HEAD NODE FAILURE DETECTION AND RECOVERY IN WIRELESS SENSOR NETWORKS

A Dissertation submitted in partial fulfillment of the requirement for the award of

Degree of

Master of Technology

In

Computer Science and Engineering

By

SURESH KUMAR

Roll No. – 2K11/CSE/17

Under the esteemed guidance of

Mr. Manoj Kumar

Associate Professor

Department of Computer Engineering

Department of Computer Engineering

Delhi Technological University

2011-2013

DELHI TECHNOLOGICAL UNIVERSITY, DELHI - 110042

CERTIFICATE

This is to certify that the work contained in this dissertation entitled "HEAD NODE FAILURE DETECTION AND RECOVERY IN WIRELESS SENSOR NETWORKS" submitted in the partial fulfilment, for the award for the degree of M.Tech. in Computer Science and Engineering at DELHI TECHNOLOGICAL UNIVERSITY by Suresh Kumar, Roll No. 2k11/CSE/17, is carried out by him under my supervision.This matter embodied in this project work has not been submitted earlier for the award of any degree or diploma in any university/institution to the best of our knowledge and belief.

Date:

Mr Manoj Kumar
RESEARCH GUIDE
Associate Professor
Dept. of Computer Engineering
Delhi Technological University, Delhi

ACKNOWLEDGMENT

My thanks and praise first and foremost goes to Almighty God for giving me the knowledge, opportunity, and the strength to accomplish this work.

I express my sincere regards and gratitude to **Mr. Manoj Kumar**, Associate Professor, Computer Science and Engineering Department, for his inspiration, expert guidance, moral boosting, continuous encouragement and appreciation, which are the vital factors in successful completion of my dissertation work. I humbly acknowledge deep gratitude towards my guide.

I sincerely extend my thanks to **Dr. Daya Gupta,** Head of Computer Science and Engineering Department for valuable guidance and support rendered for my thesis work.

Date:

Suresh Kumar
Roll no: 2k11/CSE/17
M.Tech. (Computer Science and Engineering)
Department of Computer Engineering
Delhi Technological University, Delhi-110042

ABSTRACT

Wireless sensor network is an emerging field and has applications in the area of disaster management (like earthquake, land sliding, tsumani etc), military services, battle field investigation, security surveillance etc. Wireless sensor network consists of hundreds and even thousands sensor nodes distributed autonomously to monitor physical or environmental phenomena. Each sensor node has limited computation, storage, sensing, wireless communication and battery power. Sensor nodes are deployed in unattended and harsh environment, hence recharging or replacement of batteries is impossible. Thus, battery constraint is an important challenge in WSN. For energy conservation and prolonging network lifetime various types of network topologies like clustering, tree-based etc are used.

Sensor nodes are prone to failure (like energy exhaustion and software or hardware failure) due to hostile and harsh environment. Node failure causes connectivity loss and sometimes partitioning of the network. In a clustered WSN, if a cluster head fails, all member nodes of that cluster get disconnected from rest of the network. It results in total data loss from the region under this cluster. It also causes holes in network topology. Therefore, for normal operation of a cluster and network connectivity, it is important to a recover the failure of a CH within appropriate time and that too with accuracy.

In this thesis, we have focused on detection of faulty CH in autonomous WSN and to recovery from it. Our aim is to develop a protocol which detects CH failures with accuracy and performs recovery. For accuracy we have used an agreement protocol in distributed manner, so that each cluster member can make a certain level of decision about failure. Each node participates to reach at final agreement about failure. Once CH failure is confirmed through agreement, an energy efficient new CH election protocol is proposed. This protocol, among other parameters takes residual energy as an essential component to finalize the selection of CH. Our protocol provides a robust and self healing approach that performs well for small to large scale WSNs

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

WSN	Wireless Sensor Network
BS	Base Station
QoS	Quality of Services
MAC	Medium Access Control
MANETs	Mobile Ad Hoc Networks
CH	Cluster Head
LEACH	Low-Energy Adaptive Clustering Hierarchy
TEEN	Threshold sensitive Energy Efficient sensor Network
APTEEN	Adaptive Threshold Sensitive Energy Efficient Sensor Network Protocol
TDMA	Time Division Multiple Accesses
CDMA	Code Division Multiple Accesses
ALU	Arithmetic and Logical Unit
PEGASIS	Power-Efficient Gathering in Sensor Information Systems
GPS	Global Positioning System
CM	Cluster Member
HNFRWN	Head Node Failure Detection and Recovery in Wireless Sensor Networks

Chapter 1

INTRODUCTION

1.1 Wireless Sensor Networks

Wireless sensor networks (WSNs) are made up of thousands of tiny devices distributed to monitor environmental conditions (like temperature, vibration, pressure etc); motion at different locations; industrial sensing, infrastructure protection, battlefield awareness etc at different regions [1]. These tiny devices are known as sensor nodes. Each sensor node consists of a radio transceiver, microcontroller, power supply, and the actual sensor (will be discuss in node architecture). Initially sensor networks were used for military purposed but now they are used for civilian application area including habitat monitoring, other applications and so on.

After the initial deployment, sensor nodes self-organize network infrastructure, through wireless communication between sensor nodes (as shown in Figure 1.1) [2]. The sensors then start collecting data about the environment. After collecting data, they process it and then send to base station. The Base station behaves like an interface between users and network. Users can retrieve information from WSNs by injecting queries and gathering results at base station. End users collect information from WSNs by connecting it to satellite via base station (as shown in Figure 1.1).

Normal features of sensor networks are self-organizing, mobility of nodes, dynamic network topology, limited power, node failures, large scale of deployment and short-range broadcast communication and multi-hop routing [3]. The strength of WSNs lies in their flexibility and scalability. The capability to self-organize and wireless communicates wirelessly makes them to be deployed in an ad-hoc fashion in remote or hazardous locations without the need of any existing infrastructure. Through multi-hop communication a sensor node can communicate to far away node in the network. This allows the addition of sensor nodes in the network to expand the monitored area and hence provides its scalability and flexibility property.

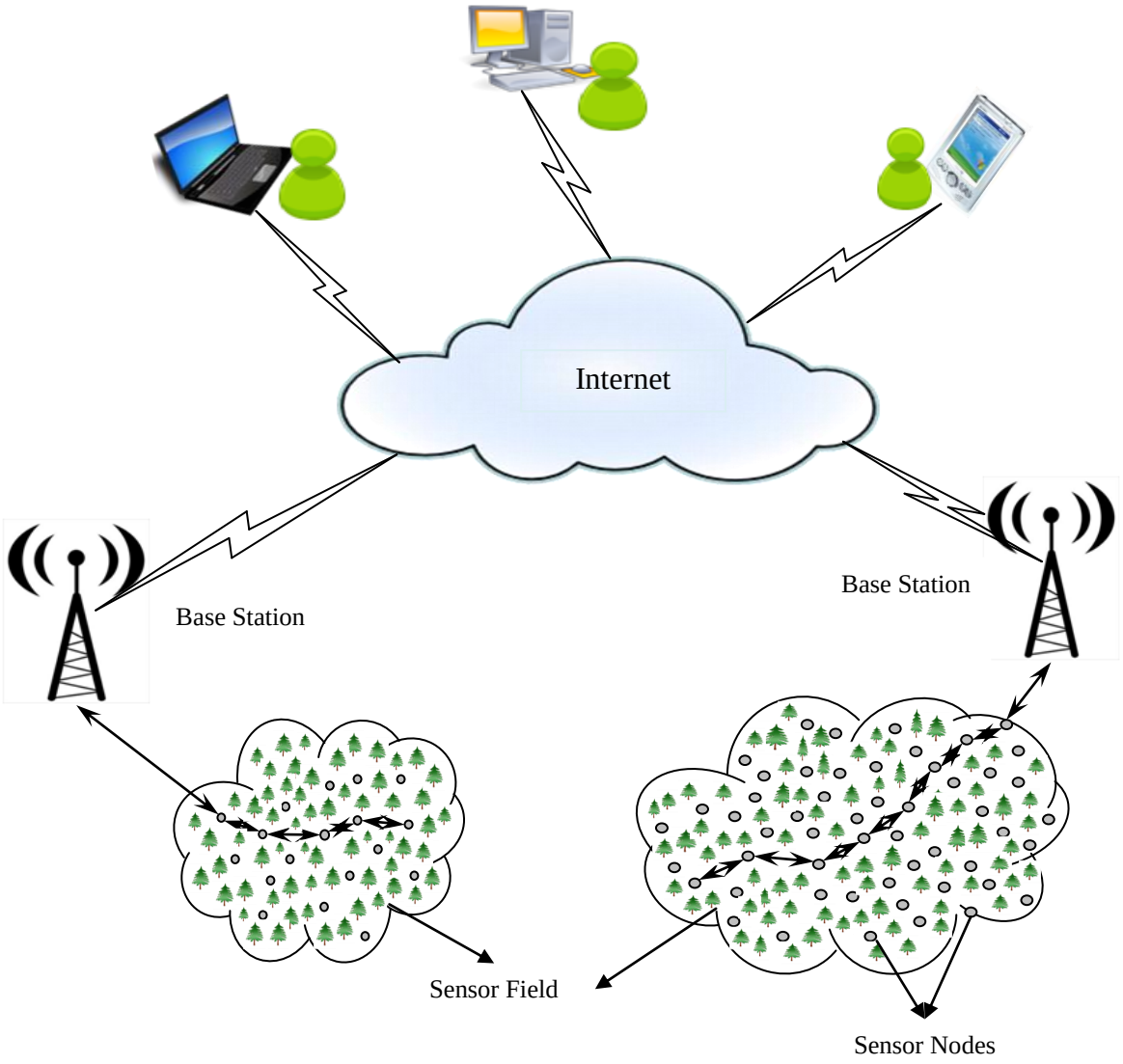

Figure 1.1 Wireless Sensor Network with Internet access

1.1.1 Types of Sensor Networks

WSNs are deployed on land, underground, and underwater. A sensor network faces different challenges and constraints according to the environment in which sensor network is deployed. There are five types of WSNs as discussed by Jennifer *et at.*[4]:

- Terrestrial WSN,
- Underground WSN,
- Underwater WSN,
- Multi-media WSN, and
- Mobile WSN.

Terrestrial WSNs [1] generally made up of thousands of inexpensive wireless sensor nodes deployed in a required area. In ad hoc deployment, sensor nodes can be dropped from airplane and placed into the target area. In a terrestrial WSN, reliable communication in an environment is very important. Sensor nodes should be able to communicate with the BS in terrestrial WSN, while battery power is a constraint. In any case, it is important for sensor nodes to conserve energy. Energy of sensor nodes can be conserved with data aggregation, removing data redundancy, reducing delay in terrestrial WSN.

Underground WSNs consist of a number of sensor nodes covered underground used to monitor underground conditions. Additional sink nodes are located above ground to transmit information from the sensor nodes to the base station. An underground WSN is more costly than a terrestrial WSN with regard to equipment, and maintenance. Underground sensor nodes are expensive because proper components must be used for reliable communication through water, and other contents. The underground environment makes wireless communication difficult due to loss of signals. An underground WSN requires meticulous planning during deployment to enhance lifetime.

Underwater WSNs comprise of a number of sensor nodes deployed underwater. The underwater sensor nodes are costly and less in density. Autonomous underwater vehicles are used for searching or gathering data from sensor nodes. Sensor nodes communicate via acoustic waves in WSN.

Another challenge is sensor node failure because of tough environmental conditions. Underwater sensor nodes must be able to self-configure and adapt to harsh sea/ocean environment. Like other WSNs, underwater sensor nodes are equipped with battery constraint which cannot be replaced or recharged. The underwater WSNs involve developing efficient networking techniques and underwater communication.

Multi-media WSNs are proposed to monitoring and track the form of multimedia such as video, audio, and imaging. Multi-media WSNs comprises of a number of sensor nodes with microphones and cameras. These sensor nodes communicates for, process, correlation, compression, and data retrieval over a wireless connection. Multi-media sensor nodes are deployed into the environment with planning for coverage guarantee. High bandwidth demand, high energy consumption, quality of service (QoS) condition, cross-layer design are area of difficulty in multi-media

WSNs. Multi-media content require high bandwidth to deliver contents. Therefore, energy consumption is high for high data rate. low energy consumption transmission techniques and High bandwidth are need to be developed. QoS is difficult to maintain in a multi-media WSNs, major reasons is delay which is variable and channel capacity which also varies. It is important to achieve a certain level of QoS for reliable content delivery. The filtering, and compression of contents can significantly improve network performance by removing redundant information and merging contents.

Mobile WSNs consist of a collection of sensor nodes that are mobile and can monitor the environment. Mobile nodes have all the ability of normal nodes like sensing, communicating and computing. The mobile nodes have the ability to change its position and self organize the network. In mobile WSNs nodes can move to gather information. The mobile nodes can communicate with each other, when they are in range and transfer sensed information. Another difference is data distribution. In mobile WSNs, data is distributed via dynamic routing while fixed routing or flooding is used in immobile WSNs. Sensor nodes placement, organization, control, coverage, energy maintenance, are key issues in mobile WSNs. Mobile WSNs applications are event monitoring, target monitoring, and monitoring of chemicals and biological hazards etc. For environmental monitoring, manual deployment might not be possible in disaster areas. Mobile sensor nodes can move to areas of events after deployment to provide the required coverage. In military surveillance and tracking, mobile sensor nodes can work together and make decisions based on the target. Mobile sensor nodes achieve greater coverage compared to immobile sensor nodes.

1.1.2 Wireless Sensor Node Architecture

For better understanding of sensor network it is important to know about all the components of sensor node. Common sensor node architecture is shown in Figure 1.2. The architecture of a generic wireless sensor node consists of four subsystems [5]:

- A computing subsystem consisting of ALU, a microprocessor and memory,
- A communication subsystem ,a short range radio for wireless communication,
- A sensing subsystem consists of a group of actuators, sensors.
- A power supply subsystem.

Each subsystem plays an important role in the sensor node.

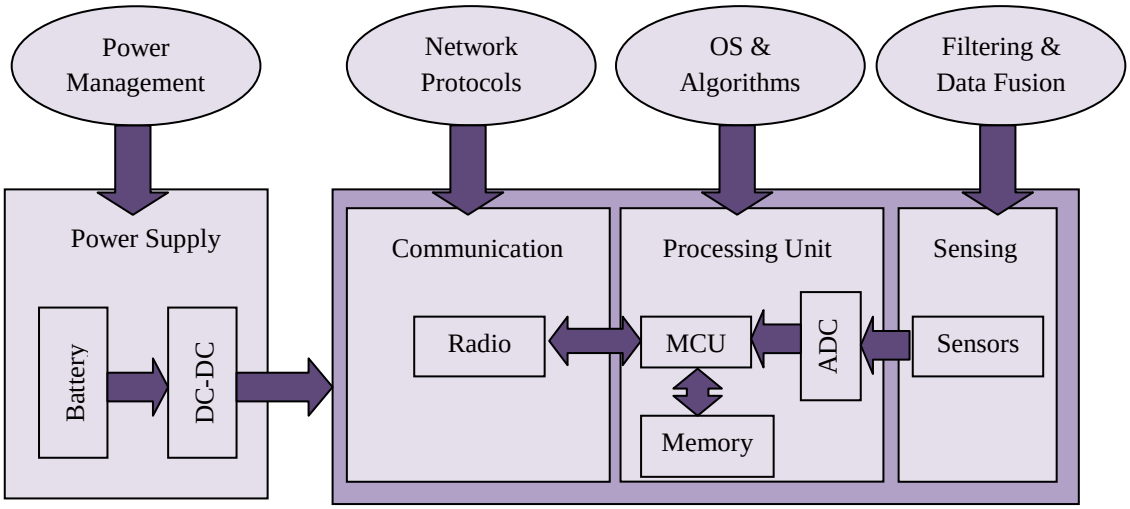

Figure 1.2 Sensor Node System

Radio: It enables wireless communication among sensor nodes and outside world. It consists of a short range radio which functions at unlicensed bands of near 2.4 GHz with a low data rate, and a single channel, and. It operates in four different modes: *Transmit*, *Receive*, *Idle* and *Standby* modes. In most radios, when radio operates in idle mode it consumes energy approximately equal to power consumed in Receive mode [6]. Thus, when it is not transmitting or receiving it is important to shut down the radio for energy conservation. Another factor, when radio changes its operating mode, a significant amount of power is dissipated in this transient activity.

Microprocessor: It provides intelligence to the sensor node. The microprocessor executes communication protocols, controls the sensors, and signal processing algorithms on the sensor data [7]. The microprocessor works in four modes: *active, idle, sleep, off.*

- In sleep mode, most internal peripherals are turned off, and can only be activated by an external event.
- In idle mode, the CPU is still inactive, but other peripherals are active, for example, the internal clock or timer.

- In the active mode, multiple sub modes may be defined based on clock speeds and voltages. In the active state, all peripherals are active.

Sensor: It translates physical phenomena to electrical signals. There are varieties of sensors that measure attributes such as temperature, light intensity, temperature, magnetic fields, sound, image, etc. Due to the diversity of sensors, there is no standard power consumption figure. For a simple sensor we assume that only the states on and off are given, and that the energy consumption within both states can be measured by time. However, more powerful sensors operate in different states, comparable to the microprocessor. To reduce energy consumption low power components can be used at the cost of performance which is not required.

Battery: The battery is an important component in sensor node. It supplies power to all component of sensor node. Therefore, sensor nodes lifetime totally depends on battery and network's lifetime depends on lifetime of sensor nodes. The amount of power drained from a battery should be checked. Since Sensor nodes are usually small, light and cheap and the size of the battery is limited. (Advancement in Battery technologies much more slower than semiconductor technologies. For example, the energy densities of Li-ion batteries only increased 50% from 1994 to 1999. While in the same period of time, the number of transistors of Intel processors doubles every 24 months.). Sensor nodes are deployed in unattended environment without any possibility of battery replacement consisting of thousands of nodes. Hence, energy consumption is vital factor to prolong sensor nodes lifetime.

1.1.3 Protocol Stack for wireless Sensor Network

The sensor network protocol stack is like the traditional protocol stack, have the following layers: Physical, Data Link, Network, Transport, and Application as discussed by Elizabeth [8] and shown in Figure 1.3. The WSN should also be aware of the following management planes for efficient functioning: Power, Mobility, and Task Management Planes.

- The Power Management Plane is for minimizing power consumption and to preserve energy.

- Mobility plane maintains a data route to the sink and manages the movement of sensor nodes .
- The task plane manages the sensing task assigned to sensor nodes so only those nodes which are necessary, are assigned sensing task and other node can focus their energy resource on routing and data aggregation.

Physical Layer: Physical Layer is responsible for frequency selection, carrier frequency generation, signal detection, modulation, and encryption. Its main priority is minimization of energy and secondary concerns are the same as those of other traditional wireless networks. The minimum output power required to transmit over a distance d is proportional to d to a power of n, where n varies from 2 to 4 and is closer to four when the antennae are near the ground as is typical in WSNs. This is due to ground-reflected rays, which causes partial signal cancellation. This problem is overcome by high node density and multi-hop communication .

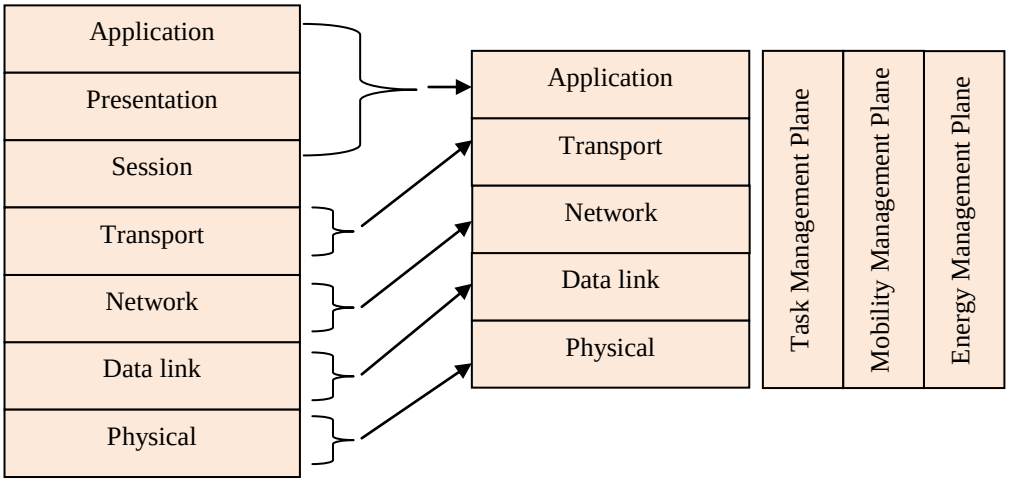

Figure 1.3 Protocol stack for WSNs

Data Link Layer: The Data Link is responsible for error control, medium access control ,the multiplexing of data streams, data frame detection,. A WSN must have a specific Medium Access Control (MAC) protocol to address the issues of data-centric routing and power conservation. The MAC protocol should meet two goals.

- The first is to create a network infrastructure, establishing communication links between nodes, and providing the self-organizing capabilities to the network.

- The second goal is to efficiently share communication resources between all the nodes.

Traditional MAC protocols do not meet above two goals as power conservation is a not a primary concern in their development and WSNs are controlled centrally have a much larger number of nodes than traditional ad-hoc networks. Any MAC layer protocol for WSNs must overcome the problem of changing topology of the sensor network due to node failure and redeployment.

Network Layer: One important function of the network layer is to provide internetworking with external networks such as other sensor networks, command and control systems and the Internet. The following considerations need to taken into account for designing network layer in a WSN:

- Power efficiency,
- WSNs are data-centric networks
- WSNs have attribute-based addressing and
- Sensor nodes are location aware.

The Link layer handles two nodes talk, the network layer is for deciding which node to talk to.

Transport Layer: The transport layer comes into play when the system needs to communicate with the outside world. Transmitting data from sink to outside user is a problem because WSNs do not use global identification and attribute based naming is used for sending the data. Very little research has been done at the transport layer.

Application Layer: At the application layer, a Sensor Management Protocol (SMP), SMP make the software and hardware of lower layers transparent to the network management applications. The programmers and system administrators interact with the sensor network using SMP. Again at application layer the infrastructure less nature of sensor networks and lack of global ids has to be considered. SMP provides the rules for the following :

- Data aggregation, attribute-based naming, and clustering
- Time synchronization

- Moving sensor nodes
- Exchange data related to the location finding algorithms
- Authentication, key distribution, and security
- Turning nodes on or off
- Querying WSN configuration status, reconfiguring the WSN

Different considerations must be taken when developing protocols for WSNs. Traditional thinking where main focus is on quality of service must be reversed. In WSNs quality of service must be traded with energy to preserve network lifetime. The focus must be on the entire network rather than for each individual node. Concern should be taken to preserve energy, allowing nodes to reconfigure , and modify tasks according to the resources available.

1.1.4 Wireless Sensor Network Challenges

WSN is an emerging area. It offers wide variety of applications and these applications can be implement in real world. To implement them more efficient protocols and algorithms are needed. Design a new protocol or algorithm addresses challenges of this field. To design a better protocol or algorithm, it is necessary to first clearly understood challenges [9]. These challenges are summarized below:

Physical Resource Constraints: The most important constraint in sensor network is the limited battery power of sensor nodes. Sensor nodes are left in unattended environment where recharge and replacement of battery is not possible. Sensor node's lifetime dependents on battery power. Thus effective lifetime of sensor network is directly dependent on battery. Hence the energy consumption is main design issue of a protocol. Limited computational power and memory size is another constraint due to that individual sensor node can store and process less amount of data. So the protocol should be simple and light-weighted. Limited bandwidth is also a constraint due to this communication delay can be high.

Ad-hoc Deployment: Sensor nodes are randomly deployed in required monitoring field without any infrastructure. For an example, for fire detection in a forest the nodes are typically dropped in to the forest from a plane. Sensor nodes itself create

connections with other nodes and form an infrastructure. Hence new protocol or algorithm should be able to handle this ad-hoc deployment.

Fault-Tolerance: Sensor nodes are prone to failure because of unattended environment. A sensor node may fail due to hardware or software problem or energy exhaustion. If few of sensor nodes fail, working protocol should handle all type of failures to maintain connectivity and prolong lifetime of network. For example, routing or aggregation protocol, must find suitable paths or aggregation point in case of these kinds of failures.

Scalability: In monitoring field, number of sensor nodes deployed could be in order of hundreds, thousands or even more. It depends upon the application. It may possible that initially deployed sensor nodes are not enough to monitor the environment. In this situation, protocol that is working upon network should be scalable and able to accommodate large number of sensor nodes.

Quality of Service: Some applications like multi-media or time critical needs QoS. Multi-media application requires enough good quality of contents (video, audio and image). In time critical application, the data should be delivered within a certain period of time from the moment it is sensed; otherwise the data will be useless. New protocols which are designed for such applications should handle QoS.

Security: In sensor networks, security is another important and challenging parameter. An effective and efficient compromise should be achieved, between security demands for secure communication and low bandwidth required for communication in sensor network. Whereas in traditional networks, the focus is on maximizing channel throughput with secure transmission.

1.1.5 Wireless Sensor Networks vs. Traditional Wireless Networks

There are many existing protocols, techniques and concepts from traditional wireless network, such as mobile ad-hoc network, cellular network, wireless local area network and Bluetooth, that are still in use in WSN, but there are lot of

fundamental differences which initiate the need of new techniques and protocols [10]. Some of the most important characteristic differences are summarized below:

- In WSNs, number of nodes is much higher than any traditional wireless network. Depending on the application, nodes may be in order of even millions. Thus, it requires an extremely scalable solution to make sure sensor network operations without any interruption.
- WSNs have large number of sensor nodes due to this addresses are not assigned to them. Instead of address-centric sensor networks are data-centric. Operations of sensor networks are concentrated on data instead of individual sensor node. Thus, sensor nodes need collaborative efforts.
- Most of traditional wireless networks use point-to-point communications, whereas sensor networks use broadcast communications.
- Sensor nodes are much cheaper than nodes in ad hoc networks.
- WSNs are event-driven or environment-driven. Sensor networks generate or collect data when any event occurs or environment changes, while human generates data in traditional networks. Hence, traffic pattern changes significantly from time to time. Mobile ad hoc Networks (MANETs) are designed for distributed computing, while sensor networks are mostly used to gather information.
- Data collected by neighboring sensor nodes is highly correlated. It has also been observed that the environmental quantities changes very slowly and some consecutive readings, sensed by sensor nodes are correlated. It is a unique characteristic of sensor network, which gives an opportunity to develop energy efficient protocols for routing and aggregation. These protocols reduce traffic and redundant data in network and prolongs network lifetime.

Thus, main focus of sensor network is to extend network lifetime, where traditional network try to maximizing throughput of a channel or minimizing node deployment in network.

1.1.6 Clustering in WSN

It is widely known that energy consumed in transfer of one bit can be used to execute number of instructions on processor of sensor nodes [11]. In densely deployed sensor network, neighboring sensor nodes produce or sense similar data due to overlapping sensing region. Due to energy constraint, it is necessary to remove data redundancy. So, all these reasons encourage to make some kind of grouping of sensor nodes so that sensor nodes can combine or compress the sensed data together in an intelligent way and transmit only compact data. It localizes most of traffic within each individual group and reduces the global data to transmit. As a result, it reduces the global traffic and contention in network. In densely deployed sensor network, process of grouping of sensor nodes is called as *clustering*. In a single cluster, to combine and compress data together in intelligent way is known as *data aggregation* [12]. There are some issues involved in process of clustering in a WSN.

- How many clusters should be formed in network that can optimize the performance of network.
- What should be the selection procedure of cluster-head in cluster.
- How many nodes should come under a cluster.
- How to introduce heterogeneity in network so that some powerful (in terms of energy) nodes can be put in the network which can work as cluster-heads and others as simple nodes working as cluster-member only.

Considering these issues, many protocols have been proposed in literature which deals with each individual issue.

1.1.7 Fault Tolerance in Clustering

In sensor network, battery is a constraint. It is challenge to extend network lifetime with limited battery. As it is well known that transmission cost in terms of energy is more higher than computing cost. Clustering is an effective way to prolong network life time. Clustering is used for data aggregation and localized traffic, which reduce global traffic.

One another challenge of this field is fault tolerance. Sensor nodes are prone to failure due to unattended and harsh environment. The failure of a sensor node affects

the normal operation of network. In case of CH failure, there is a loss of connectivity of all sensor nodes within that cluster and hence may disrupt the operation of the entire cluster. Therefore, it is important to recover the failure of CH within appropriate time for normal operation of cluster and maintenance of network connectivity. There are some issues involved in fault handling of cluster head (CH) in sensor network:

- In case of CH's failure how the cluster members of this cluster know about the CH failure.
- If few of cluster members observes CH failure and others observes CH as alive, then how they would know about actual situation.
- When cluster member detects that CH has been failed, who would start election procedure for new CH.
- Another issue is, how data would be handled during election.

1.2 Motivation

In WSN, there are so many challenges and issues as already have been discussed above. The main challenges are how to provide maximum lifetime to network and how to provide robustness to network. As sensor network totally rely on battery power, hence the main objective is maximizing lifetime of network.

The energy consumption can be reduced by removing data redundancy among the data that can be achieved by data aggregation. Effective data aggregation can be achieved by clustering in which a node is fixed for data aggregation which is called CH. All senor nodes within the cluster send data to CH which is further forwarded to sink. CH acts as a gateway node through which all cluster members are connected to remaining network or sink. As sensor nodes are prone to failure due to unattended and harsh environment, if CH fails, cluster member (that send data to this aggregator node) must be aware about the failure of the CH for the normal operation of the network. Fault detection and recovery of CH is important and challenging problem in WSNs. After failure detection, new CH should be elected in energy efficient manner.

1.3 Problem Statement

Clustering is an efficient way to energy conservation. In case of CH failure, cluster members of that cluster disconnects from remaining network. Therefore, all

cluster members have to know about failure and way to over come the failure for connectivity maintenance. Therefore, specific objectives of this study are:

- To devise an approach to detect CH failure.
- To devise an agreement scheme to agree on failure.
- To develop a scheme to elect a new CH.

1.4 Thesis Outline

The thesis is organized in the following way:

Chapter 1: This chapter starts with a introduction of WSNs, type of WSNs, protocol stack in WSNs,sensor node architecture ,difference between WSNs vs. traditional networks, challenges of sensor network, clustering in WSN and fault tolerance in clustering followed by the motivation of this work.

Chapter 2: It gives a detailed overview of clustering in WSN. This chapter also presents the literature survey on fault tolerance in WSN.

Chapter 3: It introduces and describes the new proposed protocol for fault tolerant clustering in a WSN. It includes system model and detailed code.

Chapter 4: It presents the performance analysis of the proposed protocol. It also provides comparative study.

Chapter 5: This chapter presents conclusion and scope of future work.

Chapter 2

REVIEW WORK

2 Introduction

Fault tolerance is very challenging problem in WSN and has attracted many researchers towards this field. As we have discussed in chapter 1 that clustering is very effective technique to reduce energy consumption in WSNs. In clustered network, CH plays the role of intermediate gateway through which data are forwarded to sink. As we know sensor nodes are deployed in harsh environment, they are prone to failures and hence we need a fault tolerance mechanism which overcomes the CH failure. In this chapter, we first discuss few CH election protocols used proposed in literature and later fault tolerant approaches will be discussed.

3 Clustering in Wireless Sensor Network

In sensor network, battery is a challenging constraint. Clustering is an effictive way of energy conservation in WSN. A group arrangement of sensor node in such an intelligent way that one sensor node acts as a leader (cluster head) and other nodes as followers (cluster members) is called *clustering* and group is called *cluster*. Cluster members sense data and send it to CH. CH combines and compresses data collected from cluster members and send to BS or sink (as shown in Figure 2.1). Such combining and compressing of data is called *data aggregation*[12].

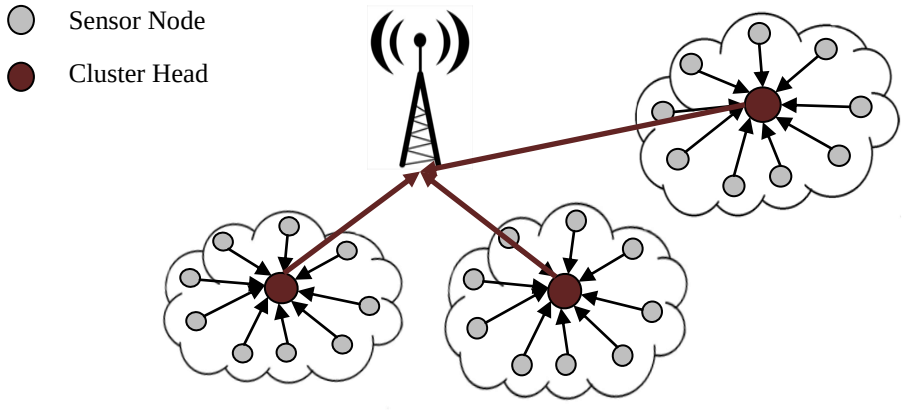

Figure 2.2 Cluster-based mechanism in WSNs

Benefits of clu

- Clustering localizes the message traffic within a cluster. All nodes of cluster communicate with rest of network via CH without direct communication. Thus clustering reduces global traffic.
- Clustering removes redundant data. Data collected by sensor nodes are sent to CH, where CH aggregates them and removes redundancy and then sends it to BS or forwards to other CHs.
- It increases lifetime of the network since only CH communicates with BS or sink while other nodes in cluster do not directly communicate with BS or sink. The role of CH rotates among cluster members, hence nodes die slowly attributed to better load distribution.
- It can also conserve communication bandwidth because it decreases the volume of data in the network by performing data aggregation at CHs.

There are numbers of clustering protocols proposed in literature [14-30]. Each protocol uses different method (like probabilistic, non probabilistic, residual energy etc.) for CH election. CH election algorithm can run on sensor nodes or some controller node or BS depending upon the approach. Here, we will discuss few of CH election protocols in detail. A comparison table 2.1 is also prepared after analyzing these protocols.

Heinzelman *et.al.* [14] purposed first clustering protocol Low-energy adaptive clustering hierarchy (LEACH) for WSN. LEACH is one of the popular hierarchical routing algorithms for WSNs. Clusters of sensor nodes are formed based on signal strength of sensor node which declares itself as CH. CH is used as a router to sink or BS (as shown in Figure 2.1). CHs create a TDMA schedule and send it to all cluster members to tell about their transmission slot to send data to CH. Cluster members send data to CH according to transmission schedule. The idea of clustering saves energy since the transmissions is done only by such CHs rather than all sensor nodes. Optimal number of CHs is approx. 5% of the total number of sensor nodes. All the data redundancy techniques such as data aggregation and others are local within the cluster. CHs alternate during lifetime for better balancing the energy dissipation of nodes. Next CH election is based on a number which is chosen by sensor nodes. This

number lies between 0 and 1.The node becomes a CH for the current round if the number is less than the following threshold:

$$T(n) = \begin{cases} \dfrac{P}{1 - P * (r \bmod \frac{1}{P})} & if\ n \in G \\ 0 & otherwise \end{cases}$$

where P is the desired percentage of CHs (e.g. 0.05), r is the current round, and G is the set of nodes that have not been CHs in the last $1/P$ rounds.

LEACH achieves reduction in energy dissipation compared to direct communication and minimum transmission energy routing protocol. In LEACH, nodes die randomly. Dynamic clustering increases lifetime of network. LEACH is a totally distributed protocol. It does not require global information about network. However, LEACH uses single-hop routing from sensor node to CH and sink or BS. Therefore, it is not applicable to large region's networks. Moreover, dynamic clustering brings extra overhead, e.g. head changes, advertisements etc., which may diminish the gain in energy consumption.

Lindsey *et al.* [15] proposed a chain-based data-aggregation protocol called Power-Efficient Data-Gathering Protocol for Sensor Information Systems (PEGASIS). The chain can be formed by the sink in centralized manner or nodes form the chain in greedy manner. The farthest node from the sink initiates chain formation and, at each step, the closest neighbor of a node is selected as its successor in the chain. In each data-gathering round, a node receives data from one of its neighbors which is farther to sink compared to it, fuses the data with its own, and transmits the fused data to its other neighbor along the chain. Eventually, the leader node which is similar to CH, receives the only two aggregated packets and transmits the aggregated data to the sink. The PEGASIS protocol has considerable energy savings compared to LEACH. The distances that most of the nodes transmit are much less compared to LEACH, in which each node transmits to its CH. The leader node receives at most two data packets from its two neighbors. In contrast, CH in LEACH performs data aggregation of several packets which is received. In contrast, a CH in LEACH has to perform data fusion of several data packets received from its cluster members. The main disadvantage of PEGASIS is the delay in-cured in transmitting

the data to BS because communication between leader node and other chain member is multi-hop unlike LEACH where data is sent directly by cluster member to CH.

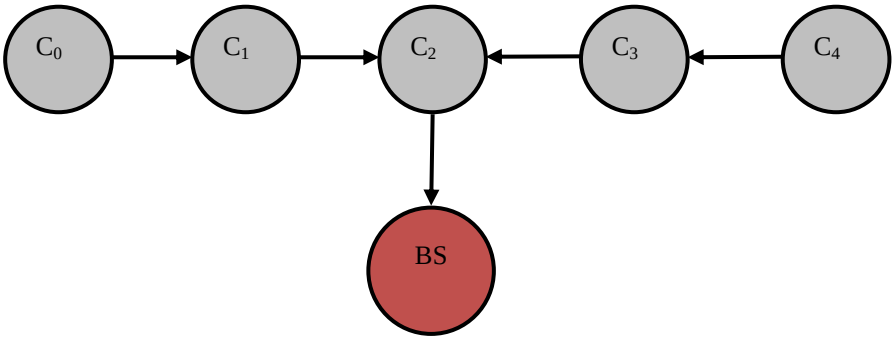

Figure 3.2 Chaining in PEGASIS

Hierarchical-PEGASIS [16] is an extension to PEGASIS proposed by *Savvides et al.*, which target at reducing the delay for packets during transmission to the Base Station. This is achieved by performing simultaneous transmission of data between neighboring nodes and then to the BS (as shown in Figure 2.3). However this leads to collision in the medium. To avoid collisions among the sensors, two approaches are used: a) Signal coding, and b) only separated nodes are expected to send data at the same time.

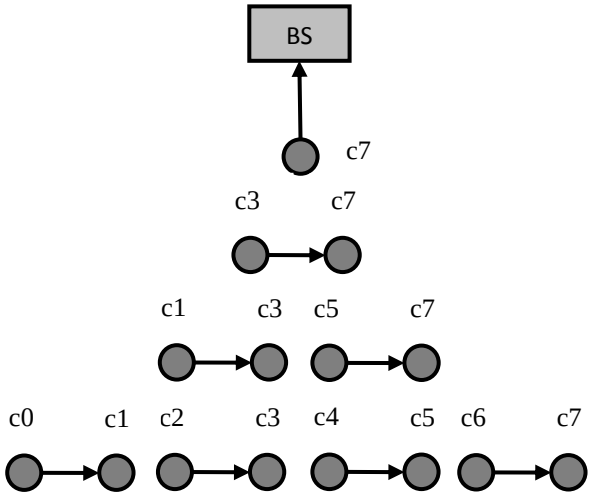

Figure 2.3 Chain formations in Hierarchical-PEGASIS

The chain-based protocol with CDMA capable nodes, constructs a chain of nodes, that forms a tree like hierarchy, and selected nodes at a particular level sends data to

the sensor nodes in the higher level . It allows data transmission in parallel and helps in reducing the delay.

Manjeshwar *et al.* Proposed Threshold sensitive Energy Efficient sensor Network protocol (TEEN) is a hierarchical protocol designed for time-critical applications and in which the network operated in a reactive mode in [17]. TEEN uses a hierarchical approach and use data-centric mechanism. The closer sensor nodes form clusters and process goes on the second level until BS is reached (as shown in Figure 2.4).

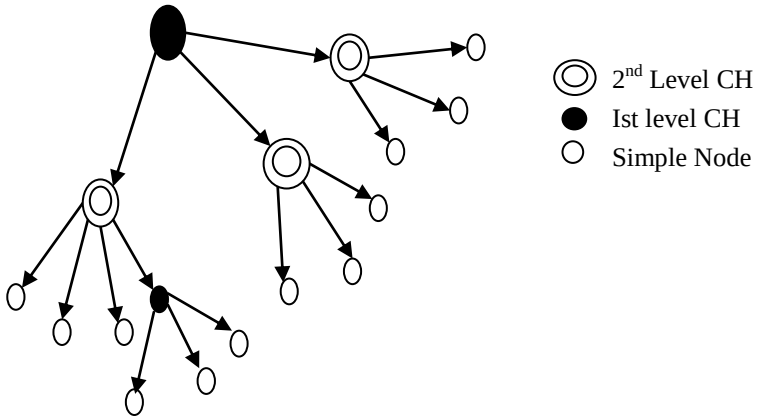

Figure 2.4 Hierarchical clustering in TEEN

After the clusters are formed, the CH broadcasts two thresholds, hard and soft thresholds, to all nodes in the cluster.

- A hard threshold is that value of an attribute greater than that a node is triggered to transmit data. Thus, the hard threshold allows the nodes to transmit when the sensed data is in the particular range.
- A soft threshold is a small change in the value of an attribute which trigger a node to transmit data again. Once a sensor node senses a value at or greater the hard threshold, it transmits data when the value of attribute changed by equal to or greater than the soft threshold.

Thus hard threshold and soft threshold reduce the number of transmissions and save the energy. Hard and soft threshold values can be adjusted to control the number of transmissions. When CHs change, new values for the above parameters are broadcasted.

The drawback of this scheme is that if the thresholds are not received, the nodes will not send the data and the user will not be able to get required data . To avoid

collision TDMA scheduling is used, but this will however introduce a delay in the reporting of the time-critical data. TEEN is not good for applications in which periodic reports are required. TEEN outperforms LEACH. The main disadvantage of the TEEN is the overhead with forming and maintaining clusters at two levels, as well as the complexity associated with threshold-based functions, and how to deal with complex queries.

The architecture of Adaptive Threshold Sensitive Energy Efficient Sensor Network Protocol (APTEEN) [18] is same as in TEEN and an extension to TEEN. APTEEN is capturing periodic data collections and reacting to time-critical events. When the BS forms the clusters, the CHs broadcast transmission schedule, the threshold values, and the attributes to all nodes.

- Attributes (A): a set of physical parameters about which the user is interested in obtaining information
- Thresholds: consists of the Hard Threshold (HT) and Soft Threshold (ST)
- Schedule: a TDMA schedule,a slot to each node
- Count Time (CT): the maximum time period between two successive reports sent by a sensor node

The node senses the environment in continuous manner and only those nodes that sense data equal to or greater than Hard Threshold transmits. Once a node senses a value greater than HT, it transmits data only when the value of that attributes changes by an amount equal to or greater than ST. If a node does not send data for a time period equal to CT, it is forced to retransmit the data. Each node in the cluster is assigned a transmission slot (TDMA schedule).

APTEEN outperform LEACH. APTEEN send data periodically to BS. The main disadvantage of the APTEEN is the overhead with forming and maintaining clusters at two levels, as well as the difficulties associated with implementing threshold-based functions, and how to deal with complex queries.

HEED [19] is a distributed clustering protocol that considers a hybrid of energy and communication cost to elect the CH. The HEED clustering operation is applied at each node for deciding if the node will become a CH or join as node in a cluster. A node with higher residual energy has a more chance to becoming a CH. The intra-cluster communication cost which is used to “break ties”, that is nodes that are

common to more than one CH. Cluster size and transmission power level of both intra–communication and inter-communication are considered as functions to determine the communication cost.

HEED outperforms LEACH in terms of prolonging network lifetime for a large network by distributing energy consumption. HEED can be applied to design sensor network that require energy efficiency, scalability, prolonged network lifetime, fault tolerance and load balancing.

LEACH-C [20] protocol is an enhancement of LEACH. It uses a centralized clustering algorithm to elect CH and same steady state phase as LEACH. During the set-up phase of LEACH-C, each node sends information about it to BS-current location (possibly determined using GPS) and residual energy level. To find out good clusters, the BS has to ensure that the load is evenly distributed among all the nodes. For this, The average node energy is computed by BS, and determines which nodes have energy below this calculated average. And nodes have energy below this average cannot be CHs for the current round.

Once the CHs and associated clusters are found, the BS broadcasts a message that obtains the CH ID . If a CH ID matches with its own ID, the node is a CH; otherwise the node determines its TDMA slot for data transmission and goes sleep till the required time to transmit data. LEACH uses distributed clustering algorithm and offers no guarantee about the placement and/or number of CHs. LEACH-C protocol can produce better performance by dispersing the CHs throughout the network.

4 Fault Tolerance in WSN

Wireless sensor network has many challenges as discussed in previous chapter. One of them is fault tolerance. Sensor nodes are prone to failure because of unattended, uncontrolled and harsh environment. A sensor node may fail due to hardware or software problem or energy exhaustion. If few of sensor nodes fail, it breaks connectivity of network. For example, if a routing or aggregator node failed the nodes that have path via routing nodes are disconnected. Hence, network should be robust to heal these failures. In this section we presents a overview of what is fault, why fault tolerance is needed and then classification of fault tolerance techniques. In WSN, node faults occur usually due to the following causes: environmental factors, battery power depletion, the failure of modules (such as communication and sensing

module) due to fabrication process problems, due to enemy attacks, due to nodes being out of the communication range of the entire network etc.

Definition of fault: *a fault occurs when a system deviates from its specification and cannot deliver its intended functionality and services. A sensor node fails due to energy depletion or hardware / software failure* [31].

Why fault tolerance: Fault tolerance is necessary to increase QoS of network. A few of reasons are as follows [32]:

- Low-cost sensor nodes are often deployed in uncontrollable, unfriendly and hostile environments. Therefore, failure of sensor nodes can occur easliy
- The applications of WSNs are being extended. WSNs are also deployed in some places where high security is required. Fault detection for sensor nodes in such specified applications is having great significance.
- It is not possible to manually examine all nodes in network whether the nodes are functioning normally or not.
- Nodes have battery constraint, so it is a common failure to occur due to battery exhaustion.

5 Fault Tolerance Approches

In WSN, number of tiny sensor nodes are deployed in harsh, uncontrolled and unattended environment. Sensor nodes fail due to failure of software or hardware or energy depletion. There are number of fault tolerant protocols proposed in literature [35-42]. Existing fault tolerant protocols can be classified in two types of approaches: centralized and distributed (as shown in Figure 2.5).

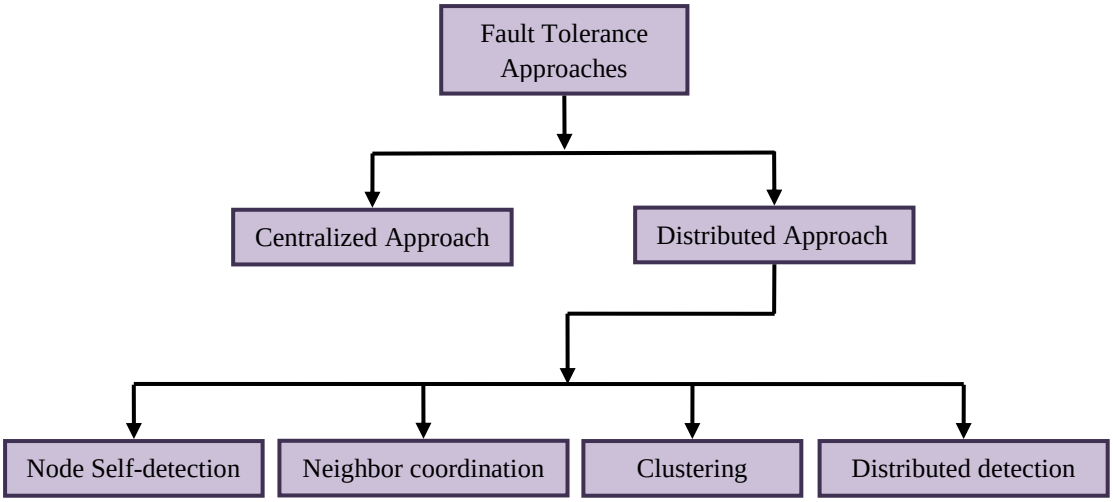

Figure 2.5 Classification of Fault Tolerant Approaches

6 Centralized Approach

Centralized approaches [33][34] are common solution to find out the cause of failures nodes in WSNs. In this approach, a logically or geographically centralized sensor node (BS, central controller or manager) is responsible for monitoring and tracing failed or suspicious nodes in the network. These kind of approaches mostly consider that the central node has unlimited resources in term of energy, so that it execute a fault tolerant algorithms. Normally the central node implements an active detection model to retrieve states of the individual sensor nodes and network performance by periodically injecting requests (or queries) into the network. It processes this information to find out the failed nodes. According to these approaches, if message transmission can be towards the central node then it extends network lifetime.

Advantage:

- It is an efficient and accurate way to identify faults of network.

Disadvantage:

- In this approach, periodically collecting all the sensor measurements and states information is not always affordable in resource-constrained sensor networks.
- As central node is responsible for all the fault detection and recovery, it becomes a single point of traffic concentration. Due to this, a high volume of message traffic and quick energy depletion happen near the central node.

- This approach becomes extremely expensive and inefficient in large-scale sensor network.
- In this approach, if multi-hop communication is used then it increases response delay from the BS for faults in the network.

7 Distributed Approach

In distribution approach [35-42], concept of local decision-making is used. It distributes fault detection and recovery uniformly into whole network. The goal of it is to allow a node to make certain levels of decision before communicating with central node. It believes that higher the decision making at sensor nodes, lesser is the information to be transmitted towards sink. In other words, the control centre should not be informed unless there is really a fault occurred in the network.

Advantage:

- It is an efficient and less expensive in terms of energy consumption for large-scale sensor network, as all faults are handled locally and in distributed manner unlike in the case of centralized approach.
- Unlike centralized approach, resource-constrained sensor networks can also afford it, as state information or measurements about sensor nodes aren't collected on a single point.
- A high message traffic problem of centralized approach can be easily solved here due to distributed fault detection and recovery.

Disadvantage:

- Energy consumption is more in distributed approaches.

Distributed approaches can further be classified as follows:

1) Node self- detection Approach
2) Neighbor Coordination Approach
3) Clustering Approach
4) Distributed Detection Approach

8 Node Self-detection Approach

In literature few researchers [35][36] have proposed node self detection method for fault tolerance. S Harte *et al*. [36] propose a node self detection method that monitors the malfunction of the physical components of a sensor node through both hardware and software interface to detect faults. In [35], node compares binary outputs of its sensors with the pre-defined fault models to detect faults.

9 Neighbour coordination Approach

Failure detection through neighbor coordination is another example of distributed fault tolerance [37][38]. Nodes coordinate with their neighbors to detect and identify the network faults (suspicious node or abnormal sensor readings) before consulting with the central node. In most cases, the central node is not aware of any failure unless something is believed to be wrong with high confidence via node coordination diagnosis. Such kind of design reduces network communication messages, and consequently conserves node energy. There are number of such designs exist according to need of individual application.

10 Clustering Approach

Clustering approach is also a way to detect faults in distributed manner. In clustering technique network is divided into cluster. In these clusters, CHs and cluster members detect faults locally in distributed manner. If a cluster handles a fault then it doesn't affect other clusters performance and operation. Even other ones don't know about it. In this approach cluster apply protocols in accordance with application of WSN. A few of them have been discussed here.

Nidhi *el at.* [39], propose FTEP which is a dynamic and distributed new CH election algorithm with fault handling capabilities based upon two-level clustering scheme. In FTEP, CH election starts if energy level of CH falls below a certain value for particular round or CH is unable to communicate due to hardware or software failure. According to residual energy levels of sensor nodes, election process appoints a CH and a back-up node to handle CH failure. CH sends state information to back-up node periodically and this back-up node maintains a timer. If timer of back-up node expires before no state information is received from the CH, it is assumed that CH has been failed. Back-up node then takes over the role of CH or CH will be elected. Only

those nodes become candidate nodes in election whose energy level is greater or equal to threshold value for that particular round. The value of energy level gets revised with each round of election. This value takes into consideration the depleting residual energy levels of candidate nodes. During the election procedure, current CH continues with its role of CH until a new CH is elected. This solves the problem of data transmission during election process. Moreover, election process is executed locally within the cluster in a distributed manner. Back-up node automatically takes over the role of CH once it detects failure of current CH.

In FTEP, Nidhi *el at.*[39] solves the problem of data loss during election of new CH. The CH election is based on residual energy of node and election process is executed locally within the cluster in a distributed manner. The back-up node handles the failure based on timer; it may be possible the state information has not been received due to any hindrance or environment condition or delay (received after expiring the timer). In these cases, back-up node gives false alarm. Back-up node is a single point to detect failure which may itself be disastrous.

Asim *el at.* [40], propose a cellular approach for fault detection and recovery. It provides network the capability to sustain in the event of failure due to energy-drained nodes. In this, network is partitioned into a virtual grid of cells (as shown in Figure 2.6), where each cell consists of a group of nodes. In each cell, a cell manager and a secondary manager are chosen to handle faults. Secondary manager works as a back up for cell manager, which will take control of the cell when cell manager fails to operate. These cells again form various groups. Each group chooses one of their cell managers to be a group manager. After the formation of virtual grid, failure detection and recovery is performed. The fault detection and recovery is performed locally with minimum energy consumption. Nodes failed due to energy drain are detected and recovered within a cell without affecting other cells or network. In this grid architecture, fault detection and recovery is implemented in distributed manner.

When a common node fails due to energy depletion, it informs to cell manager and no recovery steps use for common node. The cell manager and secondary cell manager are known to their cell members. If cell manager's energy drops below the threshold value, it sends a message to its cell member including secondary cell manager. This gives an indication for secondary cell manager to stand-up as a new

cell manager and the existing cell manager becomes common node and goes to a low computational mode. Common nodes will automatically start treating the secondary cell manager as their new cell manager and the new cell manager upon receiving updates from its cell members; chooses a new secondary cell manager. It is more energy efficient and faster. It detects only those failures that are due to low energy level. Other failures are not handled.

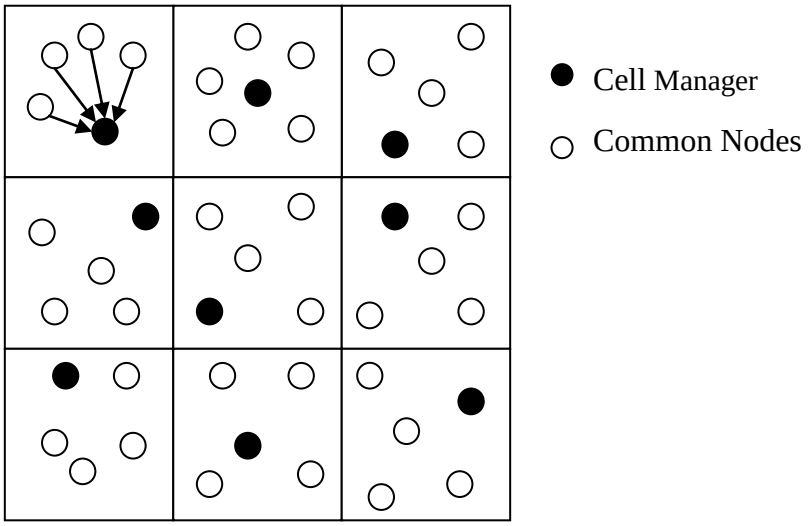

Figure 2.6 Virtual Grid of Nodes

In [41] Venkataraman el at., propose a localized cluster based method for fault detection due to energy exhaustion like cellular approach and network connectivity recovery which is energy efficient and responsive. It focuses on node notifying its neighboring nodes before it completely shut down due to energy exhaustion. In [41], Venkataraman *el at.* divide nodes of cluster into four types: boundary node, pre-boundary node, internal node and the CH and form a tree structure (as shown in Figure 2.7). According to node type, there are four failure mechanisms to handle failure. Boundary nodes do not require any recovery, but pre-boundary nodes, internal nodes and the CHs have to take appropriate actions to stay connected. Usually, when node's energy reaches below a threshold value, it sends a failure report message to its parent and children. It initiates the failure recovery procedure so that failing node parent and children remain connected to the cluster. A join request message is sent by the healthy child of the failing node to its neighbors. All the neighbors with in the

transmission range respond with a join reply message/join reject message messages. The healthy child of the failing node then selects a suitable parent by checking whether the neighbor is not one among the children of the failing node and whether the neighbor is also not a failing node.

In case of CH failure, children of the failing CH node exchange their energy status. The children, who are failing, are not considered for the new CH election. The healthy child with the maximum residual energy is elected as the new CH. After the new CH is elected, the other children of the failing CH are attached to this new CH and the new CH becomes the parent for these children. The failing CH also makes the new CH as its parent. Venkataraman el at. propose mechanism that performs all fault detection and recovery locally in cluster and provides network connectivity. In this mechanism nodes exchange messages to inform about failure. In case of CH failure, number of messages are exchanged to recover from it. Hence it consumes lot of energy. It detects only failures that occur due to low energy level. There is no provision of back-up node in case of sudden failure.

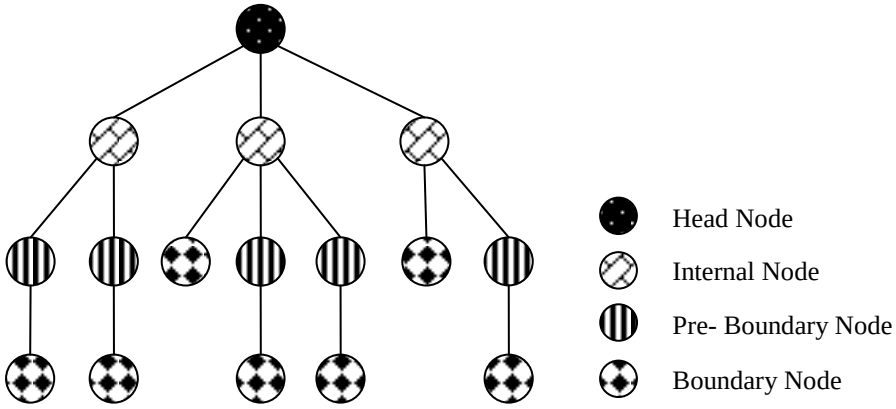

Figure2.7 Cluster Topology

11 Distributed Detection Approach

The basic idea of distributed detection is to let each node make certain level of decision on faults. This approach is especially energy-efficient and ideal for data centric sensor applications. However, there remain various research challenges in order to achieve a better balance between fault detection accuracy and the energy usage of the network. Usually, the efficiency of such failure detection schemes is

counted in terms of node communication costs, precision, detection accuracy and the number of faulty sensor nodes tolerable in the network.

Venkataraman *el at.*[42] propose a fault tolerance clustering method. It incorporates two types of nodes: gateway nodes and sensor nodes. Gateway nodes are less energy constrained nodes (CHs) and sensor nodes are energy constrained. The gateway nodes maintain the state of sensor nodes as well as multi-hop route for collecting sensor nodes.

Gateways periodically send status updates through inter-gateway communication (as shown in Figure 2.8). Status updates inform all the gateways about the location of the rest of the clusters in network. Status messages also act as heartbeat messages from the gateways informing about their presence. If link fails between two nodes, status updates can be missed. Hence a consensus has to be reached from all gateways before recovery commences. A gateway should not be considered completely failed until even one of the gateways in the network is able to communicate with it. Once the gateways reach a consensus about the occurrence of a fault, the next step is to identify the type of faults and allocate the sensor nodes to new clusters. The status message is parsed to extract the identity of sensor nodes that cannot communicate with the gateway due to range faults in the gateways. When a gateway is identified as completely failed all the sensor nodes in its cluster are recovered. It improves the stability of the network and reduces the overhead of re-clustering and network reconfigurations. Due to consensus it detects faults with high accuracy. When a gateway node dies, the cluster is dissolved and all of its nodes are reallocated to other healthy gateways. It consumes more time as all the cluster members are involved in the recovery process of a gateway nodes. Since the gateway nodes are less energy constraint and more static than the rest of the network nodes, sensor nodes close to the gateway node die quickly and create holes near to gateway nodes and decrease network connectivity.

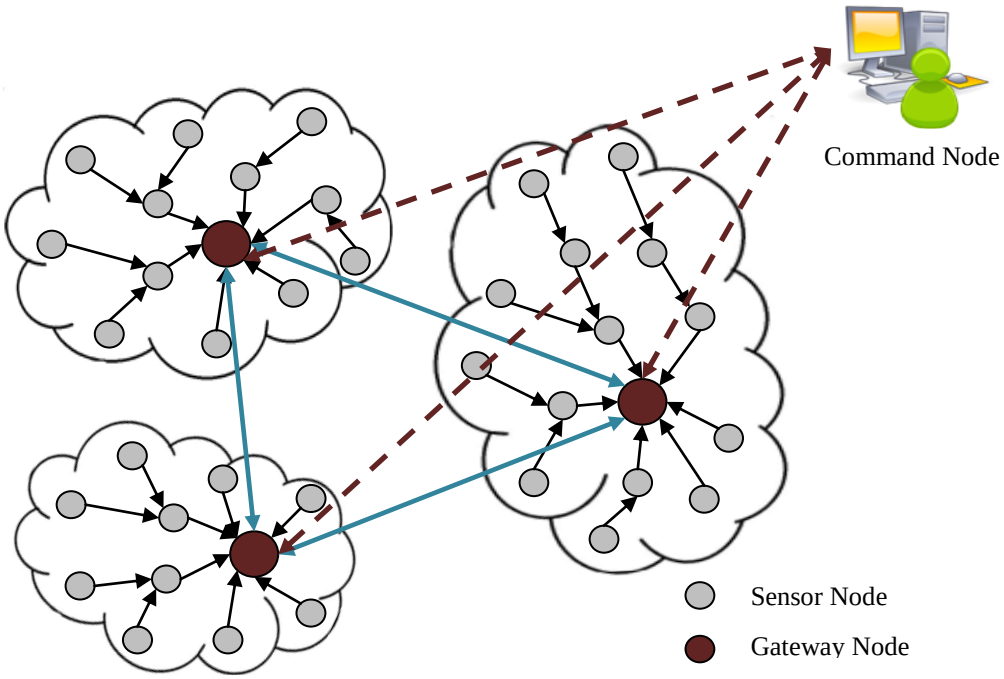

Figure 2.8 Multi-gateway Clustered Sensor Network

12 Chapter Summary

This chapter presents an overview of CH election and fault tolerance techniques.

- In centralized approach a central sensor node like BS or central controller/ manager or sink is used to handle faults in network.
- In distributed approach, there is no central node and each node makes a certain level of decision about fault.
- A few protocols (related to clustering and distributed detection approach) are elaborated in deep to explain distributed approach.

Chapter 3

Head Node Failure Detection and Recovery in Wireless Sensor Networks

13 Introduction

In previous chapter we have discussed about the WSN and various failure detection and recovery protocols for CH. Now we present our proposed protocol *"Head Node Failure Detection and Recovery for WSNs"* to handle CH failure problem. In protocol description, assumptions about network and energy model that has been considered for proposed protocol are discussed in detail. After this a detailed description of the algorithm with *pseudo code* and *flow chart* is presented. Like other protocols there are few issues involved in our protocol. These issues will be presented in brief and finally we summarize this chapter.

14 HNFRWN Protocol

In previous chapter, election and fault tolerant protocols for CH are discussed. Fault tolerant protocols are broadly categories in two categories: centralized approach and distributed approach. Focusing on the distributed approach, we found that existing protocol try to detect and recover CH failure locally within a single cluster in distributed manner. In this approach, CH failure detection and recovery process of a cluster does not affect working of remaining network. They distribute the load of fault tolerance; hence bottleneck on few nodes does not occur. They localize the network traffic, hence it reduce the global traffic. Different protocols use different methods to detect and recover a CH failure, some of them use concept of backup nodes; few of them use agreement protocol etc. Considering above advantages, we have taken this approach in our protocol design.

To increase accuracy of detection and avoid unnecessary energy consumption caused due to mistaken detection, we have used an agreement protocol to design our approach HNFRWN. In HNFRWN. each cluster member individually and independently makes decision about failure of CH. For confirmation about failure, cluster member runs a distributed agreement protocol among multiple cluster members to reach an agreement on the failure of CH. Recovery process starts only

after confirmation about CH failure from all cluster members. It conserves energy and increases accuracy because of less chances of mistake.

HNFRWN is energy aware as it takes residual energy of sensor nodes in election of CH. The protocol also balances the network load by rotating the role of CH among cluster members in a cluster.

15 System Model

3.3.1 Assumptions

We make few assumptions as follows:

- Nodes fail due to energy depletion or any hardware or software problem.
- All nodes are homogenous, immobile and have limited energy and initially have same amount of energy.
- Each node has two transmission power levels.
- Transmission power is uniform across network.
- Every node knows its current energy level [39].
- A message sent by a node is received correctly with in a finite time by all nodes in the cluster.
- The clusters regions are static formed at the start of the network. After that CH rotates.
- Nodes know about their location [39].
- The BS is fixed and is located outside the network.
- All nodes are time synchronized.

3.3.2 Network Model

Figure 3.1 shows the network model used. Various symbols and terms used are shown in Table 3.1. All sensor nodes are homogeneous, which have two transmission modes i.e. high power transmission mode for communication between CHs and BS and low power transmission mode for communication between cluster members and CH. The distribution of sensor nodes is uniform throughout the environment. Communication medium is radio links. Links between two sensor nodes is considered bidirectional. There is only single channel for communication between sensor nodes.

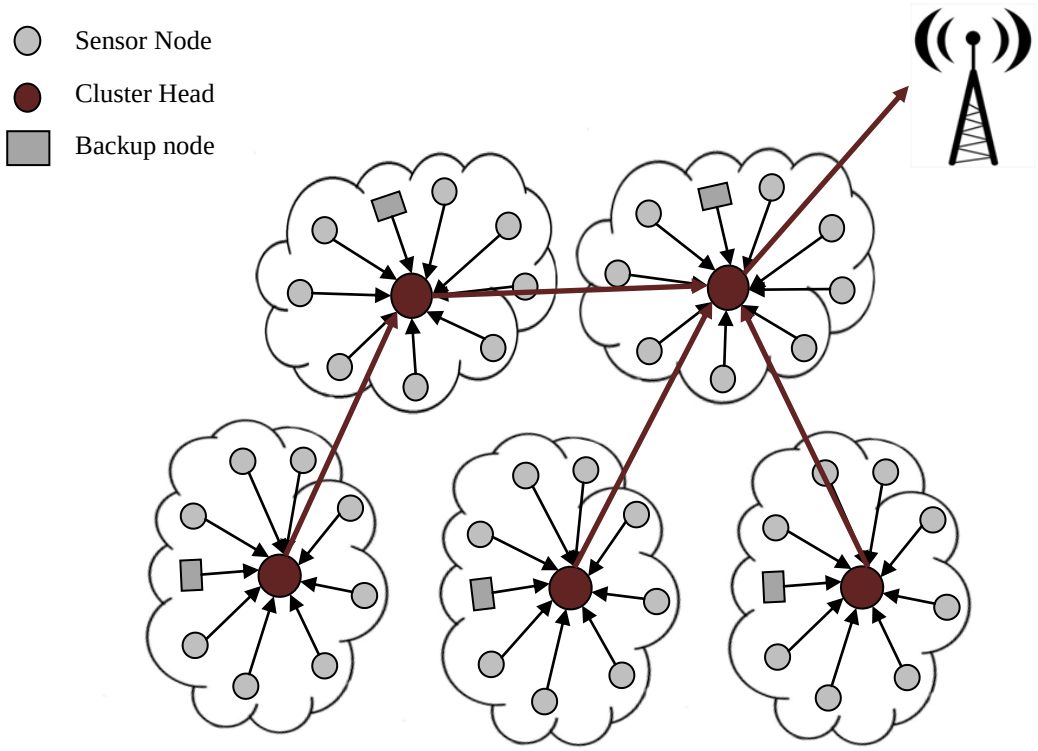

Figure 3.4 Network Model

During the network deployment, all the sensor nodes are assigned same initial energy value. All sensor nodes are assumed to know their geographical location [43]. We assume that clusters may overlap during election procedure so that every sensor node comes under at least one cluster. Initially, some sensor nodes are randomly selected as CHs and they announce their energy levels and location information. These CHs start working in high power transmission mode while other regular sensor nodes work in low power transmission mode.

Table 3.1

Notations used to explain Algorithm

d	Distance that message travels
b	Number of bits in the message
e_{tx}	Energy dissipated in transmitter electronics per bit (taken to be 50nJ/bit)
e_{amp}	Energy dissipated in transmitter amplifier (taken to be 50nJ/bit)
e_{rx}	Energy dissipated in receiver electronics per bit (taken to be 50nJ/bit)
E_{tx}	Energy consumed in transmission
E_{rx}	Energy consumed in receiving
SV	Status Vector
Loc_j	Location of node j
CH_i	Cluster head of cluster i
c_i	Cluster i
$e_{n_i}^{crnt}$	Current energy of node i
e^{max}	Energy level at which sensor node can participant in election of CH
$e^{min'}$	Energy level at which current CH starts election process
$e^{min''}$	Energy level up to which election process must be completed
TR	Transmission range of node
CS	Candidate Set

3.3.3 Sensor Node's Energy Model

A sensor node consists of sensors, analog signal conditioning, data conversion circuitry, digital signal processing and a radio link [14]. Each component of sensor node consumes energy for sending and receiving data. The following energy consumption model shows the energy, consumed by components of sensor node as shown in Figure 3. 2).

Assuming $1/d^2$ path loss, the energy consumption on each sensor node is:

$$E_{tx} = \left(e_{tx} + e_{amp} \times d^2\right) \times b \qquad (1)$$

$$E_{rx} = e_{rx} \times b \qquad (2)$$

According to eq. 1, the transmitter unit consumes energy E_{tx} to send b bits; where e_{tx} is the energy consumed by transmitter electronics per bit and e_{amp} is the energy used by amplifier per bit. According to eq. 2, the receiving unit consumes E_{rx} energy to receive b bits, where e_{rx} is the energy used by receiver electronics per bit.

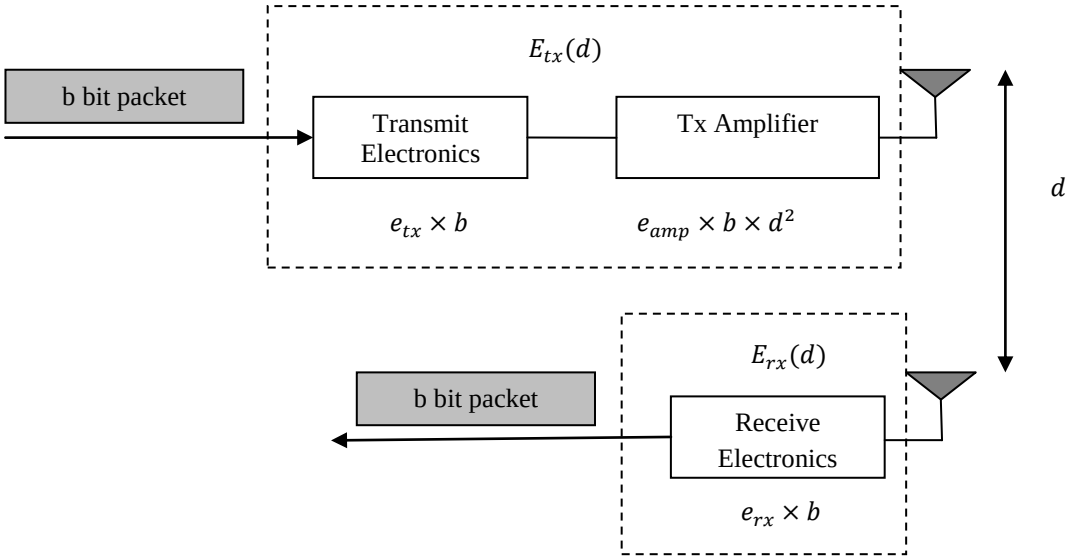

Figure 3.5 Radio Model

Table 3.1 summarizes the meaning of each term and its typical value. The values for e^{max}, $e^{min'}$ and $e^{min''}$ are updated during each election process. Typically, value of e^{max} for next election round is set to the average value of the energy levels of all candidate nodes during current election round. The values of $e^{min'}$ is set according to e^{max}. The values of $e^{min''}$ is set according to $e^{min'}$ as follows:

$e^{min''} = e^{min'}$ - (energy consumption during election process + energy consumption in data transmission during that period)

16 Description of HNFRWN Protocol

HNFRWN works in two phases namely: setup phase and steady state phase (as shown in Figure 3.3). Setup phase runs only once, when network starts working. In setup phase, clusters are formed and remain fixed through-out the lifetime of network. Steady state phase consists of three phases: CH election, failure detection and failure recovery. Failure detection runs parallel with network operation.

3.4.1 Setup Phase

Clusters are formed only once during the setup phase before the network starts to run (as shown in Figure 3.4). Initially, some sensor nodes are randomly selected as a CH, because energy of each sensor node is equal in amount. CHs send advertisement messages that contain energy and location information of CHs to neighboring sensor nodes. Each sensor node that listen to this advertisement message responds with a return message comprising its residual energy and location. However, a sensor node may be in the range of multiple CHs, but finally it must be associated with a single CH. If any sensor node falls within the overlapping region of more than one CHs, it decides its association to a CH by calculating the value of e/d (energy/distance). CH that has maximum e/d value is selected as final CH by that sensor node. If more than one CHs yields same maximum e/d value, then any of them is randomly selected. If a sensor node does not fall within the range of any CH, it declares itself as a CH and gets activated in high power transmission mode. When clusters are established, the CHs collect the data from cluster members, perform local data aggregation and send it to BS or sink node in multi-hop manner.

Clusters form circles of radius $TR/2$. $TR/2$ size is taken to confirm that every node in cluster able to communicate with other nodes within a single-hop in same cluster.

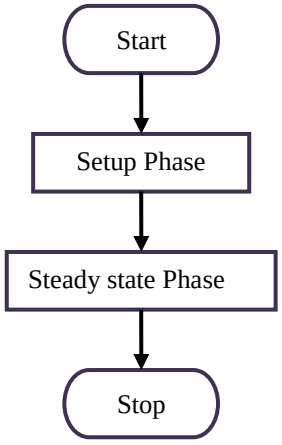

Figure 3.6 Flow chart of Protocol phases

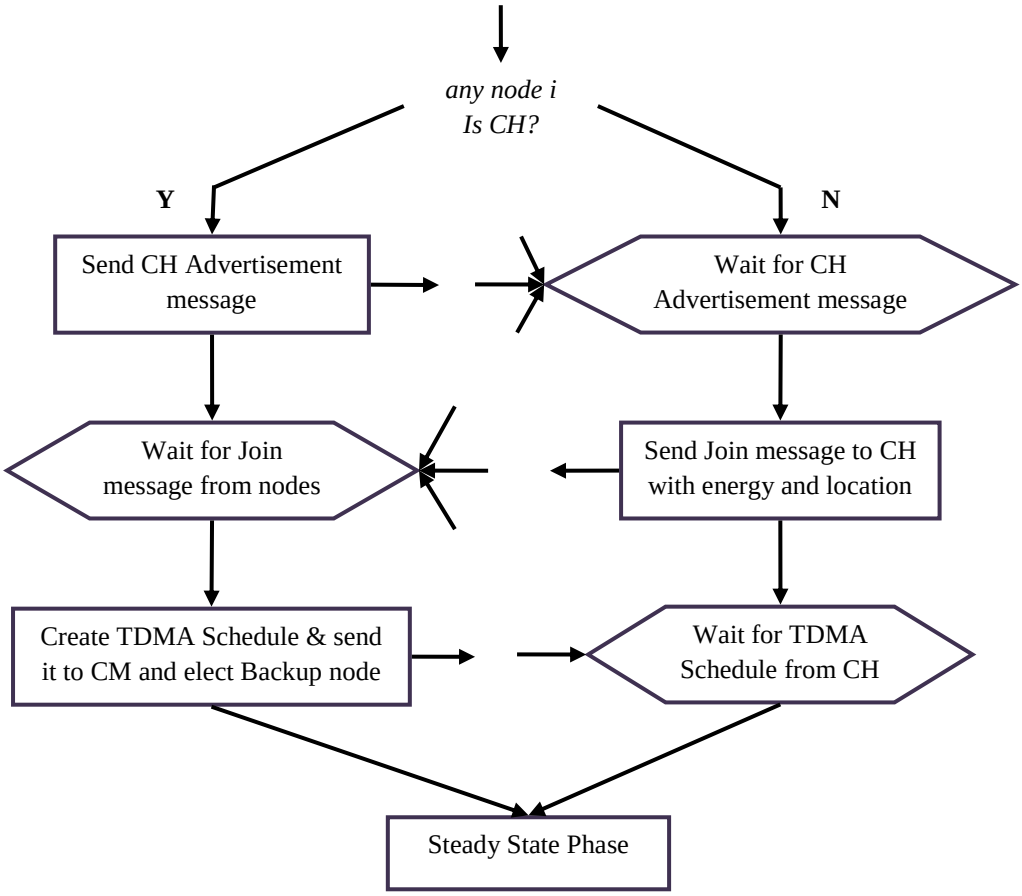

Figure 3.7 Setup Phase

3.4.2 Steady State Phase

Once cluster is formed, CH creates a TDMA schedule for cluster members and sends it to them. Sensor nodes sense data and send it to CH according to TDMA schedule. This process continues for all clusters until CH's current energy level ($e_{CH_i}^{crnt}$) equals to or less than $e^{min'}$ or CH fails. Then CH starts election process of new CH for next round or recovers from failure respectively (as shown in Figure 3.5 and Figure 3.6).

```
I. Steady State Phase
// Set of cluster in network
C = ( c1, c2 , c3.....ck )
while ( network is alive && observation period has not
expired)
{
  for any cluster head (CHi) of ci
  {
    if ( e_CHi^crnt > e^min' ) then
      network perform normal operation
    else       //when ( e_CHi^crnt ≤ e^min' ) then
      Call election algorithm (ci , ei^max)
      where i = 1, 2, 3, ..... k
      //when cluster head fails
    if (failure detection algorithm( )) then
    {
      New CH ← back-up CH
      Call new back-up node election (ci, ei^max)
    }
  }
}
```

Figure 3.8 Pseudo Code for Steady State Phase

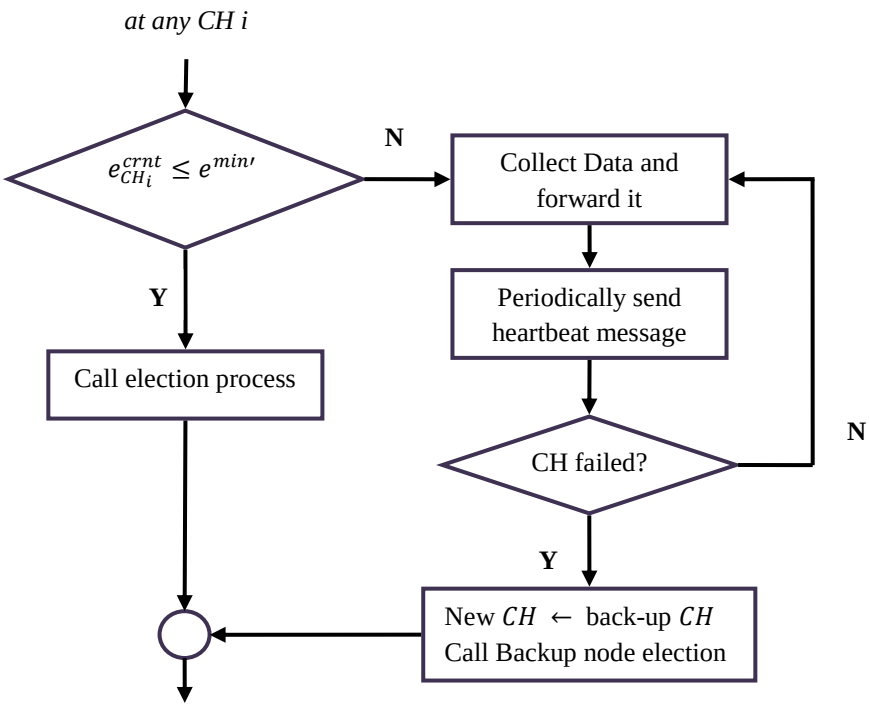

Figure 3.9 Flow Chart of Steady State Phase

CH Election: CH broadcasts e^{max} for next round in low power transmission mode, which is average energy of those cluster members who participated in last election process. All sensor nodes within the cluster listen to message and compare e^{max} with their current energy level ($e_{n_i}^{crnt}$). Sensor node which have $e_{n_i}^{crnt}$ greater than or equal to e^{max} , marks itself as a participant for election process(as shown in Figure 3.7 and Figure 3.8). All participant sensor nodes broadcast their $e_{n_i}^{crnt}$ and location (Loc_i) in low transmission mode. All participant sensor nodes can listen to each other because all sensor nodes are within low power transmission range of each other. Because of this, all participant sensor nodes know about $e_{n_i}^{crnt}$ and Loc_i of each other. Hence, each participant sensor node is aware about higher energy participant sensor node. The participant sensor node with highest value of $e_{n_i}^{crnt}$ promotes itself as CH and gets activated in high power mode; whereas sensor node with second highest energy upgrades itself as backup CH. New CH receives $e_{n_i}^{crnt}$ and Loc_i of all participant sensor nodes during election process, it calculates average of all $e_{n_i}^{crnt}$ and gets value of e^{max}, which is used for next round. Both new CH and backup node know the value of e^{max}. All participant sensor nodes mark themselves as non-participant sensor nodes again. The previous CH also starts working in low power mode.

II. Election Algorithm (c_i , e_i^{max})

{

Candidate_set(CS) = {$\emptyset$}

// set of node in cluster c_i

N_i = {n_1, n_2 , n_3……n_j} where j is the number of nodes in a cluster

CH_i broadcast e_i^{max} to each cluster member (CM) in N_i

for each node n_d in N_i where $d = 1, 2, 3\ldots\ldots j$

Mark $n_d \leftarrow np$ // mark all nodes as non-participant

for any node n_d , if ($e_{n_d}^{crnt} \geq e_i^{max}$)

{

mark $n_d \leftarrow p$ //mark it as participant node

$CS = CS \cup n_d$

//add n_d to candidate_set (CS = {n_1, n_2 , n_3……n_q})

where q is the number of nodes in candidate set

}

For each node n_f in CS where $f = 1, 2, 3\ldots\ldots q$

{

//broadcast its current energy level and location

broadcast_msg ($e_{n_f}^{crnt}$, Loc_f)

// receive messages broadcasted by other candidate

Receive_msg ($e_{n_f}^{crnt}$, Loc_f)

}

New $CH \leftarrow$ highest energy node

New back-up node $\leftarrow$ second highest energy node

//calculate e_i^{max} for next election round

Set $e_i^{max} = \left(\sum_{f=1}^{f=j} e_{n_f}^{crnt}\right)/j$

//both new CH and back-up node knows the value of e_i^{max}

mark all nodes(including previous cluster head) in candidate_set(CS) non- participant

}

Figure 3.7 Pseudo Code for CH Election

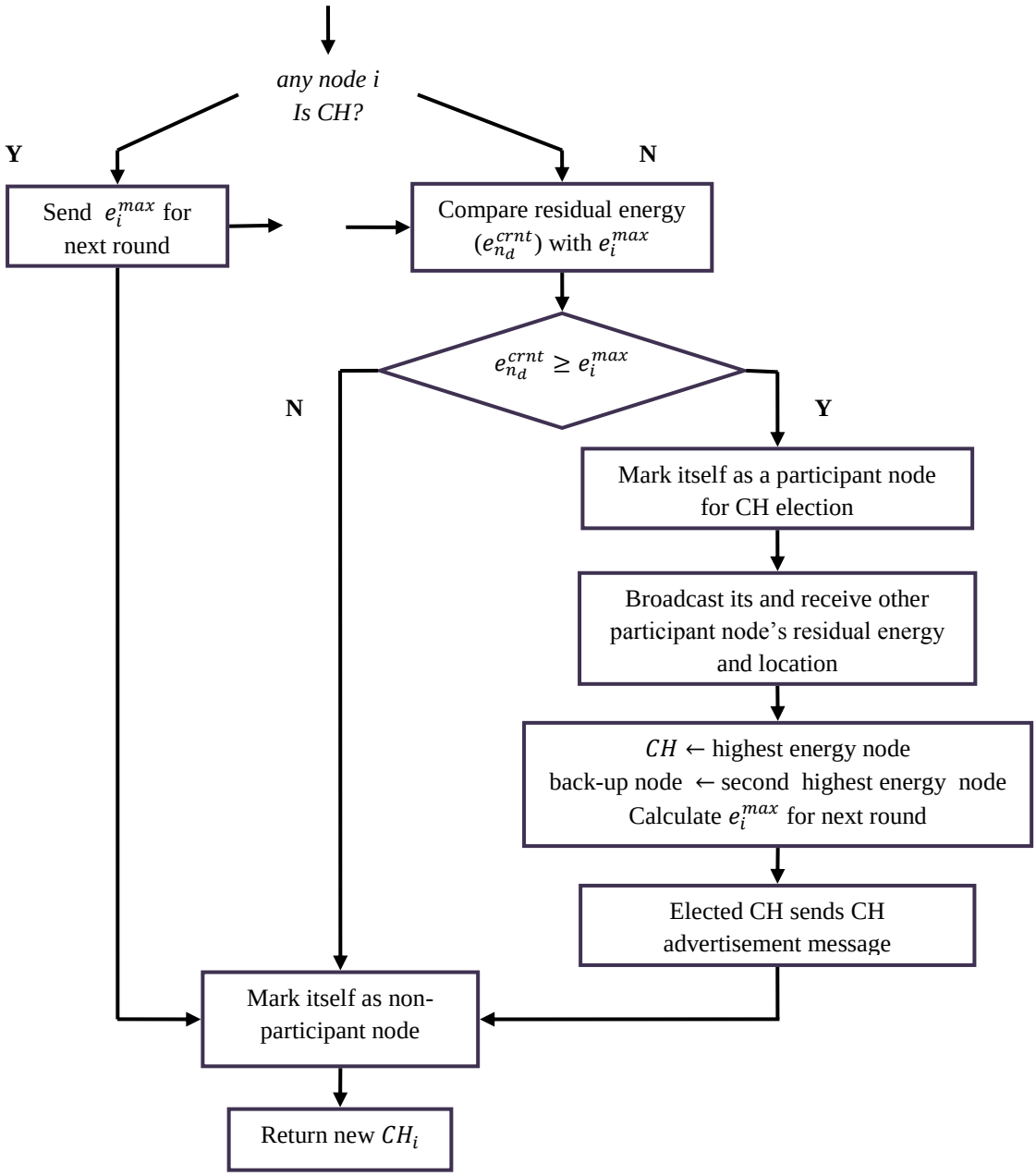

Figure 3.8 Flow Chart for CH Election

Failure Detection: The detection process runs parallel with normal network operation by periodically performing a distributed detection process at each cluster member (as shown in Figure 3.9 and Figure 3.10). For failure detection mechanism each cluster member maintains a status vector and a timer. In status vector each bit corresponds to a cluster member. Initially all bits are set to zero of status vector on each sensor node. A bit in the vector is set once its corresponding cluster member detects that CH has failed. CH of each cluster periodically sends a hello message (i.e. notification that CH is alive) to cluster members after a certain time interval. Cluster members also know

about time interval, CH sends it to cluster members. After that time interval cluster member, who does not listen hello message, sets its corresponding bit as one in status vector and locally decides that CH has failed and broadcasts data plus status vector. Other cluster members also listen this message. They extract status vector from message and merge it with own status vector and this process continuous up to the end of the TDMA schedule. At the end of the TDMA frame, cluster members reach on an agreement about failure of CH. If all bits of status vector are set then it is decided that CH has failed.

III. Cluster Head Failure Detection Algorithm (c_i)

```
{
  //Set of nodes in cluster c_i
  N_i = {n_1, n_2 , n_3.....n_j} where j is the number of nodes in a
  cluster
  Every node  n_d in N_i maintains a SV of size j − 2
  Every node resets the SV
  // after every t time interval,  CH_i  sends heartbeat
  message
  if (t = 100 × n )then where n = 1, 2, 3, ….
  {
    CH_i broadcasts heartbeat message
    for every node n_d in N_i
    {
       if (n_d doesn't listen heartbeat message) then
                SV[d] = 1
      Broadcast_message (DATA + SV)  at their TDMA Schedule
      Receive_message (DATA + SV)  by other nodes and merge
      with its SV
     }
        for d = 1 to j − 2
       if ( SV[d] = 1 ) then
          Return true              // true for failure
   }
  else
  {
    for every node n_d in N_i
    sends DATA  at their TDMA schedule
  }
}
```

Figure 3.9 Pseudo Code for Failure Detection

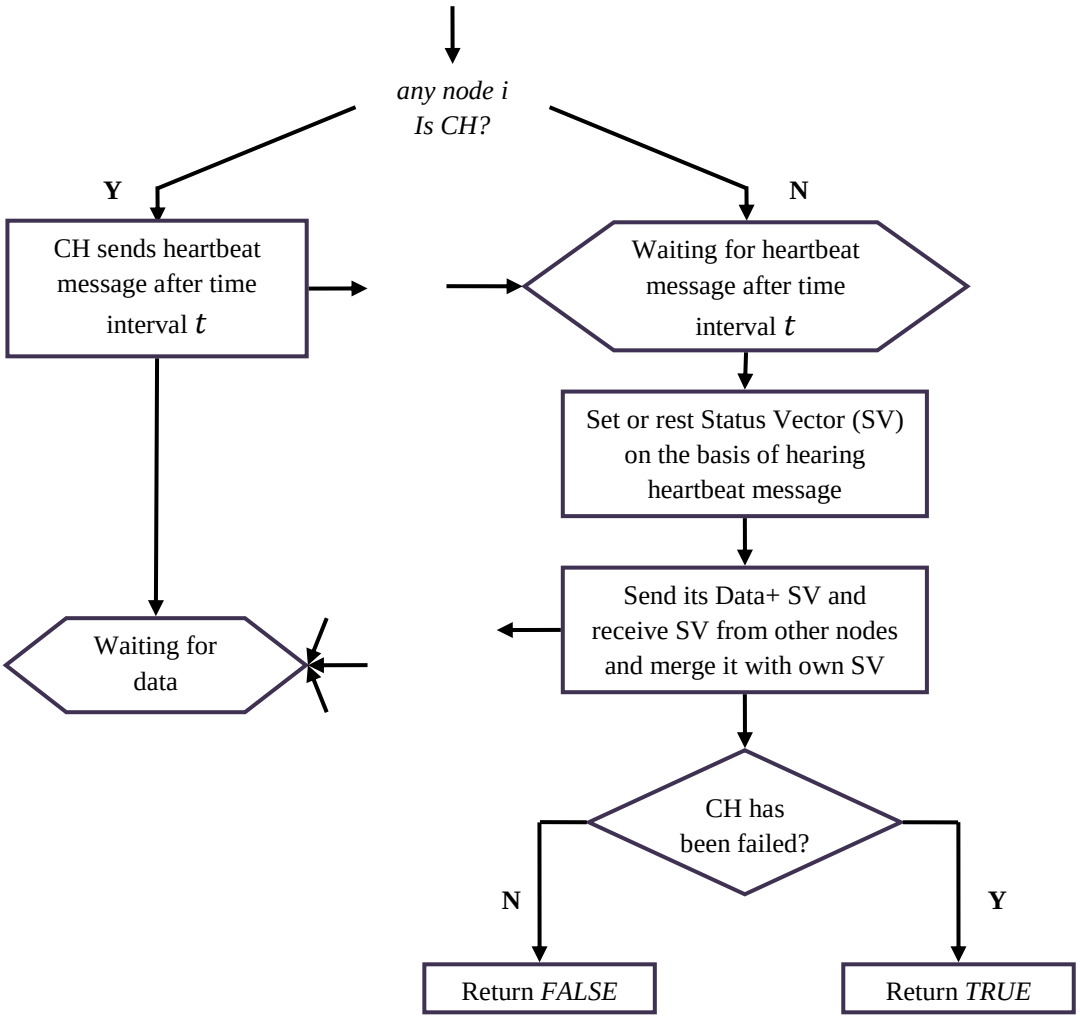

Figure 3.10 Flow Chart for CH Failure Detection

Failure Recovery: By using agreement protocol when cluster members confirm about CH then cluster member who has last slot in TDMA schedule informs to backup node about failure. Backup node elects itself as a CH by sending an advertisement message in high power transmission mode (as shown in Figure 3.4). It keeps on working as CH till its residual energy level reaches a critical limit or it fails. New backup node is required for new CH depending on application, so CH start election process for new backup node by sending e^{max} in low power transmission mode. Backup node election process is similar to election process of CH.

17 Chapter Summary

In this chapter we describe system model with various assumptions. After that proposed protocol HNFRWN is explained in detail.

- HNFRWN takes the advantageous features of distributed approaches and agreement protocol.
- In this each node makes a certain level of decision about failure.
- It is a load balancing protocol because CH is rotated after a certain threshold energy value.
- As it uses clustering approach, hence it reduces global traffic.

Chapter 4

RESULTS AND DISCUSSION

18 Introduction

In previous chapter, we have explained our proposed protocol for fault detection and recovery for CH in a cluster-based wireless sensor networks, called HNFRWN . In this chapter, we analyze the performance of our protocol. We will show how our protocol outperforms FTEP in terms of energy efficiency and accuracy. This chapter explains simulation environment parameters and detailed analysis of results.

19 Simulation Analysis

3.4.3 Simulation Setup

- A square field of 100 X 100 m^2 is taken. In this field 10 nodes are deployed randomly and one BS fixed at origin (0, 0). Out of these nodes node 1 is CH, node 2 is backup node and others are cluster members.
- The basic simulation parameters are given in Table 4.1.

Table 4.1
Experimental Parameters

Parameter	Value
Area of sensor field	100×100 m^2
Sink position	At origin (0,0)
Initial energy per node	1J
Sensing Interval	0.5 s
High transmission range	60 m
Low transmission range	20 m
Cluster Size	10, 20, 30

3.4.4 Simulation Metrics

In order to evaluate the performance of HNFRWN protocol, we take following metrics/clustering attributes:

***CH election overhead*:** It is defined as energy consumed in electing a CH in a network. It is the energy consumed by total number of messages exchanged among sensor nodes for electing CH.

***Network lifetime*:** This metric gives the time up to which a network remains alive. It shows number of rounds (including fault tolerance) up to which network remains alive for different number of nodes in network. One round consists of an operation of network from sensing the phenomenon to receiving data at sink node including election process and fault handling if any.

***Energy Consumed in Electing new CH and Back up Node*:** It shows energy consumption in fault handling with different number of faulty nodes. Energy consumption is the energy consumed by message exchange to elect CH and back up node.

Detection Accuracy: It shows how accurately fault can be detected by nodes. The detection accuracy is defined by the probability of false alarm, which is the probability that an operational CH is mistakenly detected as a faulty one. Detection accuracy performance is measured under different packets loss rates and cluster sizes.

3.4.5 Simulation Run

For our proposed protocol HNFRWN , Initially CH sends advertisement message to form cluster with low transmission power mode. Sensor nodes that are in range of CH send join message and form cluster.

After formation of cluster, CH creates TDMA schedule and sends to cluster members

After getting TDMA schedule, cluster member send data to CH according to its time slot.

3.4.6 Simulation Results and Discussion

To analyze results, we executed HNFRWN protocol with different number of nodes, number of times and failure frequency.

CH election overhead: It can be observed from Figure 4.1 that HNFRWN consumes slightly more for CH failure recovery as compared to FTEP. This is because HNFRWN elects back up node as new CH and also elects new back up node for new CH which results into more number of messages exchanged. In FTEP, back up node is not elected for new CH.

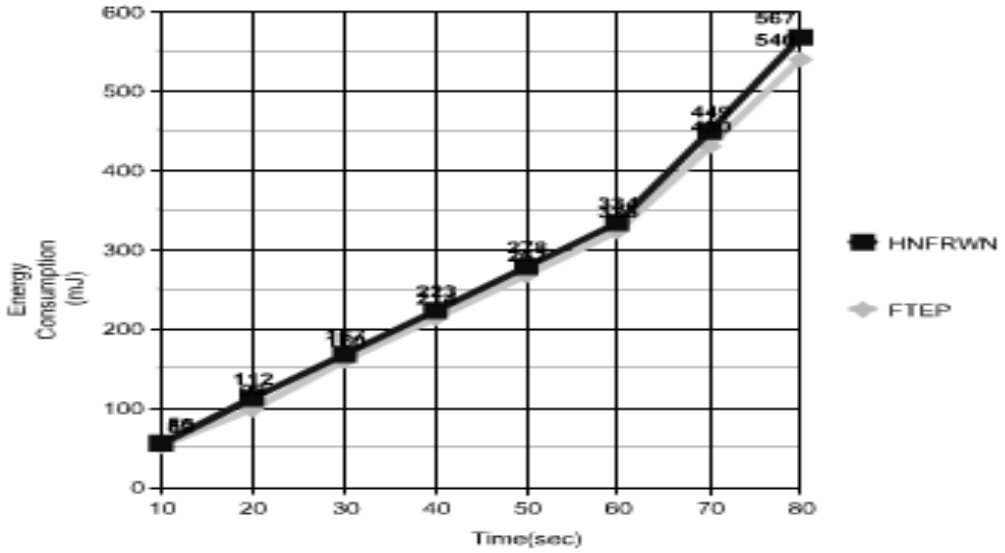

Figure 4.1 CH Election overhead

Network lifetime: As the number of nodes increases, network lifetime increases. But after certain number of nodes, the network life time starts decreasing due to more overhead of cluster maintenance.

***Energy Consumed in Electing Back up Node*:** From 4.2 shows the energy consumption in fault recovery increases as the number of failures increases. HNFRWN consumes slightly less energy as compared to FTEP.

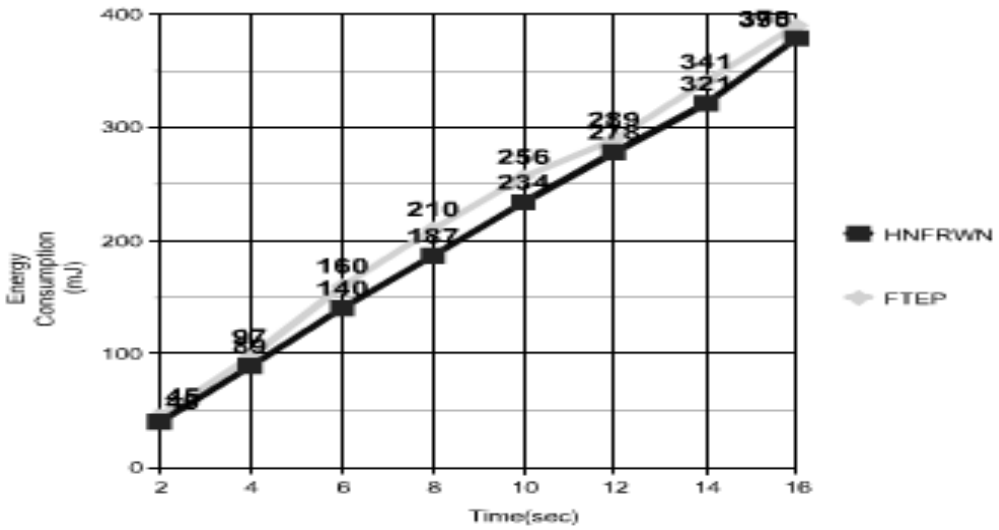

Figure 4.2 Energy Consume in Fault Handling

Detection Accuracy: we can observe the effects of the packet loss rate on detection accuracy for different cluster size. For simulation, we consider the packet loss rate range from 0.2 to 0.4. It can be observed that with the increase of the packet loss rate the probability of false alarm positive increases, which leads to lower detection accuracy. A larger number of sensor nodes lead to a smaller probability of false alarm positive, i.e., higher detection accuracy. As expected, HNFRWN can achieve high detection accuracy.

20 Chapter Summary

In this chapter we analyse the performance of our protocol and discuss results. For that following environment is used:

- From results, HNFRWN consumes slightly more energy in fault handling.
- HNFRWN detects fault with more accuracy.

Chapter 5

CONCLUSION AND FUTURE WORK

20.1 Conclusion

Wireless sensor networks have limited battery power. Most of energy of sensor nodes is consumed in communication. To save communication energy data aggregation is performed for which different approaches are used. Clustering is an energy efficient aggregation approach. As we know sensor nodes are prone to failure due to uncontrolled and hostile environment. Lots of protocols have been proposed for handling CH failure and maintain connectivity of network.

In our thesis work, we design an agreement-based distributed and scalable fault detection and recovery protocol for CH which runs parallel with normal operation of the network. It uses advanced features of clustering and agreement protocol for energy conservation and precision respectively. In this, each cluster member makes a certain level of decision individually and independently for which each member maintains a status vector. This status vector is updated periodically after getting heartbeat message from CH. When all cluster members are sure about failure, then backup node is elected as a CH. CH is elected on the basis of residual energy of sensor node from same cluster which reduce the overhead of re-clustering. Thus, our protocol increases detection accuracy and saves energy by electing CH from same cluster.

It is an energy efficient and load balancing protocol. It detects fault with high accuracy as compared to where only backup node detects faults by getting periodically state information from cluster head. It elects CH on the basis of residual energy. Simulation results show that if number of cluster members are more in a cluster detection accuracy increases. If packet loss rate increases, detection accuracy decreases.

20.2 Future Work

Proposed protocol has scope for future as follows:

- Our protocol works on cluster-based approach and use only single-level hierarchy. It can be extended for multi-level hierarchy to make it scalable.
- We assume that there is single hop communication within cluster and each node is in communication range of other node. It can be extended for multi-hop communication within cluster for more energy conservation.

- It can be modified for heterogeneous network where few nodes have comparatively rich resources than others. Such nodes as a CH for lifetime of network.

REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey On Sensor Networks," *IEEE Communications Magazine*, Volume 40, Number 8, pp. 102-114, 2002.

[2] Yang Yu, Viktor K Prasanna and Bhaskar Krishnamachari, "Information Processing and Routing in Wireless Sensor Networks," Dec 2006. or http://www.worldscibooks.com/compsci/6288.html

[3] Mohammad Ilyas and Imad Mahgoub.,"Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems," 2005.

[4] Jennifer Yick, Biswanath Mukherjee, Dipak Ghosal, "Wireless sensor network survey," *Computer Networks Elsevier,* Vol. 52, pp. 2292–2330, 2008.

[5] V. Raghunathan, C. Schugers, S Park, and M.B. Srivastava, "Energy-Aware Wireless Microsensor Networks," *IEEE Signal Processing Magazine*, Volume 19, Number 2, pp. 40-50, 2002.

[6] L.F.W. van Hoesel, "Sensor on Speaking Terms-Schedule Medium Access Protocol for WSN," Phd Dissertation, pp. 171-177.

[7] C. Talarico, J.W. Rozenblit, V. Malhotra, and A. Stritter, "A New Framework For Power Estimation Of Embedded Systems," *Computer*, volume 38, number 2, pp. 71-78, 2005.

[8] ElizabethGoff, http://www.thecourse.us/Students/Wireless_Sensor_Networks.htm.

[9] Ismail H. Kasimoglui Ian .F. Akyildiz, "Wireless sensor and actor: research challenges," *Elsevier Journal*, Vol. 2 (38), pp. 351-367, 2004.

[10] R. K. Ghosh , Vijay Garg , M. S. Meitei , S. Raman , A. Kumar and N. Tewari, "Dense cluster gateway based routing protocol for multi-hop mobile ad hoc networks," *Ad Hoc Networks*, Vol. 4, No. 2, p.168-185, March, 2006

[11] Jonathan Jen-Rong Chen Prasan Kumar Sahoo and Ping-Tai Sun, "Efficient security mechanisms for the distributed wireless sensor networks," *Proceedings of the IEEE Third International Conference on Information Technology and Applications (ICITA'05)*, Sydney, Australia, pp.541-546, 4-5 July, 2005.

[12] Sajid Hussain and Abdul W. Matin Jodrey, "Energy efficient hierarchical cluster-based routing for wireless sensor networks," *Technical Report - TR-2005-011*, 2005, 073720m@acadiau.ca.

[13] A.A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, Vol. 30, pp. 2826–2841, 2007.

[14] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless sensor networks," *in: Proceeding of the Hawaii International Conference System Sciences,* Hawaii, pp. 3005-3014, January 2000.

[15] S. Lindsey and C.S. Raghavendra, "PEGASIS: power efficient gathering in sensor information systems," in*: Proceedings of the IEEE Aerospace Conference*, Big Sky, Montana, Vol.3, pp. 3-1125- 3-1130, March 2002.

[16] S. Lindsey, C.S. Raghavendra, and K. Sivalingam, "Data gathering in sensor networks using the energy*delay metric," *in: Proceedings of the IPDPS Workshop on Issues in Wireless Networks and Mobile Computing*, San Francisco, CA, April 2001.

[17] Manjeshwar and D.P. Agrawal, "TEEN: a protocol for enhanced efficiency in wireless sensor networks," *in: Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, San Francisco, CA, April 2001.

[18] Manjeshwar and D.P. Agrawal, "APTEEN: a hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks," *in: Proceedings of the 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile computing*, Ft. Lauderdale, FL, pp.195-202, April 2002.

[19] Ossama Younis and Sonia Fahmy, “HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks,” *IEEE Transaction on Mobile Computing*, Vol.3, No.4, pp. 366-379, October-December 2004.

[20] M. Bani Yassein, A. Al-zou'bi and Y. Khamayseh, W. Mardini, “Improvement on LEACH Protocol of Wireless Sensor Network (VLEACH),” *JDCTA: International Journal of Digital Content Technology and its Applications*, Vol. 3, No. 2, pp. 132 -136, 2009.

[21] Udit Sajjanhar and Pabitra Mitra, “Distributive energy efficient adaptive clustering protocol for wireless sensor networks,” *in Proceeding of International Conference on Mobile Data Management (MDM07)*, Mannheim, Germany, pp. 326-330, May 7 -11, 2007.

[22] Omar Moussaoui and Mohamed Naimi, “A distributed energy aware routing protocol for wireless sensor networks,” *in Proceeding of ACM PEWASUN’05*, Montreal, Quebec, Canada, pp. 34-40, October 10-13 2005.

[23] Bhuvaneswari P.T.V, Vaidehi V and Shanmugavel S, “SPEAR: sensor protocol for energy aware Routing in wireless sensor network,” *in Proceeding of IEEE Third International Conference on Wireless Communication & Sensor Networks (WCSN -2007)*, Allahabad, December 13-15, pp.133-137, 2007.

[24] Amir Sepasi Zahmati, Bahman Abolhassani, Ali Asghar Beheshti Shirazi and Ali Shojaee Bakhtaran, “An Energy-efficient protocol with static clustering for wireless sensor networks,” *Internationa Journal on Electronics, Circuits and Systems,* Volume. 1, issue 2, pp. 135-138, 2008.

[25] Fan Xiangning and Song Yulin, “Improvement on LEACH Protocol of Wireless Sensor Network,” *International Conference on Sensor Technologies and Applications,* pp. 260-264, Oct.14-20, 2007.

[26] V. Loscrì, G. Morabito and S. Marano, “A Two-Levels Hierarchy for Low-Energy Adaptive Clustering Hierarchy,” *IEEE 62nd Vehicular Technology Conference (VTC-2005-FALL)*, Vol.3, Sept. 25-28, 2005.

[27] Wendi B. Heinzelman, Anantha P. Chandrakasan and Hari Balakrishnan, “An Application-Specific Protocol Architecture for Wireless Microsensor

Networks," *IEEE Transaction on Wireless Communications*, Vol. 1, No. 4, pp. 660-670, Oct. 2002.

[28] Sajid Hussain and Abdul W. Matin, "Hierarchical Cluster-based Routing in Wireless Sensor Networks," *in Proceeding of 5th Intl. Conf. on Information Processing in Sensor Network (IPSN06)*, USA, April 19-21 2006.

[29] Chuan-Ming Liu, Chuan-Hsiu Lee and Li-Chun Wang, "Power efficient communication algorithms for wireless mobile sensor networks," *in Proceeding of ACM PEWASUN' 04*, Venice, Italy, pp. 121-122, October 7, 2004.

[30] Dissertation, Hang Zhou, Zhe Jiang and Mo Xiaoyan, "Study and Design on Cluster Routing Protocols of Wireless Sensor Networks," 2006.

[31] Luciana Moreira S´a de Souza, Harald Vogt and Michael Beigl, "A Survey on Fault Tolerance in Wireless Sensor Networks," http://www.cobis-online.de/.

[32] Peng Jiang, "A New Method for Node Fault Detection in wireless Sensor Networks" *Journal of Sensors 2009*, Vol. *9*, 2009, pp. 1282-1294, www.mdpi.com/journal/sensors/.

[33] Winnie Louis Lee, Amitava Datta and Rachel Cardell-Oliver, "WinMS: Wireless Sensor Network-Management System," *An Adaptive Policy- Based Management for Wireless Sensor Networks,* Tech. Rep. UWA-CSSE-06-001 2006, UWA, Australia.

[34] Nithya Ramanathan, Kevin Chang, Rahul Kapur, Lewis Girod, Eddie Kohler and Deborah Estrin, "Sympathy for the Sensor Network Debugger" *In 3rd Embedded networked sensor systems,* 2005, San Diego, USA: ACM Press.

[35] Farinaz koushanfar, Miodrag Potkonjak, and Alberto Sangiovanni-Vincentelli, "Fault Tolerance Techniques for Wireless Ad Hoc Sensor Networks," 2000.

[36] S Harte, A Rahman and K M Razeeb, "Fault Tolerance in Sensor Networks using Self-Diagnosing Sensor Nodes," *in IE2005,* 2005: IEEE.

[37] Jinran Chen, Shubha Kher and Arun Somani, "Distributed Fault Detection of Wireless Sensor Networks," *in DIWANS'06,* 2006, Los Angeles, USA: ACM Pres.

[38] Anmol Sheth, Carl Hartung and Richard Han, “A Decentralized Fault Diagnosis System for Wireless Sensor Networks,” *in 2nd Mobile Ad Hoc and Sensor Systems,* 2005, Washington, USA.

[39] Nidhi Bansal, T. P. Sharma, Manoj Misra and R. C. Joshi, “FTEP: A Fault Tolerant Election Protocol for Multi-level Clustering in Homogeneous Wireless Sensor Networks,” *16th IEEE International Conference on Networks, ICON 2008*, Dec, 2008.

[40] M.Asim, H.Mokhtar, and M.Merabti, “A cellular approach to fault detection and recovery in wireless sensor networks,” *Third International Conference on Sensor Technologies and Applications 2009, SENSORCOMM '09*, pp. 352 – 357, 18-23 June 2009.

[41] G. Venkataraman, S. Emmanuel and S.Thambipillai, “A Cluster-Based Approach to Fault Detection and Recovery in WSNs,” *IEEE ISWCS 2007*, pp. 35-39, 2007.

[42] G. Venkataraman, S. Emmanuel and S.Thambipillai,“Energy-efficient cluster-based scheme for failure management in sensor networks” *IET Commun*, Volume 2, Issue 4, pp.528 – 537, April 2008.

[43] Kai Wai Fan, Sha Liu and Prasun Sinha,“ Structure-free data aggregation in sensor networks,” *IEEE Transactions on Mobile Computing,* Vol.**6** (8), pp. 929–942, 2007.

[44] OMNeT++ website, www.omnetpp.org.