5.1.5 QoSEC: Summary of results

This section, discusses the simulations carried out in OMNeT++ to evaluate the performance of the proposed QoSEC protocol and compare the results against the contemporary LEACH, EEMC and CPCP protocols. Simulation results demonstrate that QoSEC has achieved significant energy savings and enhanced the network lifetime, as well as the coverage lifetime. The use of super-nodes and the inclusion of multi-parameters (energy, coverage) in the cluster-head-selection algorithm and sleep/wake scheduling algorithm in QoSEC have helped the sensor nodes to dissipate their energy at a much more balanced rate, while at the same time keeping good coverage of the sensing area. The other protocols are not equipped with these capabilities. The results produced also demonstrate that QoSEC is effective in improving the other QoS parameters, such as average delay, packet loss ratio, throughput and coverage lifetime. Table 4.3 summarizes the findings from this research work.

Table 4.3. Summary of performance comparison between LEACH, CPCP, EEMC and QoSEC

Protocols compared (240 nodes)	**Average energy per packet (J)**	**Average delay per packet(s)**	**Packet loss ratio (%)**	**Throughput (packet/s)**	**Coverage lifetime (s)**
QoSEC (The proposed protocol)	0.041533	0.768	10.3	15	13617
CPCP	0.0399	0.848	16	12	12900
EEMC	0.03811	0.878	18.2	13	9000
LEACH	0.041533	0.988	25.2	11	6050

4.5.2 QoS based Energy Efficient Sleep/wake (QoSES) Scheduling Protocol for Wireless Sensor Networks: Result and Analysis

Based on the developed system model, simulations are carried out using OMNet++ (Varga and Hornig 2008) to evaluate the performance of the proposed QoSES protocol. Performance of proposed protocol is compared with the two contemporary protocols: S-MAC (Wei et al. 2002) and anycast protocols (Joohwan et al. 2008). The following are the details of the simulation setup, energy model, and discussion of the results.

4.5.2.1 Simulation Setup

Simulations were conducted in the sensing area of 200 x 200m^2 and the number of sensor nodes varied from 20 to 400 for different experiments. Sensor nodes were randomly deployed and the random deployment is achieved by choosing (x, y) locations based on a uniform distribution. The BS is fixed and located at the center of the network. The simulations were conducted with a communication range equal to double of their sensing range. The simulation parameters are adopted as used in FlexiTP (Lee et al. 2008) which are based on Mica2Mote hardware. See Table 4.4 for the listing of the simulation parameters.

Table 4.4. Simulation parameters for QoSES protocol

Simulation Parameter	**Value**
Sensing Area	200×200m^2
Bandwidth	38.4Kbps
Transmission Range	50m
Transmit mode power	60mW
Receive mode power	30mW
Idle mode power	30mW
Sleep mode power	0.003mW
Transition power	30mW
Transition time	2.45ms
Packet Size	96bytes
Time slot size	42ms
Simulation time	300s

4.5.2.2 Event Summary for QoSES

The following is the event summary of the main events as used in the QoSES protocol (see Table 4.5).

Table 4.5. Events summary for QoSES

Sno	Event Name	Description
1.	*Initialization()*	This event is executed in the start, when the node is initialized. It involves initialization of node parameters, some of which include node ID, initial residual energy, random location coordinates etc.
2.	*SN_W*	This is activated when a node's self timer expires. Senor nodes then change their state from sleep to active.
3.	*SN_S*	This is activated when a node has finished sending/receiving data or the self timer expires. The senor node then changes its state from active to sleep.
4.	*E_DETECT*	This event is activated when a node sensesany event. It is a self message.
5.	*SEND_MSG*	This event is activated when a node receives a self message event E_DETECT. It involves sending the data to the next hop neighbor using RTS/CTS/DATA.
6.	*RECEIVE_MSG*	This event is activated when a node receives any message RTS/CTS/DATA. It involves establishing communication with the sender node and receives the data using RTS/CTS/DATA.
7.	*End Timer*	An event triggered when the WAIT timer expires. It can be at the completion of any timer event detection, sleep or wake timer.

4.5.2.3 Energy Model

The same energy model as presented in section 4.5.1.3 is used here.

4.5.2.4 *Results and Discussion*

The performance of QoSES is compared against the S-MAC and anycast protocols. Experimental parameters, such as average delay per packet, energy per packet, average packet loss, and throughput, are used to measure the performance of QoSES.

a) *Average Delay per Packet*

In this experiment, delay values are measured by changing the number of sensor nodes from 20 to 240. As shown in Figure 4.9, the average delay experienced by the proposed QoSES protocol is the least, while anycast being the second and S-MAC has the worst delay time. In QoSES, as nodes are given different wake intervals according to their traffic requirement with respect to their position in the network, their topological importance and their proximity from the event. The proposed protocol is able to minimize delay at each hop because nodes do not have to wait long for the wakeup interval of the next hop. As a result, the average delay per packet in QoSES is less than with the anycast protocol and S-MAC. In the anycast protocol, though the node has multiple next-hop relaying nodes by virtue of the anycast packet-forwarding scheme, which helps to find the next hop neighbor in a quick manner, it still does not consider the varied traffic requirements of different nodes. Thus, node wait time increases as packets approach the nodes near the BS. Therefore, it has greater delay than the proposed protocol. In the S-MAC protocol, nodes have fixed wake intervals for the whole network irrespective of their traffic requirement, thus, each node has to wait for the wake interval of the next hop. However, as all nodes have to relay their data all the way to the BS using multi-hop communication. Hence, it involves many relay nodes to reach the BS, which increases end-to-end delay. Considering the local traffic at each node, the sensed data has to wait for some time at each node to get attention, thus delay becomes longer when a packet approaches the nodes near the BS. The problem gets worse when the packet approaches the nodes near the BS as the packet suffers maximum delay. It makes the anycast protocol and S-MAC prone to a longer delay. Furthermore, as the number of nodes increases, the difference between QoSES and the other two strategies increase as well. For QoSES, the performance remains the same for the increased number of nodes, since the wake interval is adaptive to the traffic load, whatever the size of the network may be. In this way, increasing the node number has no effect on QoSES. Therefore, it suggests that the proposed QoSES protocol is more scalable than S-MAC and anycast protocols.

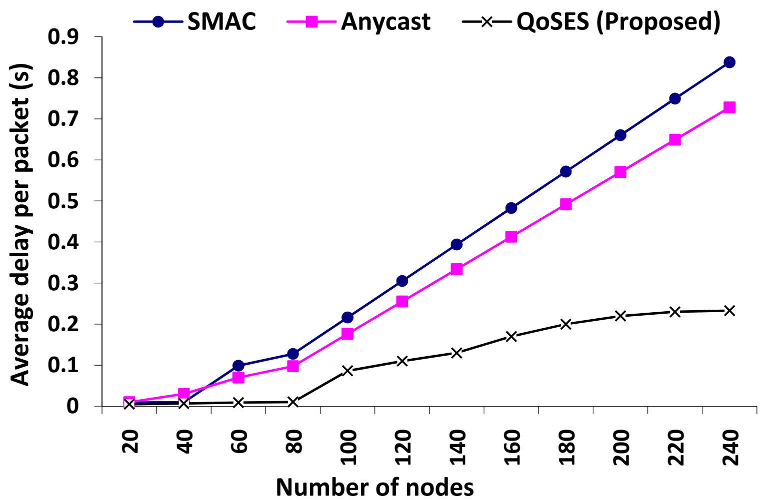


Figure 4.9 Average delay per packet different numbers of sensor nodes.

b) Average Energy per Packet

In this experiment, average energy per packet is plotted on the *y*-axis, with a varying number of sensor nodes (from 20 to 240) on the x-axis (see figure 4.10). It can be observed that the average energy consumption per packet for the proposed QoSES protocol is less than the other two: anycast protocol and S-MAC protocol, indicating that the proposed protocol network lifetime is greater than anycast and S-MAC protocols. The reason for the lifetime increase in the proposed protocol is that, in QoSES the wake intervals of the nodes are defined according to the traffic requirements of the nodes identified by their position in network, their topological importance and their proximity to the event. Doing so, the proposed protocol avoids the case where the nodes remain awake and idle as no traffic is to be forwarded (idle listening). Whereas, in the anycast protocol many nodes stay awake to provide alternate paths for routing and mostly they remain idle, as expected traffic requirements of nodes are considered while setting up the sleep/wake schedule. It results in increasing the wake node staying idle, which significantly limits the network lifetime. Similarly in the S-MAC protocol, the random sleep/wake schedule is defined for all the nodes which increase the number of wake idle nodes especially as they move away from the BS. As nodes away from the BS have to do less relaying jobs. It uses the energy of the nodes in idle listening and ultimately network lifetime of the network is minimized. Hence the proposed protocol has less energy per packet than both anycast and S-MAC protocols.

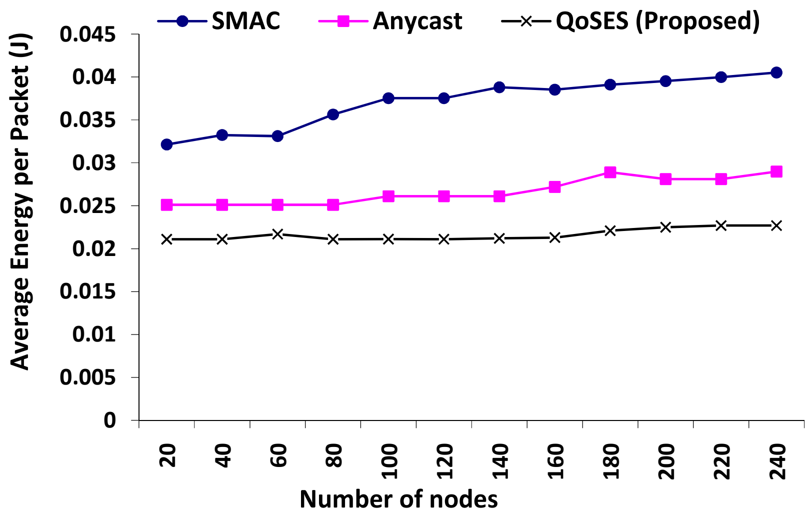


Figure 4.10 Average energy per packet for different numbers of sensor nodes

c) *Packet Loss Ratio (%)*

In this experiment, the measurement of packet loss ratio for the three protocols with varying numbers of sensor nodes (20-240) is done (see figure 4.11). The packet loss ratio for all protocols increases as the number of nodes increases but their slopes are different. The packet loss ratio of the proposed protocol increases at the slowest rate because it considers the traffic pattern of different nodes and accordingly assigns wake intervals, which results in less packet loss. For S-MAC the packet loss ratio grows at a relatively high speed because it uses a random sleep/wake schedule, which increases schedule misses and ultimately increases the packet loss ratio.

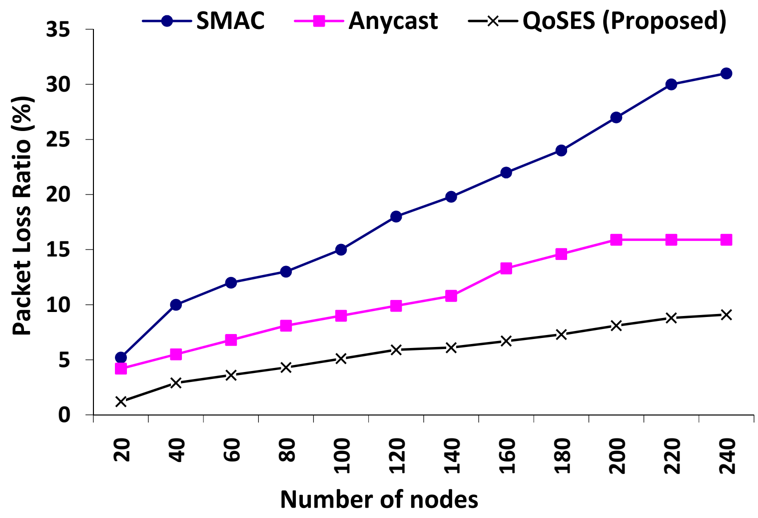


Figure 4.11 Packet loss ratio for different number of sensor nodes

It is clear from Figure 4.11 that for the whole simulation, the proposed QoSES protocol has far less packet loss ratio as compared to S-MAC and anycast protocols. It is because in S-MAC and anycast protocols, there is no congestion control mechanism to ensure adaptability in the wake interval, thus, collision occurrence is higher. It makes S-MAC and anycast more vulnerable to packet loss, thus reducing transmission reliability. With QoSES, nodes have adaptive wake intervals according to their traffic load requirement, thus, traffic flows through the network smoothly. Furthermore, in QoSES, critical nodes in terms of connectivity have longer wake intervals, which contribute to decreasing packet loss. Nodes detecting events and nodes in its vicinity are assigned greater wake intervals, which also contribute in minimizing the lost packets. Hence, QoSES is less prone to packet loss than anycast and S-MAC protocols. As a result, the QoSES protocol has outperformed the other two protocols in terms of packet loss ratio.

d) Throughput (Packets per Second)

In this simulation, the numbers of nodes were varied from 20 to 240 and the throughput is measured at the BS. It can be seen from Figure 5.12 that by increasing the number of sensor nodes, the throughput for QoSES, S-MAC and anycast protocols have increased. However for a greater network size after that, QoSES has achieved a throughput greater than both the S-MAC and anycast protocols. It shows that the QoSES protocol is scalable and can perform better as the size of the WSN becomes larger. Simply put, out of the three evaluated protocols, QoSES has the best throughput, while S-MAC is second, and anycast protocol has the least.

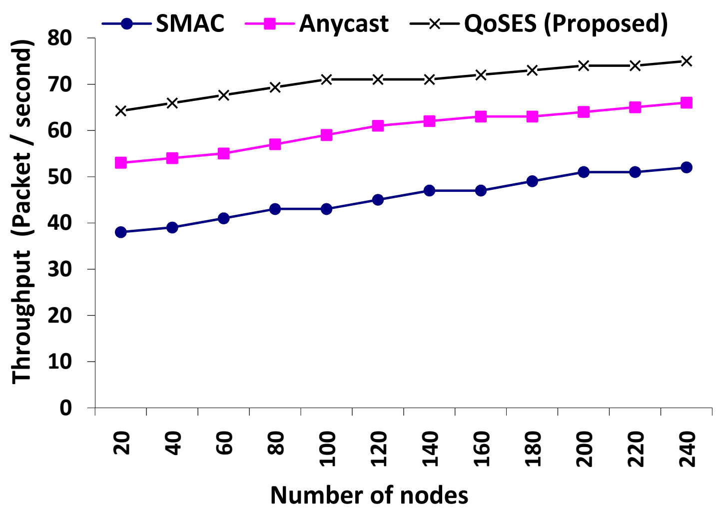


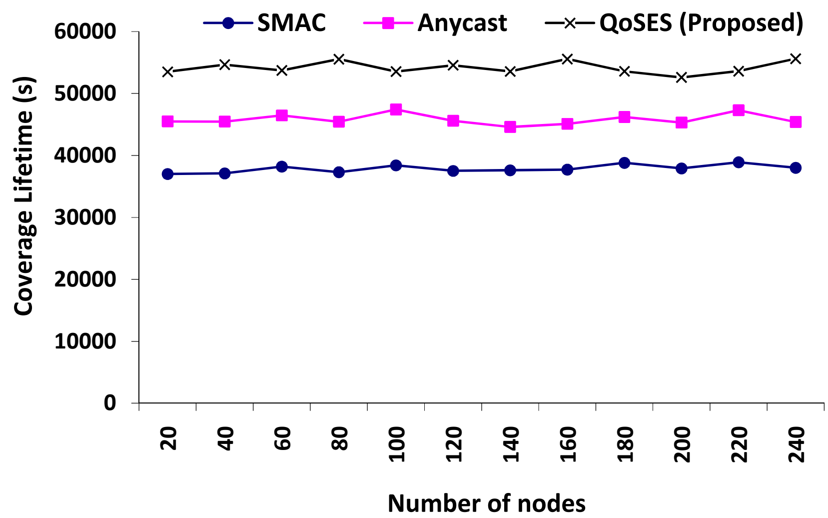
Figure 4.12 Throughput (packet per second) for different numbers of sensor nodes

e) *Coverage Lifetime*

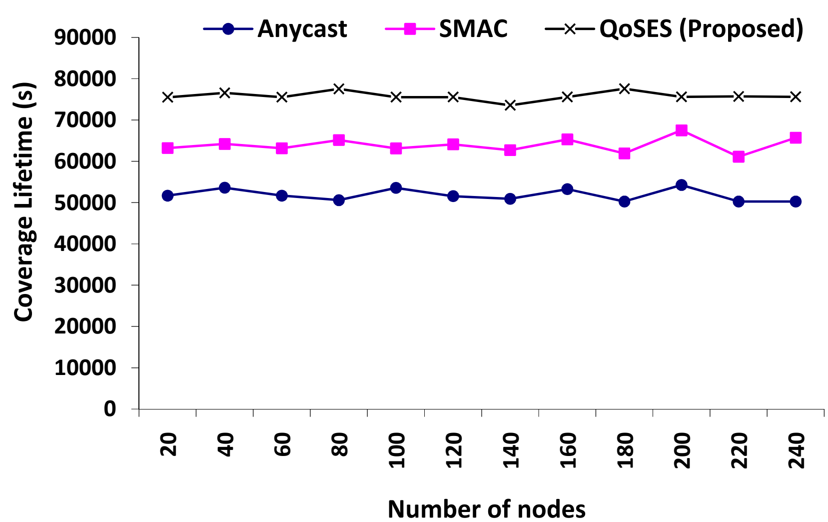
In this experiment, the number of nodes is varied from 20 to 240, and coverage lifetime (100%, 90%) is measured. Figure 4.13(a) provides coverage lifetime for 100% coverage area, while Figure 4.13(b) depicts coverage lifetime for 90% coverage area.

It can be observed that the proposed protocol outperforms anycast and S-MAC in terms of coverage lifetime in both cases of 100% and 90% coverage. The proposed protocol has coverage lifetime better than the anycast protocol, because in S-MAC approach nodes use random sleep/wake scheduling. It increases the idle listening and involves more schedule misses because while defining the sleep/wake schedule, the traffic requirements of the nodes are not considered. It results in wastage of the node's energy and ultimately its battery expires. It results in the information of small hot spots/energy holes. Ultimately, these energy holes emerge as coverage holes and limit the coverage lifetime of the S-MAC approach. Whereas, in the proposed protocol, the sleep/wake schedule of the nodes are assigned according to the variable traffic requirements of the nodes defined by their position in the network, their topological importance and their proximity to the event. It helps in minimizing idle listening, hence saving the node energy which results in better coverage. It results in a balanced use of energy of all sensor nodes in the network. Ultimately, no coverage hole is formed, hence, increasing the coverage lifetime of the network when compared with the S-MAC approach. Hence, proposed protocol has better coverage than the S-MAC protocol.

The proposed protocol has a coverage lifetime better than the anycast protocol, because in anycast protocols more than one path is made active to have smooth routing. It wastes the energy of some of the nodes in idle listening and schedule misses because in making nodes active, traffic requirement of the nodes is not considered. These nodes soon expire their energy and appear as coverage holes in the network. Whereas, in the proposed protocol, the coverage hole is avoided by having efficient use of the nodes' energy by assigning the nodes sleep/wake schedule according to their traffic requirement with respect to their position in the network, their topological importance and their proximity to the BS. It decreases the idle listening and schedule misses.



(a) Coverage Lifetime considering 100% coverage



(b) Coverage Lifetime considering 90% coverage

Figure 4.13 Coverage lifetime for different number of sensor nodes: (a) 100% coverage and (b) 90% coverage

It denies formation of any energy or coverage hole in the proposed protocol. It results in a balanced use of energy of all sensor nodes in the network. Ultimately, no coverage hole is formed, hence, increasing the coverage lifetime of the network when compared with the anycast approach. In this way, the proposed protocol coverage lifetime is better than the anycast and S-MAC protocols.

4.5.2.5 QoSES: Summary of results

This section discusses the simulations carried out in OMNeT++ to evaluate the performance of the proposed QoSES protocol and compares the results against the contemporary S-MAC and anycast protocols. Simulation results demonstrate that the proposed protocol has significantly reduced the end-to-end delay, as well as improved the other QoS parameters of average energy per packet, average delay, packet loss ratio, throughput and coverage throughput. Table 4.6 summarizes the findings from this research work.

Table 4.6. Summary of performance comparison between QoSES, S-MAC, and Anycast protocol

Protocols compared (240 nodes)	**Average energy per packet (J)**	**Average delay per packet (s)**	**Packet loss ratio (%)**	**Throughput (packet/s)**	**Coverage lifetime (s)**
QoSES (The proposed protocol)	0.0227	0.233	10.3	15	55617
Anycast	0.0289	0.838	18.2	13	45400
S-MAC	0.0405	0.728	25.2	11	38000

4.5.3 Energy Efficient and QoS based Mobile (QoSEM) Sink based Routing Protocol for Clustered Wireless Sensor Networks: Result and Analysis

Based on the developed system model, simulations are carried out using OMNet++ (Varga and Hornig 2008) to evaluate the performance of the proposed QoSEM protocol. Performance of the proposed QoSEM protocol is compared with the two contemporary protocols, static sink and mobile sink. Experimental parameters, such as energy per packet, average delay per packet, average packet loss, throughput, and coverage lifetime are used to evaluate the QoSEM performance.

The following are the details of the simulation setup, energy model, and discussion of the results.

4.5.3.1 Simulation Setup

Simulations are conducted in the sensing area of 200 × 200 m^2 and the number of sensor nodes varies from 20 to 240 for different experiments. Sensor nodes are randomly deployed, and the random deployment is achieved by choosing (x, y) locations based on a uniform distribution. One sink is fixed and located at the center of the network and the other sink is mobile and moves in the network. The simulations are conducted with a communication range which is equal to double of their sensing range. The simulation parameters used are same as given in Table 4.1.

4.5.3.2 Event Summary for QoSEM

The following is the event summary of the main events as used in QoSEM (see Table 4.7).

Table 4.7. Events summary for QoSEM

Sno	Event Name	Description
1.	*Initialization()*	This event is executed in the start, when the node is initialized. It involves initialization of node parameters, some of which include node ID, initial residual energy, random location coordinates etc.
2.	*SS_ANN*	This event is activated when a super-node receives a static sink announcement event. This event is the advertisement by the static sink to the super-nodes in the network. It involves maintaining/updating a multi-hop

Sno	Event Name	Description
		path to that static sink.
3.	*MS_ANN*	This is activated when a mobile sink reaches any new location and needs to advertise itself to the neighboring super-nodes by sending this event.
4.	*SN_ANN*	This event is activated when a CH node receives a super-node announcement of an event. This event is the advertisement by the super-node for the CH nodes in its vicinity. It involves maintaining/updating the multi-hop path to that super-node.
5.	*MS _QW_ANN*	This event is activated when a super-node receives a queue weight announcement from the mobile sink. It involves calculating and sending the required queue weight information to the mobile sink.
6.	*E_DETECT*	This event is activated when a node senses any event. It is a self message.
7.	*SEND_MSG*	This event is activated when a node receives a self message event E_DETECT. It involves sending the data to the CH.
8.	*RECEIVE_MSG*	This event is activated when a CH/normal/super-node receives any message. It involves establishing the communication with the sender node and receives the data.
9.	*SN_ Status*	This event is activated when a node receive *CH_ANN* from some other node. It involves either the change of CH status from *NULL* to new CH or *NULL* to make itself CH.
10.	*End Timer*	An event triggered when the WAIT timer expires. It can be at the completion of any self timer.

4.5.3.3 Energy Model

The same energy model as presented in section 4.5.1.3 is used here.

4.5.3.4 Results and Discussion

Performance of QoSEM is evaluated and compared with contemporary static sink and mobile sink strategies. Metrics used for performance comparison are average energy per packet, network lifetime, throughput, average delay per packet, packet loss ratio and coverage lifetime. Some of the findings of our experiments are as presented below:

a) Average Energy per Packet

In this experiment, average energy per packet is measured by varying the number of sensor nodes (20-240). Average energy consumption per packet for the proposed protocol (see Figure 4.14) is less than the static sink. The reason is that, in static sink nodes near the static sink remains the same and they have to relay the data of the nodes away from the static sink. It results in formation of hotspots or energy holes near the sink; hence, the energy per packet of the node in the static sink is greater. Conversely, in the proposed protocol a combination of static and mobile sink approaches is used, added by the traffic prioritization mechanism by which packets are relayed either to the static sink or mobile sink based on its QoS requirement. In the proposed protocol by virtue of the mobile sink, the energy hole problem is minimal, as nodes near to the sink are high energy nodes and they change over time with mobile sink movement. It results in a balance use of energy across the whole sensor network. Additionally, in the proposed protocol a static sink is used for forwarding only the delay sensitive data which minimizes the possibility of hotspot formation. Energy savings in the proposed protocol is complemented by use of super-nodes which also minimizes the inflow of data towards the node near the static. As only super-nodes on behalf of neighboring nodes, forward the data to the static sink. Thus, the proposed protocol improves the lifetime of the WSN as compared to the static sink. Ultimately, it decreases the average energy per packet in the proposed protocol.

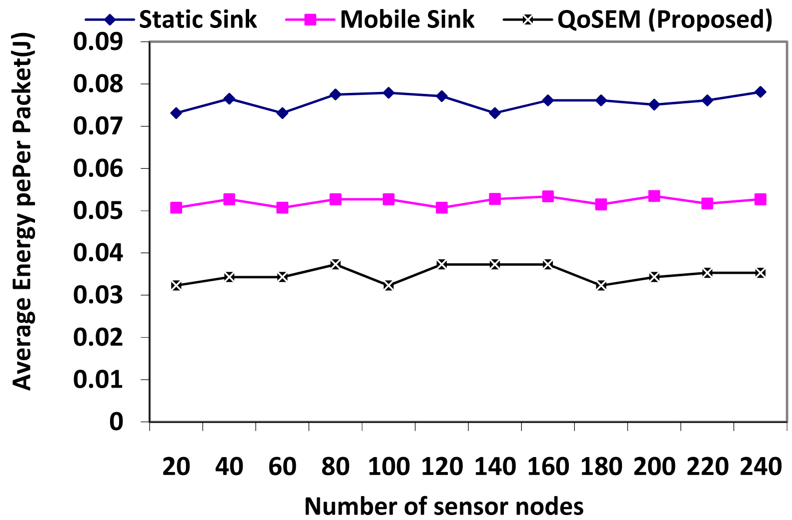


Figure 4.14 Average Energy Per Packet under different numbers of sensor nodes

The mobile sink protocol has also got greater average energy per packet than the proposed protocol. Although in the mobile sink approach, due to sink movement, a big hotspot is not formed around the nodes as the case of a static sink. But still, as in the mobile sink approach; nodes near the sink at each stay point of the mobile sink remain the same throughout the network lifetime. These nodes have repeatedly relayed the data of the nodes away from the mobile sink at that stay point. Thus, the mobile sink is vulnerable to mini energy holes and mini- hotspot problems around the stay points of mobile sink. Hence, energy per packet of the mobile sink approach is greater than the proposed protocol. However, in the proposed protocol a combination of static and mobile sinks is used, this results in a balanced use of sensor node energy across the whole network. Furthermore, energy savings is complemented by the traffic prioritization mechanism and use of super-nodes. Hence, average energy per packet of the proposed protocol is less than the mobile sink approach.

b) Network Lifetime

In this experiment, network lifetime is measured; see Figure 4.15 for the network lifetime corresponding to three protocols. It can be observed that the proposed protocol has the longest $T_{100\%}$, followed by the mobile sink, and the static sink approach has the shortest network lifetime. Indicating that the proposed protocol improves the lifetime of the WSNs as compared to the static sink and mobile sink approaches. The proposed protocol has a greater lifetime than the static sink approach, because in the static sink, all the nodes relay their data through the same nodes which are near the sink. Over time, these nodes expire forming a hotspot/energy hole near the static sink which significantly limits the network lifetime. Whereas, in the proposed protocol, the mobile sink is used in addition to the static sink, which moves across the network to gather data. In this way, nodes near the mobile sink change with every mobile sink movement. Hence, all the nodes use their energy in a balanced way and no energy hole is formed, which lengthens the network lifetime. Furthermore, in the proposed protocol, the balanced use of sensor node energy is complemented by use of the prioritization mechanism by which traffic is routed to the appropriate sink (static or mobile) based on its QoS requirement, which contributes to maximizing its network lifetime.

The mobile sink approach has less network lifetime than the proposed protocol. It is because of the fact that, although the mobile sink approach minimizes the formation of hotspots by its constant sink movement, still at the stay point of the mobile sink, the nodes near to the sink remain same and relay the data on behalf of nodes away from sink. This causes the formation of mini hotspots or energy holes, which minimize the network lifetime, not as adverse as with the static sink approach but it still affects the network lifetime. Whereas, in the proposed protocol data forwarding is solely relying neither on the static sink nor on the mobile sink, rather a combination of two is used which ensures a balanced use of energy of the nodes across the network, resulting in an increase of the network lifetime. Hence, the proposed protocol has better network lifetime than both the static and mobile sink approaches.

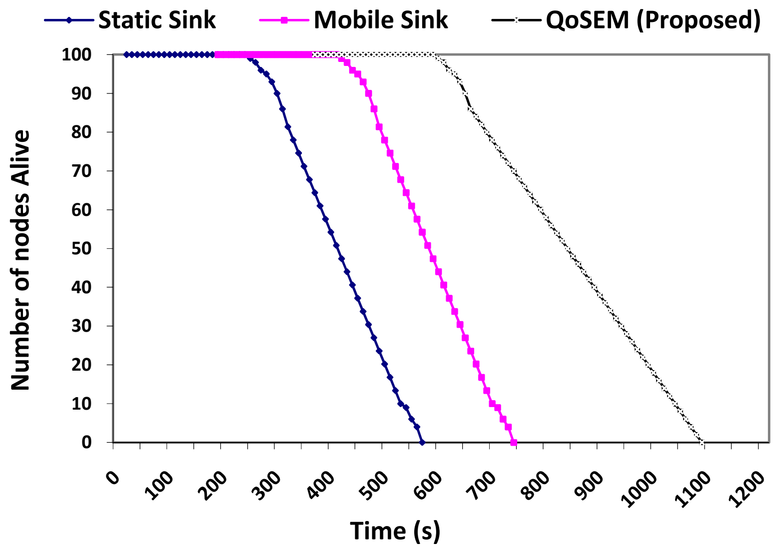


Figure 4.15 Network Lifetime

c) Throughput (Packets per second)

In this experiment, throughput is measured in terms of packets per second by varying the number of sensors from 20 to 220. Figure 4.16, shows the simulation results for all three protocols compared. It can be observed that the proposed protocol outperforms the static sink and mobile sink approaches, in terms of throughput. The reason for that is, in the static sink nodes near the sink remain the same and are vulnerable to energy holes and hotspot problems. Thus, in the static sink approach, data cannot be forwarded to the BS once any energy hole (hotspot) is formed in the network. Therefore, due to the formation of energy holes, multi hop communication cannot be done across that region. It results in limiting the number of packets which can be successfully forwarded to the BS per second (throughput). Whereas in the proposed protocol the energy hole issue is addressed by virtue of a mobile sink, as the immediate neighbor of the sink changes due to regular sink movement. Consequently, the hotspot problem is minimized and more packets can be forwarded to the BS (throughput.

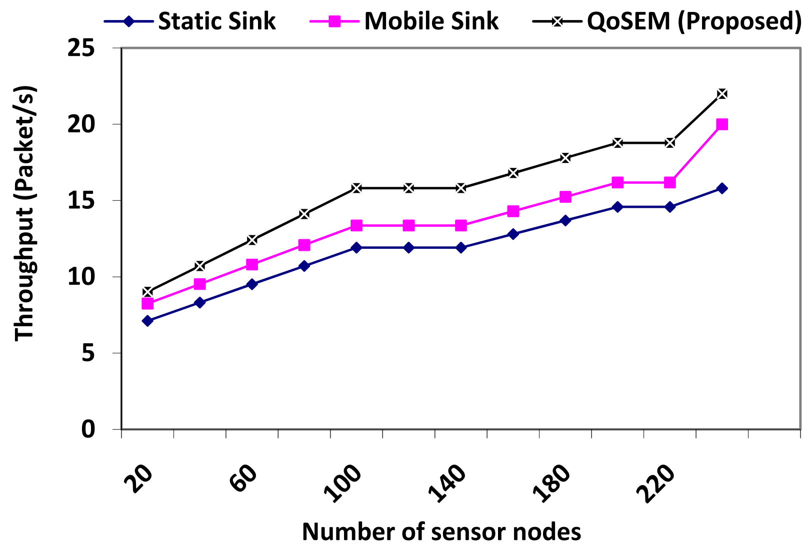


Figure 4.16 Throughput (packet per second) under different numbers of sensor nodes

Mobile sink approaches have also got less throughput than the proposed protocol because in the mobile sink approach, all the nodes have to wait for the mobile sink to come in its vicinity for data forwarding. In the process, some of the packets expire their time to live (TTL) and cannot be forwarded to the mobile sink. It limits the number of packets received by the mobile sink. Whereas in the proposed protocol, as the static sink is employed in addition to the mobile sink and a prioritization mechanism is set in place to make decisions to forward the data according to its QoS requirement onto the static sink or mobile sink. Delay constrained messages in terms of traffic type or content are sent to the static sink and delay tolerant packets are sent to the mobile sink. It ultimately increases the throughput as by virtue of the static sink and prioritization mechanism more packets can make their way to the sink, which otherwise could not be forwarded in the mobile sink approach. Furthermore, in the proposed protocol the combination of static and mobile sinks, avoids the mini hotspots which are generally formed near the mobile sink stay points in the network in the mobile sink approach; moreover, due to this mini hotspot/energy hole phenomenon the number of packets forwarded to the mobile sink at a particular stay point is reduced. Whereas in the proposed protocol, two channels, static sink and mobile sink, are available to the sensor nodes for data forwarding, which increases the number of packets forwarded. This causes the proposed protocol having better throughput than the mobile sink approach.

Consequently, the proposed protocol has better throughput than both of the other two protocols (static sink and mobile sink approaches). Furthermore, it can be seen from Figure 4.16, that the proposed protocol is more scalable and performs better with bigger network size than do the other protocols.

d) Average Delay per Packet

In this experiment, delay values are measured by changing the number of sensor nodes from 20 to 240. As shown in Figure 5.17, the average delay experienced by the proposed protocol is the least, while the static sink is the second and the mobile sink has the worst delay time The static sink approach has got a greater delay than the proposed protocol, because in the static sink approach all nodes have to relay their data all the way to the BS using multi-hop communication, and on the way it involves many relay nodes to reach the BS. Considering the local traffic at each node, the sensed data has to wait for some time at each node to get attention, as well as the time required to forward the data to the next node or BS. Additionally, the static sink approach has got no prioritization mechanism which can treat the incoming traffic based on its QoS requirement. It makes the static sink prone to longer delay. Whereas in the proposed protocol, the combination of static and mobile sinks is used and prioritization of packets is done based on their QoS requirements. Delay sensitive packets are sent to the static sink and delay tolerant to the mobile sink based on the QoS (delay) requirements of each packet. It makes the delay of the proposed protocol less than the static sink approach.

The mobile sink approach has an average delay more than the proposed protocol because in the mobile sink approach nodes have to wait for the mobile sink to come in its vicinity to forward the data. It results in increasing the delay especially for the group of nodes which are visited last by the mobile sink. In the proposed protocol, every node is supposed to send their data to the local sink (super-node) in their vicinity, which means that each packet does not wait long when it is propagated to the static sink or mobile sink. Furthermore, by virtue of the prioritization mechanism the packets are sent either to the static sink or mobile sink based on the QoS requirements of the traffic type and content. That is why the proposed protocol has less delay than the mobile sink approach.

Figure 4.17 shows that, as the number of nodes increases, the difference between the proposed protocol and the other two approaches (static sink and mobile sink) also increases. For the proposed protocol, the performance remains the same for the increased number of nodes, since super-nodes and static sinks are always there to be used as a local sink, whatever the size of the network may be. In this way, increasing the node number has no effect on the proposed protocol. Therefore, it suggests that the proposed protocol is more scalable than the mobile sink and static sink protocols.

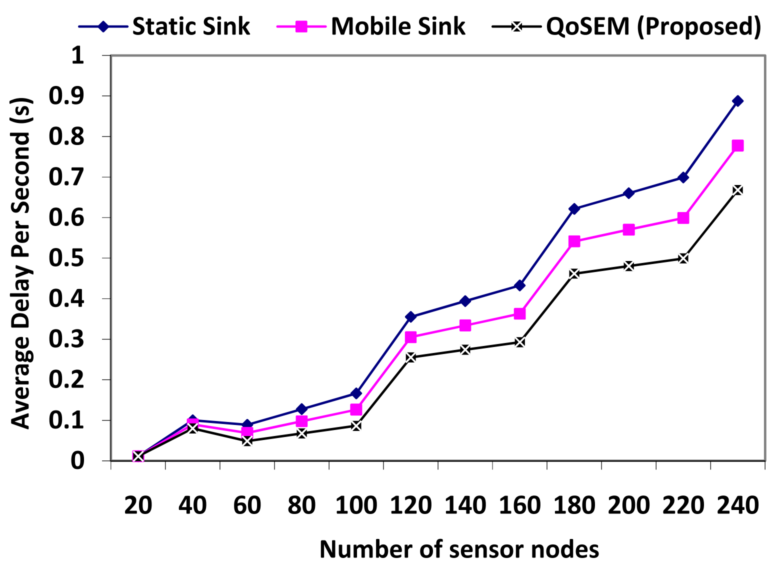


Figure 4.17 Average delay per packet for different number of sensor nodes

e) Packet Lost Ratio (%)

In this experiment, packet loss ratio for the three protocols is measured by varying the numbers of sensor nodes (20-240). It is clear from Figure 4.18 that, the proposed protocol has far less packet loss ratio as compared to the static sink and mobile sink approaches. The proposed protocol has less packet loss ratio, than the static sink approach, because in the static sink, a node needs to send data all the way to an intended BS using multi-hop communication. It involves many relay nodes before data reaches the static sink. It makes the static sink more vulnerable to packet loss, thus reducing transmission reliability. Whereas, in the proposed protocol, packets are forwarded either to the static sink or mobile sink, according to the proposed underlying prioritization mechanism. The priority mechanism forwards the packets according to their QoS requirement based on the traffic content and type. It decreases the packet loss ratio, as packets are treated according to the differentiated QoS requirements and thus, more packets manage to reach the sink.

The mobile sink approach has also got a packet loss ratio greater than the proposed protocol, because in the mobile sink approach, data at any particular node has to wait for the mobile sink to come into its vicinity to forward data. In the process, packets may expire their time to live (TTL), hence cannot be delivered to the mobile sink. Whereas, in the proposed protocol, the static sink is used in addition to the mobile sink, it increases the successful delivery of the packets. As now, packets have two channels to send the data to according to the message priority based on thier QoS requirements. Additionally, at each node, the CH and super-node prioritization of data is done based on the message content and type. Forwarding data to the static or mobile sink is also based upon the QoS the data. Hence, the proposed protocol has less packet loss ratio than the mobile sink approach. Furthermore, in the proposed protocol, nodes are required to forward their data to a local sink (a super-node). The super-node will send the data to the static/mobile sink via multi-hop super-node communications. Hence, the proposed protocol involves far less relay nodes, which makes it less prone to packet loss. As a result, the proposed protocol outperforms the other two protocols in terms of packet loss ratio.

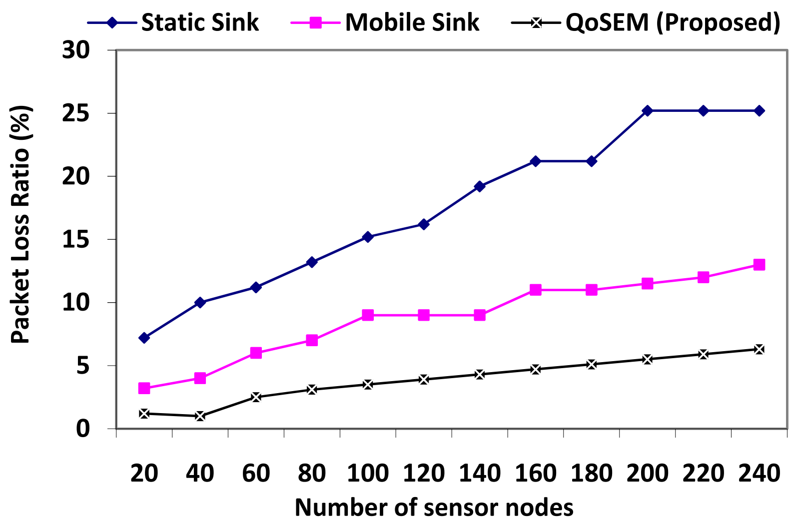
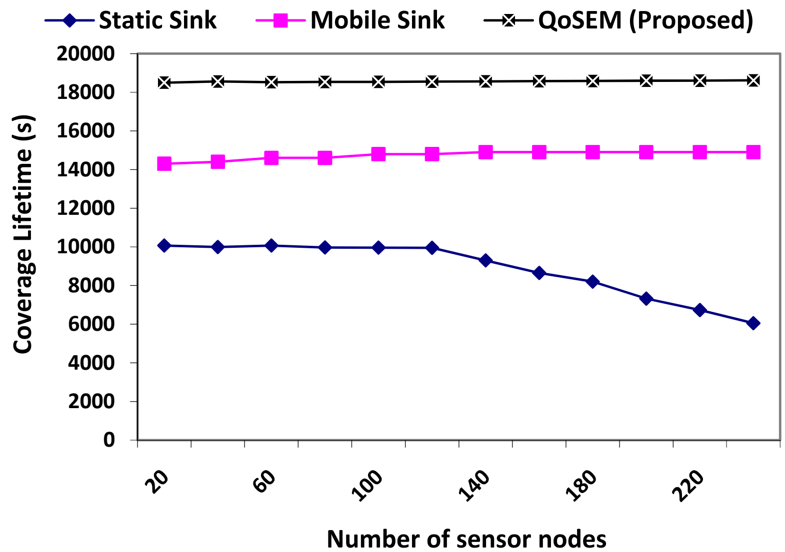


Figure 4.18 Packet loss ratio for different number of sensor nodes

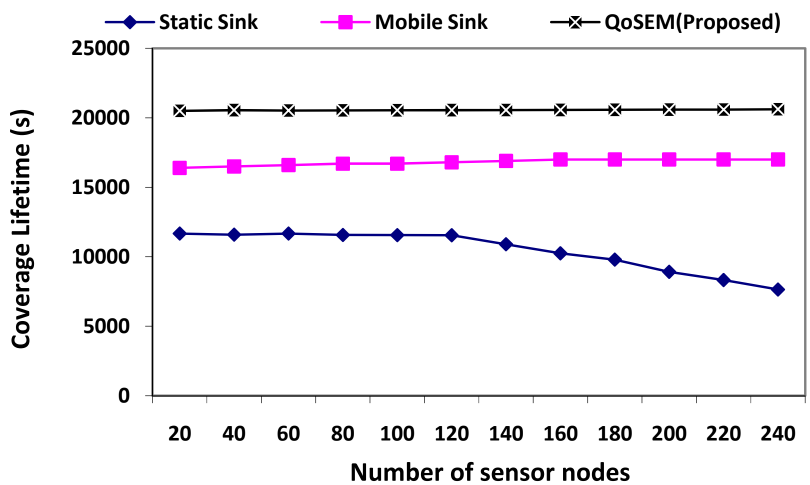
f) Coverage Lifetime

In this experiment, the number of nodes is varied from 20 to 240, and the coverage lifetime (100%, 90%) is measured. Figure 4.19 (a) provides the coverage lifetime for 100% of the coverage area, while 4.19 (b) depicts the coverage lifetime for 90% of the coverage area.

From Figure 4.19 (a) and Figure 4.19 (b), it can be observed that the proposed protocol outperforms the static sink and mobile sink in terms of coverage lifetime in both cases of 100% and 90% coverage. The proposed protocol has a coverage lifetime better than the static sink approach, because in the static sink approach nodes have to relay the data all the way to the BS. Consequently, nodes near the sink have to relay the data on behalf of almost the whole network. It makes nodes near the static sink die earlier, resulting in creation of a coverage hole. It considerably decreases the coverage lifetime. Whereas, in the proposed protocol, use of a mobile sink avoids the formation of a coverage hole near the sink. As with the mobility of the mobile sink, nodes near the sink change resulting in a balanced use of node energy across the whole network. Furthermore, in the proposed protocol, use of static and mobile sinks distribute traffic between the static and mobile sinks based on its distinct QoS requirement, hence ensuring balanced use of energy across the whole network. Additionally, a traffic prioritization mechanism ensures the use of sensor node energy for relaying important data. Using these heuristics, unlike the static sink approach, the proposed protocol denies formation of any energy or coverage hole and ultimately increases coverage lifetime.



(a) Coverage Lifetime considering 100% coverage



(b) Coverage Lifetime considering 90% coverage

Figure 4.19 Coverage lifetime for different numbers of sensor nodes: a) 100% coverage and b) 90% coverage

The proposed protocol has a coverage lifetime better than the mobile sink approach. The reason is that, in the mobile sink approach, mini hotspots are formed near the stay points of the mobile sink. At all stay points of the mobile sink, nodes near to the mobile sink have to relay the data of the nodes away from the mobile sink. It results in formation of small hotspot/energy holes at all the stay points of the mobile sink across the network. Ultimately, these energy holes emerge as coverage holes and limit the coverage lifetime of the mobile sink approach. Whereas, in the proposed protocol these mini-hotspots/energy holes are denied by using a static sink in addition to a mobile sink. It results in a balanced use of energy of all sensor nodes in the network. Ultimately, no coverage hole is formed, which results in increasing the coverage lifetime of the network when compared with the mobile sink approach. As a result, the proposed protocol outperforms the other two protocols in terms of coverage lifetime.

4.5.3.5 QoSEM: Summary of results

This section, discussed the simulations carried out in OMNeT++ to evaluate the performance of the proposed QoSEC protocol and the results are compared against the contemporary static sink and mobile sink protocols. Simulation results demonstrate that QoSEM has achieved significant energy savings and enhanced the network lifetime. The use of a combination of static and mobile sinks and the inclusion of a prioritization mechanism based on traffic content and type in QoSEM have helped the sensor nodes to disseminate their energy in a much more balanced rate, decreased delay and provided QoS to all type of traffic. The results demonstrate that QoSEM is effective in improving QoS parameters, such as average delay, packet loss ratio, and throughput, when compared with static sink and mobile sink approaches. Table 4.8 summarizes the findings from this research work.

Table 4.8. Summary of performance comparison between static sink, mobile sink and proposed protocol

Protocols compared (240 nodes)	**Average energy per packet (J)**	**Average delay per packet(s)**	**Packet loss ratio (%)**	**Throughput (packet/s)**	**Coverage lifetime (s) (100%)**
QoSEM (The proposed protocol)	0.0453	0.668	6.3	22	18617
Mobile Sink	0.05271	0.778	13	20	14900
Static Sink	0.0861330	0.888	25.2	15.8	6050

4.6 Summary

This chapter has presented different types of network simulators, and selection criteria for WSN simulator. It has also discussed important characteristic of the OMNeT framework and WSN simulation design in OMNet++. The rest of the chapter discusses the simulation results and analysis of the three proposed protocols, i.e., QoSEC, QoSES and QoSEM, in terms of performance metrics including average energy per packet, average delay per packet, packet loss ratio, throughput, network lifetime and coverage lifetime.

CHAPTER 5

CONCLUSION AND FUTURE WORK

This chapter concludes this thesis with the conclusion of the research work in the QoS based Energy Efficient Routing in Wireless Sensor Networks. It outlines the research work that has been done, the major contributions, and lastly, the highlights of the recommendation for future work.

5.1 Thesis Summary

Sensor nodes have limited energy and low bandwidth constraints. These constraints are added to by large scale deployment and a distributed nature, which have made energy aware data gathering as a core design issue at different layers of a network protocol stack including the network layer itself. Due to the large scale, random deployment, limited hardware resources, un-rechargeable battery power, hostile environment and failure prone nature, routing in WSNs is a challenging area in WSNs. In routing, most of the protocols found in literature, consider energy awareness as a key design issue to maximize the network. Factors like throughput, latency and delay are not issues of primary concern in these protocols and the approach is acceptable; this is because they mostly deal with a small amount of data flowing in low rates. However, with the emergence of new WSN applications which involve multimedia and imagining sensors, routing in a WSN has come across new challenges. Reporting of data in these multimedia and imaging WSN applications, requires minimum end-to-end delay within an acceptable limit. In such applications, in addition to energy efficiency; latency, throughput and delay also become issues of primary concern. Such performance metrics are usually referred to as the quality of service (QoS) of the communication network. Generally, these applications deal with real time data and need a certain bandwidth with the minimum possible delay. In such scenarios, to guarantee the reliable delivery of the real-time data, a service differentiation mechanism is needed. QoS based energy efficient routing in WSNs will provide the energy efficient path as well as guarantee a certain bandwidth with minimum possible delay.

In this thesis, the QoS based energy efficient routing protocol for wireless sensor networks is presented, which can extend the network lifetime as well enable fulfilling the QoS requirements of the data. The proposed protocols consider energy efficiency as well as QoS as primary design objectives. To achieve QoS and energy efficiency in this thesis, the main problem is divided in to three sub-problems, namely: clustering, sleep/wake and mobile sink protocols. To achieve QoS and energy efficiency three protocols are developed, namely: the QoS based energy efficient clustering (QoSEC) protocol, QoS based energy efficient sleep/wake (QoSES) scheduling protocol and QoS based energy efficient mobile sink (QoSEM) based routing protocol for WSNs. Extensive simulations are carried out to evaluate the performance of the proposed protocols, namely: QoSEC, QoSES, and QoSEM; their performances have been compared with state of the art contemporary approaches. The results demonstrate that QoSEC, QoSES, and QoSEM have successfully minimized the end-to-end delay, as well as improved the other QoS routing parameters like coverage lifetime, network lifetime, average energy per packet, packet loss ratio, and throughput.

5.2 Conclusion

In this thesis, to achieve QoS based energy efficiency routing in a WSN, three protocols are developed namely, the QoS based energy efficient clustering (QoSEC) protocol, the QoS based energy efficient sleep/wake (QoSES) scheduling protocol and the QoS based energy efficient mobile sink (QoSEM) based routing protocol for WSNs.

Firstly, an energy-efficient clustering-based data gathering and routing technique for the WSN is proposed; it is called the QoS Based Energy Efficient Clustering (QoSEC) Protocol for WSNs. With QoSEC, to achieve energy efficiency that prolongs network and coverage lifetime, some nodes with additional energy resources, termed as super-nodes, in addition to normal capability nodes, are deployed. Multi-hierarchy clustering is then implemented by placing super-nodes (acting as local sinks) at the top tier, the cluster head (normal node) at the middle tier, and a cluster member (normal node) at the lowest tier in the hierarchy. Clustering with normal sensor nodes is done by optimizing the network/coverage lifetime. Specifically, the improvement on energy efficiency and coverage lifetime is achieved through the implementation of a *cluster-head-selection* algorithm and a *sleep/wake scheduling* algorithm. In summary, with QoSEC, the hot spot problem in the WSN is resolved, and importantly the network/coverage lifetime of the WSN is prolonged. Simulations have been carried out in OMNet++ to evaluate the performance of the proposed QoSEC protocol and the results are compared against the contemporary LEACH, EEMC and CPCP protocols. Simulation results demonstrate that QoSEC has achieved significant energy savings and enhanced the network lifetime, as well as the coverage lifetime. The use of super-nodes and the inclusion of multi-parameters (energy, coverage) in the cluster-head-selection algorithm and sleep/wake scheduling algorithm in QoSEC have helped the sensor nodes to dissipate their energy in a much more balanced rate, while at the same time kept good coverage of the sensing area. The other protocols are not equipped with these capabilities. The results produced also demonstrate that QoSEC is effective in improving the other QoS parameters, such as average delay, packet loss ratio, and throughput.

Secondly, in this thesis, The QoS Based Energy Efficient Sleep/Wake Scheduling (QoSES) Protocol for WSNs is proposed. To reduce delay, QoSES identifies nodes for different sleep/wake periods according to their traffic load at three levels: a) Nodes with different sleep/wake schedule requirements according to their variable traffic load based on their distance from BS. b) Nodes with different sleep/wake schedule requirements according to their different traffic load based on their topological importance in the network. c) Nodes with different sleep/wake schedules based on handling burst traffic in the proximity of the event occurrence node. It then assigns different active intervals to the nodes, according to variable traffic load requirements defined by the node position in the network, the node topological importance, and by handling burst traffic in the proximity of the event occurrence node. Using these heuristics, QoSES reduces end-to-end delay and maximizes the throughput by minimizing the congestion at nodes having a heavy traffic load. Simulations are carried out to evaluate the performance of the proposed protocol, by comparing its performance with S-MAC and anycast protocols. Simulation results demonstrate that the proposed protocol has significantly reduced the end-to-end delay, as well as improved the other QoS parameters of average energy per packet, average delay, packet loss ratio, throughput and coverage throughput.

Thirdly and lastly, in this thesis, an energy-efficient and QoS based routing technique for the WSN is proposed and is called the QoS based Energy Efficient Mobile Sink (QoSEM) based Routing Protocol for the Clustered Wireless Sensor Network. With QoSEM, to ensure QoS for different traffic types, prioritization of data is done based on message type and content. Data forwarding decisions are based on the priority of each packet, to ensure that the message is treated fairly as per their distinct QoS requirement. To address energy efficiency (hot spot problems) and high end-to-end delay problems, a combination of a mobile and static sink is used for data gathering. Delay sensitive messages are sent through the static sink and delay tolerant messages are sent through the mobile sink. Furthermore, to minimize delay in mobile sink data gathering, movement of the mobile sink is associated with the priority messages which nodes need to send i.e. in each cycle, a mobile sink moves across the whole network visiting different nodes based on the priority of the packets to be forwarded. In this way, the proposed protocol incurs less end-to-end delay, is energy efficient and able to ensure QoS. In summary, with QoSEM, the hot spot problems in WSNs are resolved with an increase in network lifetime, and more importantly delay and QoS is improved. Simulations have been carried out in OMNet++ to evaluate the performance of the proposed QoSEM protocol and the results compared against the contemporary mobile sink and mobile sink protocols. Simulation results demonstrate that QoSEM has achieved significant energy savings and enhanced the network lifetime. The use of a combination of static and mobile sinks, and the inclusion of a prioritization mechanism based on traffic content and type in QoSEM have helped the sensor nodes to disseminate their energy at a much more balanced rate, decreased delay, and it provides for QoS to all types of traffic. The other protocols are not equipped with these capabilities. The results produced also demonstrate that QoSEM is effective in improving the other QoS parameters, such as average delay, packet loss ratio, and throughput.

5.3 Research Contributions

In this thesis, QoS aware Energy Efficient Routing in WSN is considered and three protocols: QoSEC, QoSES and QoSEM were proposed. The expected research contributions from this research work have been discussed in section 1.10. However those contributions can also be expressed in terms of improvement in the performance parameters observed during the performance evaluation experiments. In this context, the main contributions of the thesis can be stated as follows:

- *Energy Efficiency:* The proposed protocols able to improve the energy efficiency by delegating resource intensive CH job to higher energy nodes, minimizing hot spot problem by using notion of super node and by giving longer active intervals to higher energy nodes. Furthermore, energy efficiency is complemented by using combination of mobile sink and static mobile sink which ensured the balanced used of sensor node energy across the network.
- *Network Lifetime:* The proposed protocols increased the network lifetime by minimizing the hot spot problem using the notion of super node, mobile sink and saved energy wastage by having efficient sleep/wake scheduling. Furthermore, use of static sink in combination with mobile sink evenly distributes the traffic among the sensor node in network. It ensures balanced use of sensor node energy across the network, which ultimately helped in prolonging network lifetime.
- *Delay improvement:* The proposed protocols achieved improvement in delay by using traffic specific sleep/wake scheduling which decrease the schedule miss and latency involved in multi-hop communication. Furthermore, use of super nodes and mobile sink helped in decreasing delay by providing local sinks near the sensor node, hence, avoiding time consuming multi-hop communication.
- *Packet Loss Ratio:* With proposed protocols packet loss ratio is decreased, as by virtue of super node and mobile sink multi-hop communication is minimized which helped in decreasing the collision and decreasing the packet loss ratio.

Furthermore, sleep/wake scheduling by using variable sleep/wake schedule decrease the schedule miss which contributed in decreasing packet loss ratio.

- *Throughput:* The proposed protocols achieved improvement in throughput by using multi-hierarchy clustering with super nodes act as local sink which decreases the hop involved to send the data to the BS, hence, resulted in improving throughput. Furthermore, improvement in throughput is complemented by using traffic specific sleep/wake scheduling (which decrease the schedule miss) and finally use of combination of mobile sink and static mobile sink helped in maximizing the throughput.
- *Coverage Lifetime:* Coverage lifetime is increased in the proposed protocol as in CH selection coverage of the sensing area is considered as one of the parameter in CH selection. Furthermore, efficient utilization of energy during sleep/wake scheduling and mobile sink data gathering helped in prolonging coverage lifetime.

5.4 Future Work

This section briefly discusses the future directions of the QoS based energy efficient routing in wireless sensor networks. The following are a few interesting research directions where work presented in this thesis can be extended in the future.

- *Fault Tolerance:* The plan is to test the behavior of the proposed protocols considering the fault tolerance feature and to observe its impact on the performance of the proposed protocols.

- *Node relocation:* It is also anticipated that further improvement in energy efficiency and QoS can be achieved by considering the concept of limited mobility of the sensor nodes, where nodes are relocated to improve QoS and energy efficiency.

- *Data Aggregation:* Another interesting research direction will be using the proposed protocols in conjunction with a QoS based energy efficient data aggregation mechanism and observing its impact on the performance of the proposed protocols.

- *Node's mobility:* In this work, static nodes are considered and it would be an interesting research direction to see the effect of mobility on the workings and performance of the proposed protocols. Mobile nodes introduce new challenges like highly changing topology and frequent routing path breakage for the proposed protocols. They require a self healing feature, so that, the proposed protocols will be able to maintain a reasonable QoS with all these frequent changes in the network topology.

REFERENCES

http://sce.uhcl.edu/yang/public/download/OMNET++.pdf. (accessed on 19 April, 2011).

Abbasi AA, Akkaya K, and Younis M. (2007). A Distributed Connectivity Restoration Algorithm in Wireless Sensor and Actor Networks. In Local Computer Networks, 2007 LCN 2007 32nd IEEE Conference on; 15-18 Oct. 2007. p 496-503.

Abtin K, Huang L, and Lakshmi V. (2006). Wakeup scheduling in wireless sensor networks. Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing. Florence, Italy: ACM.

Akyildiz IF, Vurany MC, Akan OB, and Su W. (2006). Wireless Sensor Networks: A Survey Revisited. COMPUTER NETWORKS JOURNAL (ELSEVIER SCIENCE).

Akyildiz IF, Weilian S, Sankarasubramaniam Y, and Cayirci E. (2002). A survey on sensor networks. Communications Magazine, IEEE 40(8). p 102-114.

Al-Karaki JN, and Kamal AE. (2004). Routing techniques in wireless sensor networks: a survey. Wireless Communications, IEEE 11(6). p 6-28.

Alemdar H, and Ersoy C. (2010). Wireless sensor networks for healthcare: A survey. Computer Networks 54(15). p 2688-2710.

Anastasi G, Conti M, Di Francesco M, and Passarella A. (2009). Energy conservation in wireless sensor networks: A survey. Ad Hoc Networks 7(3). p 537-568.

Andras V. (2008). OMNeT++ Discrete Event Simulation System. Available: http://www.omnetpp.org. (accessed on 19 April, 2011).

Armstrong T. "Wake-up Based Power Management in Multi-hop Wireless Networks", . Term Survey Paper for ECE1717 Quality of Service Provisioning in Mobile Networks course available at url: http://wwweecgtorontoedu/ trevor/Wakeup/indexhtml.

Atkinson R, and D. C. (October 2008). Digital Quality of Life – Understanding the Personal & Social Benefits of the Information Technology Revolution. The Information Technology and Innovation Foundation, Washington DC.

Baker D, and Ephremides A. (1981). The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm. Communications, IEEE Transactions on 29(11). p 1694-1701.

Baker D, Ephremides A, and Flynn J. (1984). The Design and Simulation of a Mobile Radio Network with Distributed Control. Selected Areas in Communications, IEEE Journal on 2(1). p 226-237.

Banerjee S, and Khuller S. (2001). A clustering scheme for hierarchical control in multi-hop wireless networks. In INFOCOM 2001 Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings IEEE; 2001. p 1028-1037 vol.1022.

Banerjee T, and Agrawal DP. (2008). Increasing Lifetime of Wireless Sensor Networks Using Controllable Mobile Cluster Heads. In Performance, Computing and Communications Conference, 2008 IPCCC 2008 IEEE International; 7-9 Dec. 2008. p 77-84.

Barbara H, Lance D, and Eric B. (2004). Flexible power scheduling for sensor networks. Proceedings of the 3rd international symposium on Information processing in sensor networks. Berkeley, California, USA: ACM.

Cardei M, Yinying Y, and Jie W. (2008). Non-uniform sensor deployment in mobile wireless sensor networks. In World of Wireless, Mobile and Multimedia Networks, 2008 WoWMoM 2008 2008 International Symposium on a; 23-26 June 2008. p 1-8.

Chandresh Pratap S, Vyas OP, and Manoj Ku T. (2008). A Survey of Simulation in Sensor Networks. Proceedings of the 2008 International Conference on Computational Intelligence for Modelling Control & Automation: IEEE Computer Society.

Cormen TH, Leiserson CE, Rivest RL, and Stein C. (2001). Introduction to Algorithms: , MIT Press,.

Curren D. (2006). “A survey of simulation in sensor networks”, [online], http://www.cs.binghamton.edu/~kang/teaching/cs580s/. (accessed on 26 March, 2011).

Dai F, and Wu J. (2004). An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. Parallel and Distributed Systems, IEEE Transactions on 15(10). p 908-920.

Demirbas M, Arora A, and Mittal V. (2004). FLOC: a fast local clustering service for wireless sensor networks. In Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS'04); June 2004; Palazzo dei Congressi, Florence, Italy.

Ding P, Holliday J, and Celik A. (2005). Distributed energy efficient hierarchical clustering for wireless sensor networks. In EEE International Conference on Distributed Computing in Sensor Systems(DCOSS'05); June 2005; Marina Del Rey, CA IEEE.

Edgar H, and Callaway J. (2003). Wireless Sensor Networks: architectures and protocols: a CRC press company, Auerbach publications.

Egea-Lopez E, Vales-Alonso J, Martinez-Sala AS, Pavon-Marino P, and Garcia-Haro J. (2005.). Simulation Tools for Wireless Sensor Networks. In Summer Simulation Multiconference – SPECTS

Estrin D, Govindan R, and Heidemann J. (1999). Next Century Challenges: Scalable Coordination in Sensor Networks. . In Fifth Annual International Conference on Mobile Computing and Networks (MobiCom 1999); Aug. 1999; Seattle, Washington, DC.

Gandham SR, Dawande M, Prakash R, and Venkatesan S. (2003). Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In Global Telecommunications Conference, 2003 GLOBECOM '03 IEEE; 1-5 Dec. 2003. p 377-381 Vol.371.

Hasbullah H, and Nazir B. (2010). Region-based Energy-aware cluster (REC) for efficient packet forwarding in WSN. In Information Technology (ITSim), 2010 International Symposium in; 15-17 June 2010. p 1-6.

Heinzelman WR, Chandrakasan A, and Balakrishnan H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In System Sciences, 2000 Proceedings of the 33rd Annual Hawaii International Conference on; 4-7 Jan. 2000. p 10 pp. vol.12.

Hui L, Zhijun M, and Shuanghu C. (2007). A Wireless Sensor Network Prototype for Environmental Monitoring in Greenhouses. In Wireless Communications, Networking and Mobile Computing, 2007 WiCom 2007 International Conference on; 21-25 Sept. 2007. p 2344-2347.

Jennifer Y, Biswanath M, and Dipak G. (2008). Wireless sensor network survey. Comput Netw 52(12). p 2292-2330.

Jin Y, Wang L, Kim Y, and Yang X. (2008). EEMC: An energy-efficient multi-level clustering algorithm for large-scale wireless sensor networks. Computer Networks 52(3). p 542-562.

Joohwan K, Xiaojun L, Shroff NB, and Sinha P. (2008). Minimizing Delay and Maximizing Lifetime for Wireless Sensor Networks With Anycast. Networking, IEEE/ACM Transactions on 18(2). p 515-528.

Jue Y, Chengyang Z, Xinrong L, Yan H, Shengli F, and Miguel FA. (2009). Integration of wireless sensor networks in environmental monitoring cyber infrastructure. Wirel Netw 16(4). p 1091-1108.

Jun L, and Hubaux JP. (2005). Joint mobility and routing for lifetime elongation in wireless sensor networks. In INFOCOM 2005 24th Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings IEEE; 13-17 March 2005. p 1735-1746 vol. 1733.

Kahn JM, Katz RH, and Pister KSJ. (1999). Next century challenges: mobile networking for \“Smart Dust\&rdquo. Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking. Seattle, Washington, United States: ACM.

Kaixin X, and Gerla M. (2002). A heterogeneous routing protocol based on a new stable clustering scheme. In MILCOM 2002 Proceedings; 7-10 Oct. 2002. p 838-843 vol.832.

Karl H, and Willig A. (2005). Protocols and Architectures for Wireless Sensor Networks: John Wiley & Sons.

Karl H, Wolisz A, Willig A, Chan H, and Perrig A. (2004). ACE: An Emergent Algorithm for Highly Uniform Cluster Formation. Wireless Sensor Networks: Springer Berlin / Heidelberg. p 154-171.

Ke W, Ayyash SA, Little TDC, and Basu P. (2005). Attribute-based clustering for information dissemination in wireless sensor networks. In Sensor and Ad Hoc Communications and Networks, 2005 IEEE SECON 2005 2005 Second Annual IEEE Communications Society Conference on; 26-29 Sept., 2005. p 498-509.

Konstantas D, Jones V, and Herzog R. (2002). MobiHealth - innovative 2.5 / 3G mobile services and applications for healthcare. In Proceedings of the Eleventh Information Society Technologies (IST) Mobile and Wireless Telecommunications.

Lee WL, Datta A, and Cardell-Oliver R. (2008). FlexiTP: A Flexible-Schedule-Based TDMA Protocol for Fault-Tolerant and Energy-Efficient Wireless Sensor Networks. Parallel and Distributed Systems, IEEE Transactions on 19(6). p 851-864.

Lessmann J, Janacik P, Lachev L, and Orfanus D. (2008). Comparative Study of Wireless Network Simulators. In Networking, 2008 ICN 2008 Seventh International Conference on; 13-18 April 2008. p 517-523.

Lin CR, and Gerla M. (1997). Adaptive clustering for mobile wireless networks. Selected Areas in Communications, IEEE Journal on 15(7). p 1265-1275.

Lin G, and John AS. (2005). Radio-Triggered Wake-Up for Wireless Sensor Networks. Real-Time Syst 29(2-3). p 157-182.

López Riquelme JA, Soto F, Suardíaz J, Sánchez P, Iborra A, and Vera JA. (2009). Wireless Sensor Networks for precision horticulture in Southern Spain. Computers and Electronics in Agriculture 68(1). p 25-35.

Lorincz K, Malan DJ, Fulford-Jones TRF, Nawoj A, Clavel A, Shnayder V, Mainland G, Welsh M, and Moulton S. (2004). Sensor networks for emergency response: challenges and opportunities. Pervasive Computing, IEEE 3(4). p 16-23.

Ma J, Bi Y, Li LSN, Khan IA, and Chen C. (2007). HUMS: an autonomous moving strategy for mobile sinks in data-gathering sensor networks. EURASIP Journal of Wireless Communication & Networking, Special Issue on Novel Techniques for Analysis & Design Volume 2007 (Article ID 64574). p 15 pages.

Mallanda C, Suri A, Kunchakarra V, Iyengar SS, Kannan R, Durresi A, and Sastry S. (2005). Simulating Wireless Sensor Networks with OMNeT++ From citeulike-article-id:1005959[online] http://csc.lsu.edu/sensor_web/final%20papers/OMNet++-IEEE-Computers.pdf.

Marko K, Mikko S, Niilo K, rkk, inen, and Pirkka T. (2009). Survey of Wireless Sensor Networks Simulation Tools for Demanding Applications. Proceedings of the 2009 Fifth International Conference on Networking and Services: IEEE Computer Society.

Marta M, and Cardei M. (2009). Improved sensor network lifetime with multiple mobile sinks. Pervasive and Mobile Computing 5(5). p 542-555.

Mehdi M, and Bernard M. (2008). A Survey on Sensor Webs Simulation Tools. Proceedings of the 2008 Second International Conference on Sensor Technologies and Applications: IEEE Computer Society.

Muruganathan SD, Ma DCF, Bhasin RI, and Fapojuwo AO. (2005). A centralized energy-efficient routing protocol for wireless sensor networks. Communications Magazine, IEEE 43(3). p S8-13.

Myers BA, Willingham JB, Landy P, Webster MA, Frogge P, and Fischer M. (2000). Design considerations for minimal-power wireless spread spectrum circuits and systems. Proceedings of the IEEE 88(10). p 1598-1612.

Nagpal R, and Coore D. (1998). An algorithm for group formation in an amorphous computer. In 10th International Conference on Parallel and Distributed Systems (PDCS'98); October 19, 1999; Las Vegas, NV,.

Nazir B, and Hasbullah H. (2010). Energy balanced clustering in wireless sensor network. In Information Technology (ITSim), 2010 International Symposium in; 15-17 June 2010. p 569-574.

Ng J, Lo B, Wells O, Sloman M, Peters N, Darzi A, Toumazou C, and Yang G. (2004). Ubiquitous Monitoring Environment for Wearable and Implantable Sensors (UbiMon). UbiComp'04 -- The Sixth International Conference on Ubiquitous Computing, Poster Proceedings: UbiComp'04.

Olariu S. (2006). Sensor Networks – A Short Course.: Old Dominion University.

Paruchuri V, Basavaraju S, Durresi A, Kannan R, and Iyengar SS. (2004). Random asynchronous wakeup protocol for sensor networks. In Broadband Networks, 2004 BroadNets 2004 Proceedings First International Conference on; 25-29 Oct. 2004. p 710-717.

Pierce FJ, and Elliott TV. (2008). Regional and on-farm wireless sensor networks for agricultural systems in Eastern Washington. Computers and Electronics in Agriculture 61(1). p 32-43.

Raul M, Miguel AF, Samuel GM, Carlos S, dio, Ferreira PJSG, and Reis MJCS. (2008). A ZigBee multi-powered wireless acquisition device for remote sensing applications in precision viticulture. Comput Electron Agric 62(2). p 94-106.

Rong Z, Jennifer CH, and Lui S. (2003). Asynchronous wakeup for ad hoc networks. Proceedings of the 4th ACM international symposium on Mobile ad hoc networking \& computing. Annapolis, Maryland, USA: ACM.

Saad EM, Awadalla MH, and Darwish RR. (2008). A Data Gathering Algorithm for a Mobile Sink in Large-Scale Sensor Networks. In Wireless and Mobile Communications, 2008 ICWMC '08 The Fourth International Conference on; July 27 2008-Aug. 1 2008. p 207-213.

Sang Hyuk L, Soobin L, Heecheol S, and Hwang Soo L. (2009). Wireless sensor network design for tactical military applications : Remote large-scale environments. In Military Communications Conference, 2009 MILCOM 2009 IEEE; 18-21 Oct. 2009. p 1-7.

Schurgers C, Tsiatsis V, Ganeriwal S, and Srivastava M. (2002a). Optimizing sensor networks in the energy-latency-density design space. Mobile Computing, IEEE Transactions on 1(1). p 70-80.

Schurgers C, Tsiatsis V, and Srivastava MB. (2002b). STEM: Topology management for energy efficient sensor networks. In Aerospace Conference Proceedings, 2002 IEEE; 2002. p 3-1099-1093-1108 vol.1093.

Seema B, and Coyle EJ. (2003). An energy efficient hierarchical clustering algorithm for wireless sensor networks. In INFOCOM 2003 Twenty-Second Annual Joint Conference of the IEEE Computer and Communications IEEE Societies; 30 March-3 April 2003. p 1713-1723 vol.1713.

Shah RC, Roy S, Jain S, and Brunette W. (2003). Data MULEs: modeling a three-tier architecture for sparse sensor networks. In Sensor Network Protocols and Applications, 2003 Proceedings of the First IEEE 2003 IEEE International Workshop on; 11 May 2003. p 30-41.

Shargal M, and Houseman D. (2009a) 5 March, 2009. The Big Picture of Your Coming Smart Grid. Smart Grid News.

Shargal M, and Houseman D. (2009b) 24 February, 2009. Why Your Smart Grid Must Start with Communications. Smart Grid News.

Somasundara AA, Kansal A, Jea DD, Estrin D, and Srivastava MB. (2006). Controllably mobile infrastructure for low energy embedded networks. Mobile Computing, IEEE Transactions on 5(8). p 958-973.

Soro S, and Heinzelman WB. (2009). Cluster head election techniques for coverage preservation in wireless sensor networks. Ad Hoc Networks 7(5). p 955-972.

Tavares J, Velez FJ, and Ferro JM. (2008). Application of Wireless Sensor Networks to Automobiles. Measurement Science Review 8(3). p 65-70.

Tijs van D, and Koen L. (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.

Varga A, and Hornig R. (2008). An overview of the OMNeT++ simulation environment. Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops. Marseille, France: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). p 1--10.

Veloso M, Bentos C, and Pereira FC. (2009). Multi-Sensor Data Fusion on Intelligent Transport Systems",. MIT Portugal Transportation Systems Working Paper Series, ITS-CM-09-02.

Vieira MAM, Coelho CN, Jr., da Silva DC, Jr., and da Mata JM. (2003). Survey on wireless sensor network devices. In Emerging Technologies and Factory Automation, 2003 Proceedings ETFA '03 IEEE Conference; 16-19 Sept. 2003. p 537-544 vol.531.

Wang AY, SeongHwan C, Sodini CG, and Chandrakasan AP. (2001). Energy efficient modulation and MAC for asymmetric RF microsensor systems. In

Low Power Electronics and Design, International Symposium on, 2001; 2001. p 106-111.

Wang G, Wang T, Jia W, Guo M, and Li J. (2009). Adaptive location updates for mobile sinks in wireless sensor networks. The Journal of Supercomputing 47(2). p 127-145.

Wark T, Corke P, Sikka P, Klingbeil L, Ying G, Crossman C, Valencia P, Swain D, and Bishop-Hurley G. (2007). Transforming Agriculture through Pervasive Wireless Sensor Networks. Pervasive Computing, IEEE 6(2). p 50-57.

Waterman J, Curtis D, Goraczko M, Shih E, Sarin P, Pino E, Ohno-Machado L, Greenes R, Guttag J, and Stair T. (2005). Demonstration of SMART (Scalable Medical Alert Response Technology). In AMIA 2005 Annual Symposium Washington DC. American Medical Informatics Association.

Weber V, and Vickery G. (2009). Smart Sensor Networks: Technologies and Applications for Green Growth. OECD Digital Economy Papers 167, OECD Publishing.

Wei Y, Heidemann J, and Estrin D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In INFOCOM 2002 Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings IEEE; 2002. p 1567-1576 vol.1563.

Xiaodong X, Weiren S, and He H. (2008). Comparison of OMNET++ and other simulator for WSN simulation. In Industrial Electronics and Applications, 2008 ICIEA 2008 3rd IEEE Conference on; 3-5 June 2008. p 1439-1443.

Xue Y, and Vaidya NH. (2004). A wakeup scheme for sensor networks: achieving balance between energy saving and end-to-end delay. In Real-Time and Embedded Technology and Applications Symposium, 2004 Proceedings RTAS 2004 10th IEEE; 25-28 May 2004. p 19-26.

Yang Y, Fonoage MI, and Cardei M. (2010). Improving network lifetime with mobile wireless sensor networks. Computer Communications 33(4). p 409-419.

Yanzhong B, Jianwei N, Limin S, Wei H, and Yi S. (2007). Moving Schemes for Mobile Sinks in Wireless Sensor Networks. In Performance, Computing, and Communications Conference, 2007 IPCCC 2007 IEEE Internationa; 11-13 April 2007. p 101-108.

Yinying Y, and Mihaela C. (2007). Movement-assisted sensor redeployment scheme for network lifetime increase. Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems. Chania, Crete Island, Greece: ACM.

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

Youssef A, Younis M, Youssef M, and Agrawala A. (2006). WSN16-5: Distributed Formation of Overlapping Multi-hop Clusters in Wireless Sensor Networks. In Global Telecommunications Conference, 2006 GLOBECOM '06 IEEE; Nov. 27 2006-Dec. 1 2006. p 1-6.

Yu-Chee T, Chih-Shun H, and Ten-Yueng H. (2002). Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks. In INFOCOM 2002 Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings IEEE; 2002. p 200-209 vol.201.

Zhang H, and Arora A. (2003). GS3: scalable self-configuration and self-healing in wireless sensor networks. Computer Networks 43(4). p 459-480.

LIST OF PUBLICATIONS

1. Babar Nazir, Halabi Hasbullah, Sajjad A Madani, Sleep/wake scheduling scheme for minimizing end-to-end delay in multi-hop wireless sensor networks, EURASIP Journal on Wireless Communications and Networking 2011, 2011:92 *Publisher: SPRINGER*
2. Babar Nazir, Halabi Hasbullah, Energy Efficient Multi Hierarchy Clustering Protocol for Wireless Sensor Network, *IET Journal of Sensor System*, under review (second round). *Publisher: IET/IEEE*
3. Babar Nazir, Halabi Hasbullah, Energy Efficient and QoS based Routing Protocol for Clustered Wireless Sensor Network, *ELSEVIER Journal of Electrical and Computers Engineering*, Under review. *Publisher: ELSEVIER*
4. Babar Nazir, Halabi Hasbullah, Mobile Nodes based Clustering Protocol for Lifetime Optimization in Wireless Sensor Network*, Global Congress on Science and Engineering (GCSE 2011),* 28-30 Dec. 2011, Dubai, UAE. *Publisher: ELSEVIER.*
5. Babar Nazir, Halabi Hasbullah, Coverage and Energy aware Clustering Protocol (CECP) for Wireless Sensor Network, *Global Congress on Science and Engineering (GCSE 2011),* 28-30 Dec. 2011, Dubai, UAE. *Publisher: ELSEVIER*.
6. Babar Nazir, Halabi Hasbullah, Low Latency Sleep/wake Scheduling Scheme for Wireless Sensor Network, *Global Congress on Science and Engineering (GCSE 2011),* 28-30 Dec. 2011, Dubai, UAE. *Publisher: ELSEVIER.*
7. Babar Nazir, Halabi Hasbullah, QoS Based Energy Efficient Routing Protocol for Clustered Wireless Sensor Network*, IEEE Conference on Open Systems (ICOS2011),* 25-28 Sep. 2011, Langkawi, Malaysia. *Publisher: IEEE.*
8. Babar Nazir, Halabi Hasbullah, QoS based Routing Protocol for Minimizing End-to-end Delay for Wireless Sensor Network, *National Postgraduate Conference (NPC),*19-20 Sep. 2011, Tronoh, Malaysia. *Publisher: IEEE.*

9. Babar Nazir, Halabi Hasbullah, Dynamic Sleep Scheduling for Minimizing Delay in Wireless Sensor Network, *Saudi International Electronics, Communications and Photonics Conference,* April 23-26 2010, Riyadh, Saudi Arabia. *Publisher: IEEE.*
10. Babar Nazir, Halabi Hasbullah, Mobile Sink based Routing Protocol (MSRP) for Prolonging Network Lifetime in Clustered Wireless Sensor Network, *International Conference on Computer Applications & Industrial Electronics (ICCAIE 2010)*, 5-7 Dec. 2010, Kuala Lumpur, Malaysia. *Publisher: IEEE.*
11. Babar Nazir, Halabi Hasbullah, Energy Balanced Clustering in Wireless Sensor Network , *International Symposium on Information Technology 2010 (ITSim 2010)*, 15-17 June 2010, Kuala Lumpur, Malaysia. *Publisher: IEEE.*
12. Halabi Hasbullah, Babar Nazir, Region based Energy aware Clustering in Wireless Sensor Network, *International Symposium on Information Technology 2010 (ITSim 2010)*, 15-17 June 2010, Kuala Lumpur, Malaysia *Publisher: IEEE.*