

**IMPLEMENTATION OF NEW FAULT TOLERANCE SOLUTION IN WIRELESS
SENSOR NETWORKS IN A MULTI-CHANNEL CONTEXT**

A

THESIS

**Presented to the
Department of Computer Science
African University of Science and Technology**

In Partial Fulfillment of the Requirements

For the Degree of

MASTER OF SCIENCE

BY

OHIDA SEFIYAT OYIZA

(40413)

SUPERVISED BY

PROF. OUSMANE THIARE

Abuja, Nigeria

May, 2016

ABSTRACT

Wireless sensor network is application specific, which is deployed in an interested area like about hundred or thousands of sensor nodes. All the sensor nodes communicate via a wireless medium and works cooperatively to sense the environment in order to achieve the required task. Such sensor nodes which is application specific needs a good fault tolerance scheme to keep the system working. Since this sensor nodes are battery operated, have a small memory, deployed in harsh environmental condition and can easily be depleted. So we improved on distributed fault tolerance algorithm which enable the detection status of a node with fewer neighbors to be accurate and can also detect the status of a node in a multi-channel context. Our distributed fault tolerance algorithm uses majority voting with priority. Although in literature fault tolerance detection in multi-channel channel context has not really been put into practice.

ACKNOWLEDGEMENT

My sincere acknowledge goes to almighty Allaah for giving me the strength, health, wealth and bounty mercy to enable me round up this MS.C and also the grace to complete my thesis. You will forever be glorified.

I heartily acknowledge Prof. Ousmane Thiare, my supervisor, for his continuous to encourage, support and guide me in my research work. I am indebted to him for his help in my professional.

I would like to extend my utmost and deepest gratitude to Prof. Traore, the head of the department of computer science.

My profound gratitude goes to my lovely mother Mrs Ohida Barikisu, to my father Mr Ohida Yakub (a.k.a Adabara), to my biggest brothers Wahab and Itopa, to my biggest sisters Salamat and Hadiza, to my three wonderful younger brothers Muhammed, Lucky and Onimisi, to my sister from different parents Sabitiu and also to my wonderful and lovely husband Kenny Salami.

I do not want to leave out those who has contributed to my life in one way or the other, in the person of Prof. Abiodun Musa Aibinu, Dr Alhassan John, Mr Femi Osho, Mr Ojeniyi and Prof Waziri all in Federal University of Technology Minna.

My appreciation also goes to my colleagues in AUST, my stubborn friend Bello Habibah, Umar Rahmah, and also to my very good friend Ahmad Rufai Enesi Abdullah. To mention but little, all others who are not here are also of important.

DEDICATION

This thesis is dedicated to almighty Allaah for his infinite mercy and also to my lovely mother Mrs Ohida Barikisu, may Allaah give you long life and prosperity to reap the work of your labour ameen.

CONTENTS

Table of Contents

ABSTRACT	I
ACKNOWLEDGEMENT	II
DEDICATION	III
1 INTRODUCTION	
1.1 Introduction	1
1.2 Problem Statement	3
1.3 Objective	4
1.4 Motivation of Work	4
1.5 Significance of Study	4
1.6 Thesis Organization	5
2 STATE OF ART	
2.1 State of Art of Multi-channel Communication in WSNs	6
2.2 Discussion of Fault-Tolerance in WSNs	7
2.3 Fault Types	7
2.4 Fault-Tolerance Occurrence at Different Level	
2.4.1 Hardware Level	8
2.4.2 Software Level	8
2.4.3 Application Level	8
2.4.4 Network Communication Level	9
2.5 Classification of Fault-Tolerance Techniques	9
2.6 Fault Detection Approaches in WSN	9
Centralized Detection Approach	9

Decentralized or Distributed Approach	10
2.7 State of Art of Fault-Tolerance in WSN	11
3.0 Improved Distributed Fault-Tolerance Algorithm	
3.1 Network Model	14
3.2 Fault Model	15
3.3 Existing Distributed Fault-Tolerance Detection Algorithm	16
3.4 Improved Distributed Fault-Tolerance Detection Algorithm	17
3.4.1 Algorithm By Peng Jiang	18
3.4.2 Step By Step Improved Algorithm	19
3.5 Discussion of Improved Algorithm	20
4.0 Simulation or Proofs of The Proposed Algorithm	
4.1 Theorems and Proofs of the Improved Distributed Algorithm	22
4.2 Analysis of The Fault-Tolerance Algorithm	24
4.3 Simulation With Omnetpp4.2 and Castalia 3.2	25
4.4 Simulation Result	26
5.0 Recommendation and Conclusion	
5.1 Resolved Issue	29
5.2 Unresolved Issue	29
5.3 Conclusion	29
5.4 Challenges	30
5.5 Future Work	30
References	31

LIST OF FIGURES

1.1 WSN Architecture	1
1.2 Component of a Sensor Node	1
2.1 Fault Tolerance Types	7
4.1 Example of Network Deployment	24
4.2 Omnet++ Model	25
4.3 Node Connection	25
4.4 Module and Connection Process in Castalia	26
4.5 Tkenv of The Omnetpp Network Topology	27
4.6 Acknowledgement Message	27
4.7 Movement of Acknowledgement	28

LIST OF TABLES

3.1 Description of the Notations Used	17
4.1 Analysis of The Example	24

CHAPTER ONE

1.0 INTRODUCTION

1.1 Introduction

Wireless sensor network consists of a number of sensor nodes and a base station (BS). A wireless sensor network is a collection of nodes organised into a cooperative networks [1], Sensors contain an on-board processor, perform simple computations within itself by using their processing capabilities to process their raw data before transmitting it to the central node sent to the BS. Wireless sensor nodes are low power, battery operated devices with limited computation and transmission ability [2].

Sensor node deployment in wireless sensor network architecture is shown in fig 1 and the component of sensor node its self is shown in fig 2.

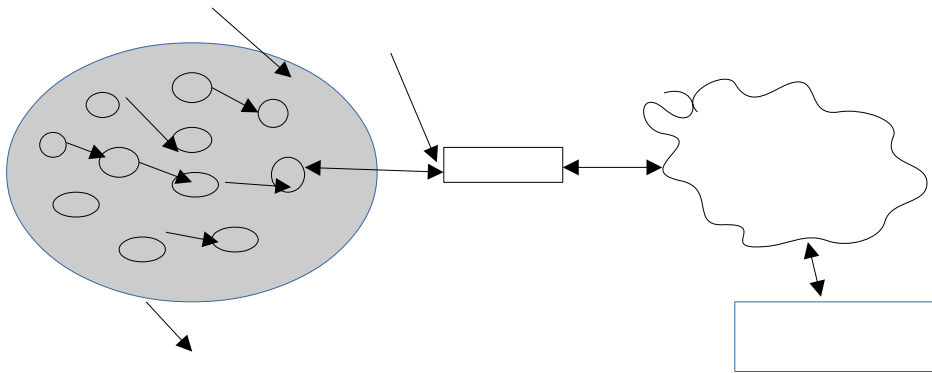


Figure 1.1 Wireless sensor network Architecture

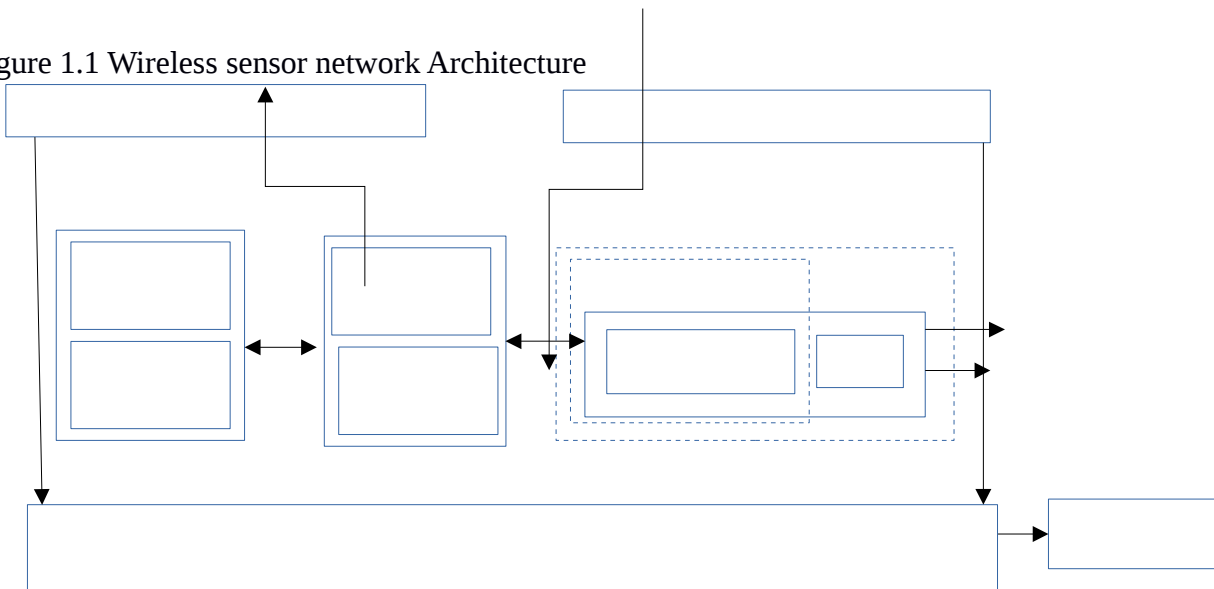


Figure 1.2: component of a sensor node

In wireless sensor network (WSN) communication using the multi-hop mode where each node communicate with the base station using an intermediate node, thus the node closer to the base station of course become the only source through which all other node send their data to the base station which give rise to interference and low throughput since all other node compete to use node closer to the base station.

The WSN technology tremendous improvement has gained application in so many areas of our daily routine such as medical care, environmental monitoring, smart buildings, banks, telecommunication industry, many other industry and military application. Most of these sensor application are in harsh environment which can cause WSNs to be prone to failure as compared to other wireless networks, this involve safe mobility and performance, data quality and energy consumption. Data quality is defined as the number of readings received by the user divided by total number of readings generated by the network during an observation period [3]. To preserve resource and achieve high quality of data, we identify the following as key requirements for FT in WSNs [3]:

1. Awareness of the network main operation and the status of the network resources.
2. Adaptability to the frequent changes in WSNs conditions.

The Physical harsh environmental condition that can affect WSNs deployment are fire, rain, humidity, floods and any other physical thing that can affect it. All these can actually cause the sensor nodes to fail or transmit error messages. To guarantee the network quality of service and performance, it is essential for the wireless sensor networks to be able to detect faults, and to perform something akin to healing and recovering from events that might cause faults in the network, hence fault tolerance should be seriously considered in many wireless sensor network deployments [7].

Fault tolerance is the mechanism put in place for sensor nodes to keep working after a failure occurred. fault tolerance is a need in this type of networks due to sensor node characteristics, radio communications and hostile environments in which these networks are deployed [4]. To successful

utilize full deployment of WSNs, fault tolerance should be put in place. One approach to achieve fault tolerance in wireless sensor networks is to deploy a small number of additional relay nodes to provide k ($k \geq 1$) vertex-disjoint paths between every pair of functioning devices (including sensors, data sinks, and other wireless equipments, all termed target nodes in this paper) so that the network can survive the failure of fewer than k nodes [6].

1.2 Problem Statement

Most of the implementation of wireless sensor networks are done in environment with harsh conditions and as such the batteries of the sensor nodes prove very difficult to charge or replace immediately when the batteries run off. The mode of communication in WSN include single-hop and multi-hop. In multi-hop wireless sensor networks bandwidth is a critical issue, because of interference between successive hops on the same path as well as that between neighbouring paths [5]. Hence in order to solve this problem of interference and maximize throughput and also to overcome fault thus several fault tolerance scheme are been diverse. Using a single channel for communication causes a heavy work load on it and many packets do not get to their destination due to collision, so when a node fail the entire network stop functioning. Due to this disadvantage a CSMA/CA or CD are been used to enable parallel communication but because of the heavy work load, this solution do not provide high channel utilization. Another cost effective solution is to use multiple channels that works for parallel data transmission based on the current WSN hardware, such as MICAz and Telos that provide multiple channels with single radio [5]. Thus in WSNs, there are issue of node failure and transmission failure which are the two main fault in Wireless sensor networks.

Fault is one of the major impact in building a Sensor networks, the ability to identify node fault in sensor network deployment is one of the major issue. Retransmission problem for the detected faulty node also account for in WSNs deployment. A faulty sensor node can send incorrect message to the base station. Imagine in military deployment of sensor nodes where sensors are expected to

detect the enemy, if such sensor failed then the whole military are in serious danger.

1.3 Objectives

The main objective of these thesis is to provide an adequate solution to fault tolerance in wireless sensor networks in a multi-channel context. The general objectives are:

1. To carry out state of the art of multi-channel communication and fault tolerance in wireless sensor networks
2. To implement a proposed algorithm for decentralized fault tolerance detection in wireless sensor network.
3. To Simulate the implemented algorithm
4. To compare implemented algorithm with already existing algorithm

1.4 Motivation of work

From our research and findings we discovered that most of the papers in literature never care about fault tolerance in multi-channel context. Their algorithm never put into consideration sensor nodes with all faulty neighbors. Some of these papers just focus on finding faulty nodes without applying the algorithm in a gradual process. That means making sure that nodes with fewer neighbors are treated last and nodes with large neighbors are treated first. Doing this ensure that sensor nodes are not mis-diagnose.

1.5 Significance of Study

There are several methods in literature for solving the problem of node failure in wireless sensor network deployment. Our improved distributed fault tolerance algorithm will enable the researchers and wireless sensor network application deployment to be able to detect node failure as fast as possible and at the same time use an alternative channel in transmitting data to the base station using the one hop neighbors.

1.6 Thesis Organization

The remaining part of these thesis is organised as follows, chapter two gives state of the art of multi-channel communication and fault tolerance in WSNs, chapter three gives the materials and methods of the thesis, chapter four gives the implementation of the algorithm and simulation and comparison with existing work and chapter five gives conclusion and future work.

2.0 STATE OF THE ART

2.1 State of Art for multi-channel communication in WSNs

The main goal of multi-channel communication protocols in WSNs is to maximize energy efficiency in order to increase the network lifetime. Due to this advantage, the state of the art research, has proposed a good number of multichannel protocol for communication in WSNs. The authors in [8], proposed a tree-based multichannel protocol which enable data collection application thereby allocating channels to the disjoint trees in order to exploit parallel transmission of data among the trees. The application specific end to end communication delay which is critical to real time communication has not been really pay attention to in the existing work. In [29], a Multichannel protocol was proposed MCRT for such real time communication using a flow based channel allocation strategy, that allocate channels to network based on many to one data flows and this protocol includes a real time packet forwarding strategy.

Channel assignment with minimum interference known as 2-hop colouring problem allows repetition of colours, so in [2], proposes a Dynamic channel allocation (DCA) algorithm as a novel solution for the 2-hop colouring problem that optimally assign minimum channel in a distributed manner to avoid interference. In reference [30], A MMAC is proposed for ATIM window such that the multi channel phase is splited at beginning of the beacon interval. The nodes that have packets to transmit negotiate channels with the destination nodes during this window and if the channel is acquired the transfer can be done.

Ramakrishnan and Ranjan [31], proposed SMC MAC, which uses a single dedicated control channel and eight data channels. It initialized all the nodes to stay in control channels such that channel negotiation is done via RTS/CTS control packets and to transmit data the first free channel is selected. Also in [32], CAM MAC was presented that uses handshake which consist of three phase for communication and this protocol solves multi-channel problems using a single transceiver

and completely eliminates synchronization through the exploitation of cooperation.

2.2 Discussion of Fault Tolerance in Wireless Sensor Network

The fault tolerance hierarchy is given based on different kind of fault in WSNs [13], this fault includes: connectivity fault, link fault, node fault and malfunctioning fault. The link fault needs relay with high power capabilities to be place between two nodes for sending information to its destination. Communication graph is used for identifying the fault.

2.3 Fault Types

There are a lot of authors who have presented fault types in their different opinions and these authors give different fault types [12], [26], [28], although there are similarities in their differentiation. The two main types of fault in wireless sensor networks deployment include node fault and transmission fault. The figure shows the fault types.

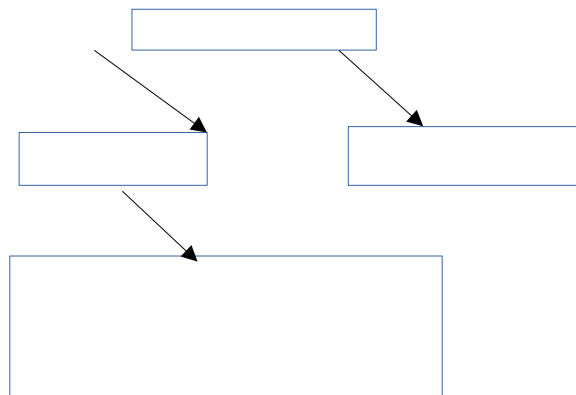


Figure 2.1: fault types

2.4 Fault tolerance occurrence at different level

Fault tolerance occurs at different levels in wireless sensor networks. There are five levels at which faults occur in WSNs [24] such as Physical layer, hardware layer, system software layer, middleware layer and application layer. According to the study, there are four main layers at which faults occur [4], [22], [23] which include hardware layer, software layer, application layer and network

communication layer.

2.4.1 Hardware level

The occurrence of fault in this layer is due to the malfunctioning of one or more sensor components such as battery, memory, wireless radio, microprocessor and sensing unit. The failure of a hardware of sensor node is said to occur due to: the quality of the low cost sensor node, limited power supply which gives incorrect reading when it falls below a certain threshold and the harsh environment in which this sensors are deployed can affect it components.

2.4.2 Software level

The sensor node software comprises of system software, which includes operating system and middleware which entails communication, routing and aggregation. The main error source in this layer in wsn is software bugs. This layer support distributed and simultaneous execution of localized algorithm. For example, in the case of energy minimization under functionality constraint requirements, several protocols have been developed for the coordination of distributed actions [24]. Since it is difficult to provide fault tolerance in an economic way at hardware level of a sensor node, numerous fault-tolerant approaches are expected at the middleware level [23].

2.4.3 Application level

Different application requires different way to handle fault tolerance scheme at this level which based on their requirements. But fault tolerance at this level can be used to address faults in any type of resources.

2.4.4 Network communication level

The fault in this layer is very sensitive because it deals with the wireless communication link it self which is very prone to failure. This fault are caused as a result of the harsh environmental condition and the interference between the sensor nodes. If two nodes are ready to communicate and one node

is within interference range of other nodes that are transmitting messages at the same time, then this two node will not be able to communicate. Inorder to overcome these, is to use error correction, retransmission and multi-channel communication.

2.5 Classification of fault tolerance techniques

Recently researcher has developed variety of techniques to handle different types of faults at different layers of the network. The classification at different level of WSNs layer is given by [22] and it explanation can be found in [22].

1. fault prevention
2. fault detection
3. fault isolation
4. fault identification
5. fault recovery

2.6 Fault detection Approaches in WSNs

Fault detection is the ability to discover fault in wireless sensor network. In fault management scheme, fault detection is the first step taken to identify failure and should be done properly by the network system. Fault detection are broadly group into two [22], [25], [26], [27], namely centralised and distributed or decentralized Approach.

2.6.1 Centralized detection Approaches

This Approach is a very common solution in WSNs where the base station or sink or a task manager detect and manages fault in the sensor network. The Central node adopts an active detection model to take back the states of network performance and individual sensor nodes by periodically injecting

queries to the network. Central approach analyzes this information to identify the failed nodes [27]. Thus every node in the sensor network will have to acquire information from the central node to know which node in the network failed or not functioning again. One of the issues with centralized approach is that the central node energy can easily deplete and heavy message traffic although it is efficient in some ways. The figure 3 below shows the diagram for centralized detection approach.

2.6.2 Distributed or Decentralized detection Approaches

in this approach each sensor has the capability to make local decision in order to identify their own faults and that within their neighbor. In this approach each sensor node sends an update message to each of its neighbors within a specific time interval, and the neighbor detect the existence of the node and send its status to the base station. The neighbor wait for ACK and if no ACK within this time interval, it send message to the base station and declare that node may be down. This approach reduces the work load on the fusion center since it only handle special cases of node failure. In order to do this, there exist node self detection, neighbor coordination and clustering Approach are the fault management distribution here.

- **Node Self Detection:** In this sensor node observes their binary output and compare it with the predefined fault models, therefore energy depletion is detected by the node itself.
- **Neighbor Coordination:** In this approach nodes coordinate with their neighbor to identify the failed nodes before contacting the fusion center. Here sensor nodes are expected to know of its physical location using expensive GPS or other GPS-less technology [24].
- **Clustering Approach:** The authors in [15] has presented cluster-based techniques for efficient node failure detection whereby the cluster head send a heartbeat messages to identify node failure. Clustering has become useful for building scalable and energy balance application for WSNs.

2.7 State of the Art of Fault Tolerance in WSNs

In the state of art of fault tolerance in WSNs many fault tolerance scheme has been proposed and implemented to spot faulty sensor nodes or recover from fault. Giving below briefly explained some of the fault-tolerant solution put in place from the existing fault tolerance mechanism in literature to handle fault or recover from failure in WSNs . Many of this mechanism are either distributed or centralised solution applied to the nodes, cluster heads(CH), base station(BS). The ability of WSNs to keep working during failure or recover from failure is very important, energy consumed can be minimised and link failure can be corrected.

The authors in [3], presented a general framework for fault tolerance in WSNs. The framework was applied on CRAFT, which is based on Checkpoint/Recovery to realize high QoD in the presence of faults and this scheme was describe in detailed [9]. The main idea of CRAFT is to tolerate failure of the sink by applying periodical data checkpointing, giving three sink S1, S2, and S3 when S3 failed it sends update to S2 to take over and so on. When this scheme was compared with NOFT, it actually consume less energy packet while also maintaining higher quality of data.

Panda and Khilar [10], proposed a distributed fault detection algorithm for soft faulty(error in message transmission) sensor node in a sparse WSNs. In this scheme every sensor node gather information only from their neighbour in order to reduce communication overhead, to predict fault status of each sensor nodes and it neighboring node a Neyman Pearson testing method is deployed and a voting scheme is used to get the final fault status of each sensor node. And when compared with Jiang and JSA algorithm it outperformed them. In [13] An adaptive distributed fault tolerance scheme was designed to maintain high detection accuracy (DA) and low false alarm rate (FAR) with increased number of sensor nodes in WSNs. Then each sensor nodes construct a neighbor table which include the reading of their own sensed data compared with the results of their neighbor and

giving decision to them if necessary. The algorithm is very scalable exception of the case where average degree of node is very low. Bhaskar and Sitharama [20], uses Bayesian algorithm for fault detection, the setback using this algorithm is that new error can be introduced if evidence from neighbor sensor is faulty.

Fault recovery and detection scheme FTMRS proposed in [12], the energy efficient fault tolerance multipath routing scheme (FTMRS) allows all nodes to transmit data to the BS through a shortest path, within a minimum time and energy loss. Two backup path are used for alternative path in case of network failure and also to handle over load traffic in the main path. Although their scheme has algorithm to reuse faulty node as traffic node and their scheme outperform some other popular scheme in WSNs. Also an energy aware [15], which also increased the lifetime WSNs by considering the failure of a cluster head to be replaced by selecting a sensor nodes that sensed same data to be put to sleep mode and this sensor node is used to replaced CH in case of failure. This algorithm is directed to CH failure in the network but in [33] cluster structure is maintained in case of failure that is caused by energy depletion of node by selecting a node with maximum energy as the cluster head and the second maximum energy as the secondary CH in case of failure. In [16], the energy aware scheme (EALD) try to reduce energy, such that the amount of energy needed to transfer message through different route is determined so that the route with significant amount of energy can be used.

In [14], A disjoint path vector (DPV) for distributed fault tolerance in heterogeneous WSNs, large number of sensor node with limited energy and computing was designed. DPV is used to construct a fault-tolerant topology to route data collected by sensor nodes to super nodes. The algorithm store full path information and provide a large search scope to discover path through out the network and at the same time maintain k-connectivity and power efficiency. Each sensor node is connected to atleast on super node by k-vertex disjoint path (KDP). To explore the network path messages which

entail path information is used. The use of super node here is very costly and there is no such thing like super to super node communication.

Also a recovery algorithm [18], for faulty nodes which prolong the lifetime of the sensor nodes in case of failure based on discovering and detection of the fault. This algorithm reduces the replacement of sensor nodes and increases routing path reusability. The weighted average algorithm for distance based fault detection [19], which enable a faulty sensor to diagnose it self by comparing it sensed data with the calculated weighted average of its neighbor sensed data. In [36], node fault detection algorithm proposed based on round trip delay, the algorithm calculate the RTD value of working nodes and highest RTD time will be taken as threshold value. The faulty node is identify by comparing RTD with the threshold value, if RTD is greater than threshold value then the node is considered as malfunctioning but RTD is infinity the node will assumed as dead. Localized fault detection algorithm [21], that enable faulty sensor to be located in WSNs deployment. Here a test result is conducted such that if the test result of the neighboring nodes that is normal is less than half of the total number of the neighbor nodes then node is said to be faulty. Further improvement on this algorithm was done by Peng Jiang [17], stating that the DFD algorithm by [21] was too harsh on sensor node which likely to be good, so he proposed an algorithm that if there is no neighbor node of node S_i whose initial detection is likely to be good and node S_i initial detection is normal, then let node S_i remain Good. This last two papers is the bases of our DFD algorithm.

CHAPTER THREE

3.0 IMPROVED DISTRIBUTED FAULT TOLERANCE ALGORITHM

This chapter explicitly explained the improved distributed fault tolerance algorithm in wireless sensor network in a multichannel context. The bases of the improved distributed fault tolerance algorithm(DFD) is extracted from paper [17].

3.1 Network Model

We consider a multi-hop wireless sensor network deployment, where each node communicate with the help of a neighboring node. Then all sensor nodes are deployed geographically in an intended area. All the sensor nodes can have the same transmission range and can communicate to two hop nodes through it one hop neighbor. That means it can only communicate with it two hop nodes through it one hop node that is detected to been fault free. The sensor node deployment is shown in figure 1 with whole area covered with sensor nodes. The circles that is black is said to be a faulty (FT) node, the white circles are said be good (GD) node and the irregular shape inside the sensor deployment area is said to be a failed area in the deployment meaning no sensor nodes in the network can communicate with the sensor nodes in this area. We are using voting system but not dependent on majority voting, we assume that each sensor in the network area has atleast three neighbors. Because both large and small amount of sensor nodes can be deployed into interested area to form a wireless sensor network. But each of the node in sensor deployment can locate it neighbors within its transmission range through ACK protocol or sending a HELLO message. Considering multi-hop communication is to enable sensor nodes to be able to communicate to the fusion center or Cluster head in case of node failure in a multi-channel communication wireless sensor network deployment.

For example, probability of sensor failure [37]. Assuming p represent the probability of sensor failure and r denote the probability that a faulty node has a communication unit which is said to be fault free. Then if total number of nodes under detection is n , np nodes are faulty. $N(1-r)$ nodes are unable to communicate with their neighbors. Just $n(1-p) + npr$ nodes are involved in fault detection.

So the new probability is defined by:

$$P = \frac{npr}{n(1-p) + npr} \quad (1)$$

$$P = \frac{pr}{1-p+pr} \quad (2)$$

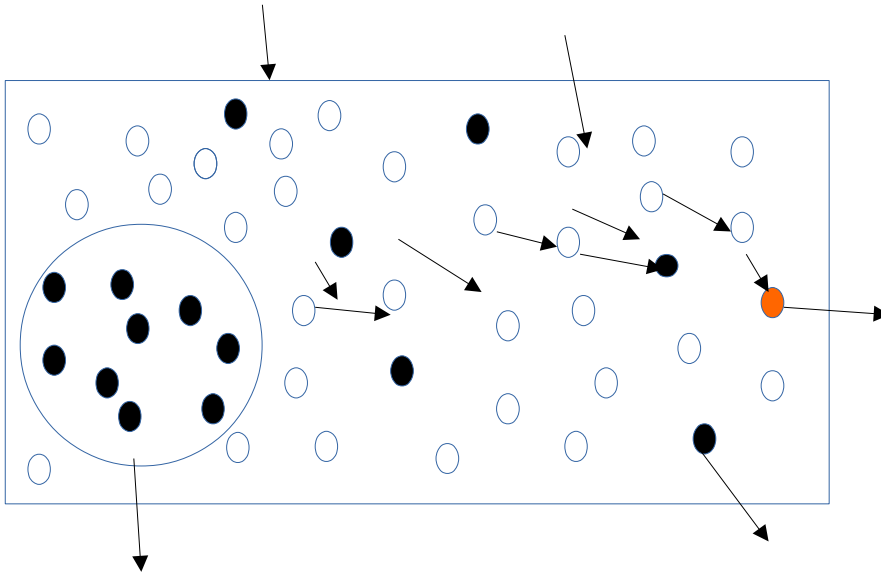


Figure 3.1: wireless sensor network deployment

3.2 Fault Model

In our sensor network deployment, we use the fault model as in [34] [35] and [17]. Faults can take place in different layers of the sensor network, which include physical layer, hardware, software and middleware layer [17]. Here we specifically focused on node faults which is hardware layer fault in the sensor network layer. The components in the hardware layer includes computation engine, power supply infrastructure and storage system, these are very reliable because BISR fault tolerance scheme will provide some kind of level fault tolerance while the hardware component such as sensors and actuators can easily malfunction. So the sensor fault we are interested in here are

calibration systematic error, random noise error and complete malfunctioning. Mostly when nodes are faulty they can still send, receive and process messages. The sensor nodes which have malfunctioning sensors can participate in network because it can still route information, the node that should be get rid off from the network are nodes with permanent fault such as lack of power in the network deployment. Therefore our fault detection algorithm to be presented can assume a good sensor node as faulty if there is no neighbor of it whose detection status is good because we are considering a multi-hop channel in a multi-channel communication where node will have to communicate via a node that is fault free.

3.3 Existing Distributed Fault Detection Algorithm

The node Distributed fault detection scheme proposed by Peng Jiang enables the nodes status to be detect by comparing the status of the node with those of it neighboring nodes. To get the test result of this nodes, if S_i is the node to be tested and S_j is the neighbor node, a test result C_{ij} is created by getting the data that is sensed from the both of them. The data at time t of the neighbor nodes should be close to each other because of their close proximity and d_{ij}^t is the difference between the data. The d_{ij}^t must not exceed a certain threshold θ_1 and at another time $t+1$, the data difference between the neighbors is d_{ij}^{t+1} and the difference between d_{ij}^t and d_{ij}^{t+1} is Δd_{ij}^t . Δd_{ij}^t must also not exceed a certain threshold θ_2 . If none of the above condition is satisfied, then we can say S_i or S_j is faulty and the test result is $C_{ij} = 1$ otherwise 0. The test result of any node S_i can be obtain with respect to each of it neighbors (S_i). If the test result C_{ij} of it neighbor (S_i) is less than half of the total number of neighbor(S_i) nodes, then the initial detection status T_i of S_i as likely normal (LG) or likely faulty(LT). Also if the number of neighbor nodes of S_i whose initial detection status is LG. Thus in order not to misdiagnose any node S_i , these approach is applied: for any node S_i and the nodes S_j which is in neighbor(S_i) with initial detection status of LG, if the nodes whose test result with S_i is 0 are greater than the nodes whose test result is 1, then the status of S_i is normal otherwise it is faulty.

If there are no neighbors of S_i whose initial detection status is LG and if the initial detection status T_i of S_i is LG, then set the status of S_i as normal GD, otherwise as faulty (FT).

3.4 Improved Distributed Fault Tolerance Detection Algorithm(IDFD)

In the improved distributed detection algorithm, the following changes is made to it. First, the algorithm ensure that, the number of neighbors for each nodes is not less than a certain predefined value because few number of neighbors will lead to the sensor node making incorrect decision thereby leading to its own mistakes and other sensors mistake so such nodes detection status is unknown. Then since we are concern with multi-hop communication in a multi-channel context, in step 3 and step 4 of Peng Jiang the following changes are made. For node S_i , if the initial detection status of its neighbors is LG and the sum of this status C_{ij} that is LG is less than or greater than or equal to half the sum of neighbors of S_i then the node S_i is said to be normal (GD) otherwise faulty (FT). Then for node S_i who does not have any neighbor whose initial detection status is LG and if the node S_i initial detection is LG, then set S_i to normal (GD) otherwise faulty (FT). The algorithm is made either simple or hard on the a sensor node whose detection status is normal.

3.4.1 Definition of Notations

The table below shows the notation used in this algorithm according to reference [17] with little additional notation

Table 3.1: Description of the notation used

β	Number of neighbors each node should have
N	Total number of sensor nodes in the network
w	Number of neighbors
p	Probability of fault in a sensor
S_i	Set of all sensor nodes
$N(S_i)$	All neighbors of S_i
x_i	Measurement of S_i
d_{ij}^t	Measurement difference between S_i and S_j at time t , $d_{ij}^t = x_i^t - x_j^t$

$\Delta t_i = t_{i+1} - t_i$	Change in the time at different moments
C_{ij}	Test result such that $C_{ij} \in \{0, 1\}$, $C_{ij} = C_{ji}$
θ_1 and θ_2	Predefined threshold values.
T_i	Tendency value of a sensor nodes $T_i \in \{LG, LT, GD, FT\}$
$\Delta d_{ij}^{\Delta t}$	Measurement difference between S_i and S_j from time t_i to t_{i+1} , $\Delta d_{ij}^{\Delta t} = d_{ij}^{t_{i+1}} - d_{ij}^{t_i} = (x_i^{t_{i+1}} - x_j^{t_{i+1}}) - (x_i^{t_i} - x_j^{t_i})$

3.4.1 Algorithm by Peng Jiang

1. For node S_i and any node S_j in $\text{Neighbor}(S_i)$, set C_{ij} as 0 and calculate $d_{ij}^{t_i}$
2. IF $d_{ij}^{t_i} / \Delta t_i > \theta_1$, set C_{ij} as 1 and turn to the next node in $\text{Neighbor}(S_i)$;
3. IF $d_{ij}^{t_i} / \Delta t_i \leq \theta_1$, calculate $\Delta d_{ij}^{\Delta t}$. IF $|\Delta d_{ij}^{\Delta t}| / \Delta t_i > \theta_2$, set C_{ij} as 1 and turn to the next node in $\text{Neighbor}(S_i)$;

Repeat above steps until the test results of each node in $\text{Neighbor}(S_i)$ with S_i are all obtained.

4. IF $\sum_{S_j \in \text{Neighbor}(S_i)} C_{ij} < \text{Num}(\text{Neighbor}(S_i)) / 2$, set initial detection status T_i of S_i as possibly

$S_j \in \text{Neighbor}(S_i)$ and $T = LG$

normal (LG), otherwise T_i is possibly faulty (LT)

5. $\text{Num}(\text{Neighbor}(S_i) | T = LG)$ is the number of neighbor nodes of S_i whose initial detection status is LG. IF $\sum_{S_j \in \text{Neighbor}(S_i)} C_{ij} < \text{Num}(\text{Neighbor}(S_i)) / 2$, set the status of S_i as

$S_j \in \text{Neighbor}(S_i)$ and $T = LG$

normal (GD), otherwise its faulty (FT)

6. If there are no neighbor nodes of S_i whose initial detection status is LG, and if the initial detection status T_i of S_i is LG, then set the status of S_i as normal (GD), otherwise as faulty (FT).

7. Check whether detection of the status of all nodes in network is completed or not. If it has been completed, then exit. Otherwise, repeat steps of (I), (II), (III) and (IV)

3.4.2 Step by Step Improved Algorithm

1. For each sensor node S_i obtain its neighbor (S_i) nodes //this ensure that the neighbor nodes of S_i does not exceed a certain value
2. IF neighbor (S_i) $< \beta$ THEN
3. $T_i = \{LG \text{ or } LF \text{ or } GD \text{ or } FT\}$
4. ELSE IF Neighbor (S_i) $\geq \beta$ THEN

//this steps get the test result C_{ij} of all nodes in the network whose neighbor is $\geq \beta$

5. FOR all nodes S_i and any node S_j in Neighbor(S_i), set $C_{ij} = 0$ and Calculate d_{ij}^t
6. IF $|d_{ij}^t| > \theta_1$, THEN
7. calculate $\Delta d_{ij}^{\Delta t}$
8. IF $|d_{ij}^t| > \theta_2$, set $C_{ij} = 1$ and consider all other nodes in neighbors(S_i)
9. END IF
10. END FOR
11. END ELSE IF
12. END FOR
13. IF $\sum_{S_j \in N(S_i)} C_{ij} < \text{Num}(\text{Neighbor}(S_i))/2$, set initial detection status of S_i is

$S_j \in N(S_i)$

14. $T_i = LG$
15. ELSE
16. $T_i = LT$
17. IF $(\sum_{S_j \in N(S_i)} C_{ij} \leq \text{Num}(\text{Neighbor}(S_i))/2 \text{ or } \sum_{S_j \in N(S_i)} C_{ij} \geq \text{Num}(\text{Neighbor}(S_i))/2)$ set status of S_i

$S_j \in N(S_i) \text{ and } T_j = LG$

$S_j \in N(S_i) \text{ and } T_j = LG$

18. $T_i(S_i) = GD$
19. ELSE
20. $T_i(S_i) = FT$

//the code above is replaced with the code below because of the idea of a multi-hop communication
 Since am dealing with a multi-hop communication in WSNs. Then the remaining undetermined nodes detection status can be achieved as follows:

21. For all nodes S_i whose $\text{neighbor}(S_i)$ is $< \beta$ Do
22. IF atleast one of it $\text{neighbor}(S_i)$ nodes of S_i has initial detection status $T_i = LG$
23. IF initial detection status of S_i is $T_i = LG$ THEN
24. $T_i(S_i) = GD$
25. ELSE
26. $T_i(S_i) = FT$
27. IF the detection status of all the nodes in the network is known. THEN
28. END Program
29. ELSE
30. Repeat all the above steps

3.5 Discussion of Improved Algorithm

The improved algorithm above can correctly identify the status of a sensor node, whose initial detection status is LG and with a small number of neighbor nodes without misdiagnosing it status. This is because we ensured that any node whose neighbor nodes is less than a certain range is not diagonize at first, because this can lead to the node misdiagnoses and that of it neighbor nodes The algorithm is complete and successful if the status of all the sensor nodes in the network are been obtained, without misdiagnosing of any sensor nodes.

The probability of a failure node S_i is p , w is the total number of neighbor of node S_i . The tendency value of every sensor node in the network is either good or faulty but can not be both. Thus giving below explained the probability of each sensor tendency in the network using Binomial distribution for the statistical probability.

The probability of diagnosing faulty node as possibly faulty (LT) is P_{ff} :

$$P(fl_f) = P \sum_{i=0}^{m-1} C_i^{(k)} (1-p)^{k-i} p^i \quad (3)$$

The probability of diagnosing a normal node (GD) possibly Normal (LG) is P_{glg} :

$$P(glg) = (1-P) \sum_{i=0}^{m-1} C_k^{(i)} (1-p)^{k-i} p^i \quad (4)$$

The probability of diagnosing a normal node (GD) possibly to faulty (LT) is :

$$P(glf) = (1-P) \sum_{i=0}^{m-1} C_j^{(k)} (1-p)^{j} p^{k-j} \quad (5)$$

The probability of diagnosing a faulty node (FT) possibly to be normal is:

$$P(flg) = P \sum_{i=0}^{m-1} C_j^{(k)} (1-p)^{j} p^{k-j} \quad (6)$$

CHAPTER FOUR

4.0 Simulation or Proofs of the Proposed Algorithm

This chapter explicitly stated the theorem of the improved proposed algorithm with proofs and

example to show that the distributed algorithm proposed for multichannel communication is correct and can be applied to real life scenario where wireless sensor network is deployed. It also explained the simulation tool that was used in carrying out the implemented algorithm.

4.1 Theorems and proofs of the improved Proposed Distributed Algorithm

Theorem 1:

In the first stage of the our proposed improved distributed algorithm, each sensor node detection status is either $T_i(S_i) = \{LF, LG, GD \text{ or } FT\}$.

Proof:

In the above theorem the following argument is proven to be correct for the first step of our algorithm. In the first step, each sensor nodes S_i obtained all it neighbor nodes $N(S_i)$. If $N(S_i)$ is less than a certain function β , then the detection status of node S_i will be LF, LG, GD or FT which means we can not conclude that the sensor node is either Good or faulty.

These step enable the sensor nodes to calculate the number of neighbors around it and compare it with the given function. If the number of the neighbor nodes $N(S_i)$ does not meet the required function, then it can not be determined to be either Good (GD) or Faulty (FT). This is because fewer number of sensor nodes can lead to incorrect readings of the sensor node itself and may lead other nodes to sensed incorrect reading as well.

Theorem 2:

In the second stage of the improved proposed distributed algorithm, detection status of each sensor node is $T_i(S_i) = \{GD \text{ or } FT\}$

Proof:

For each sensor nodes whose neighbor nodes $N(S_i)$ is greater than or equal to β , then the detection status of node S_i is $T_i(S_i) = GD \text{ or } FT$. In the above theorem if the neighbor nodes of sensor node S_i

is $N(S_i)$ greater than or equal to β , then it initializes its state with respect to its neighbor nodes $N(S_i)$ that is C_{ij} is 0 or C_{ij} is 1. So giving the initial state of the neighbor nodes $N(S_i)$ of sensor node S_i , the sensor node will be able to determine its own possibility based on if the total number $T_i(N(S_i))$ that is Likely to Good less than or greater or equal to the half of the total number of neighbor nodes and if sensor node S_i is itself likely to be good. Then we say that the detection status of sensor node S_i is Good. That is $T_i(S_i) = GD$ otherwise $T_i(S_i) = FT$. If the detection status of all the sensor nodes whose neighbors $N(S_i)$ is greater than β is known then it distributes state or makes the detection status of other sensor nodes whose neighbor is less than β to be easily detected.

Theorem 3:

In the third stage of the proposed improved distributed algorithm, every sensor node whose detection status was originally not detected is detected.

Proof:

For each of the sensor nodes S_i whose neighbor nodes $N(S_i)$ was initially less than the certain function β is determined in order to know their detection status. For the last stage of our algorithm we considered all the sensor nodes whose initial detection status is not determined. If the detection status of the sensor node is LG and any of its neighbor nodes also has detection status of LG then sensor nodes S_i detection status $T_i(S_i) = GD$ otherwise its FT. This ensures that even if a sensor node is good (GD) and none of its neighbors is GD then that sensor node itself is declared to be bad since it can not communicate with any of its one hop count neighbor nodes and can not give room for multi-channel or multi-hop communication.

Theorem 4:

In the last stage of the proposed improved distributed algorithm, the steps in the algorithm is

repeated.

Proof:

In this stage we repeat all the steps of the algorithm to make sure that the detection status of all the sensor nodes is obtained.

Having gone through theorem 1-4 the algorithm is proven but this does not give good proving of an algorithm simulation result may be used to demonstrate this action for better results.

4.2 Analysis of the Fault Tolerance Algorithm

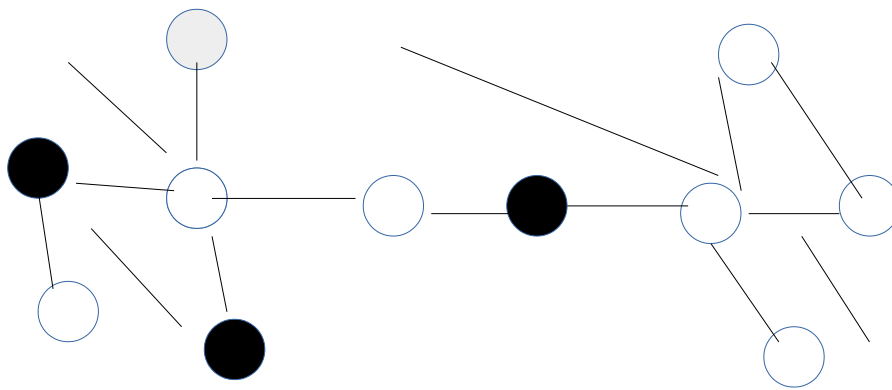


Figure 4.1: example of the network deployment

Table 4.1: showing the result of the example below

S_i with $S_j \geq \beta$	S_j with $C_{ij} = 1$	S_j with $C_{ij} = 0$	0	1	2
0	2, 4	1, 3, 5	LG	GD	GD
1	2	0	-	-	GD
2	-	1, 0, 3	LG	FT	FT
3	2	0	-	-	GD
4	-	0	-	-	FT
5	6	0, 8	LG	GD	GD
6	-	5, 7	-	LT	FT
7	6	8, 9, 10	LG	GD	GD
8	-	5, 7, 9	LG	GD	GD

9	-	7, 8, 10	LG	GD	GD
10	-	7, 9	-	LG	GD

In the above table, out of 11 sensor nodes that was deployed. Three out of the sensor nodes were discovered to be Faulty (FT) and eight of the sensor nodes are Good (GD). The Good sensor nodes can communicate with it one hop neighbor and even beyond as our assumption.

4.3 Simulation With Omnetpp and Castalia

The simulation tool intended to use is Omnetpp 4.2 and Castalia 3.2. OMNeT++ is an object-oriented structured discrete event simulator, which means Objective Modular Network Testbed in C++. The Omnetpp consist of hierarchical nested module as shown in the figure below, where the module can comprises of compound and simple module.

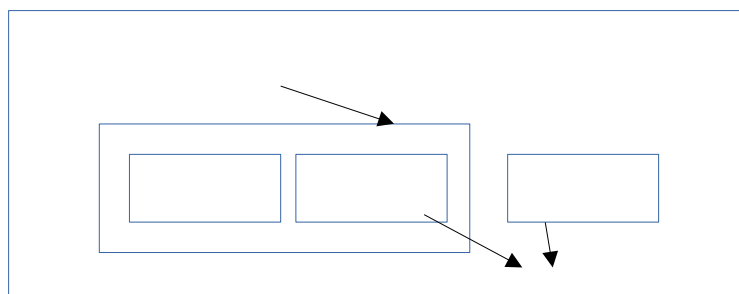


Figure 4.2: Omnet++ Model

Each of the sensor nodes in omnet++ network description file is connected via an arrow which is a bidirectional connection as shown below

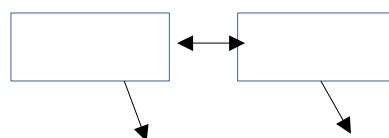


Figure 4.3: node connection

Castalia is a simulator for Wireless Sensor Networks (WSN), which is based on OMNeT++ platform and can be used by researchers and developers who want to test their distributed

algorithms and/or protocols in realistic wireless channel and radio models, with a realistic node behaviour especially relating to access of the radio.

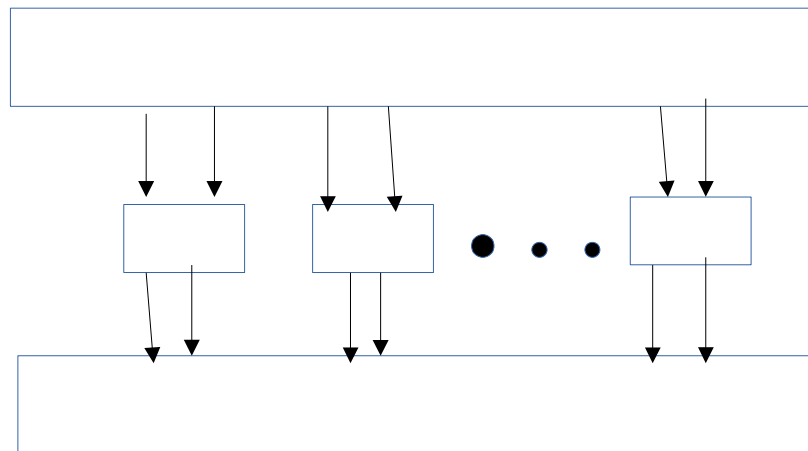


Figure 4.4: Modules and connection process in castalia.

4.4 Simulation Result

the simulation result is better off with the castalia, here we just show what we can achieved with the omnetpp4.2. Showing the interface of the outcome of our coding. Our result shows that the nodes can actually exchange message or acknowledgement with each other as will be shown in the later image. Below is the few sensors we deployed in our network description file (NED). The NED file give us the description of our network topology. This network include a number of component description such as the channels and simple or compound module. The file containing NED file is usually saved as .ned file when coding because this is the convention in Omnetpp4.2.

Figure 4.5: Tkenv of the network topology

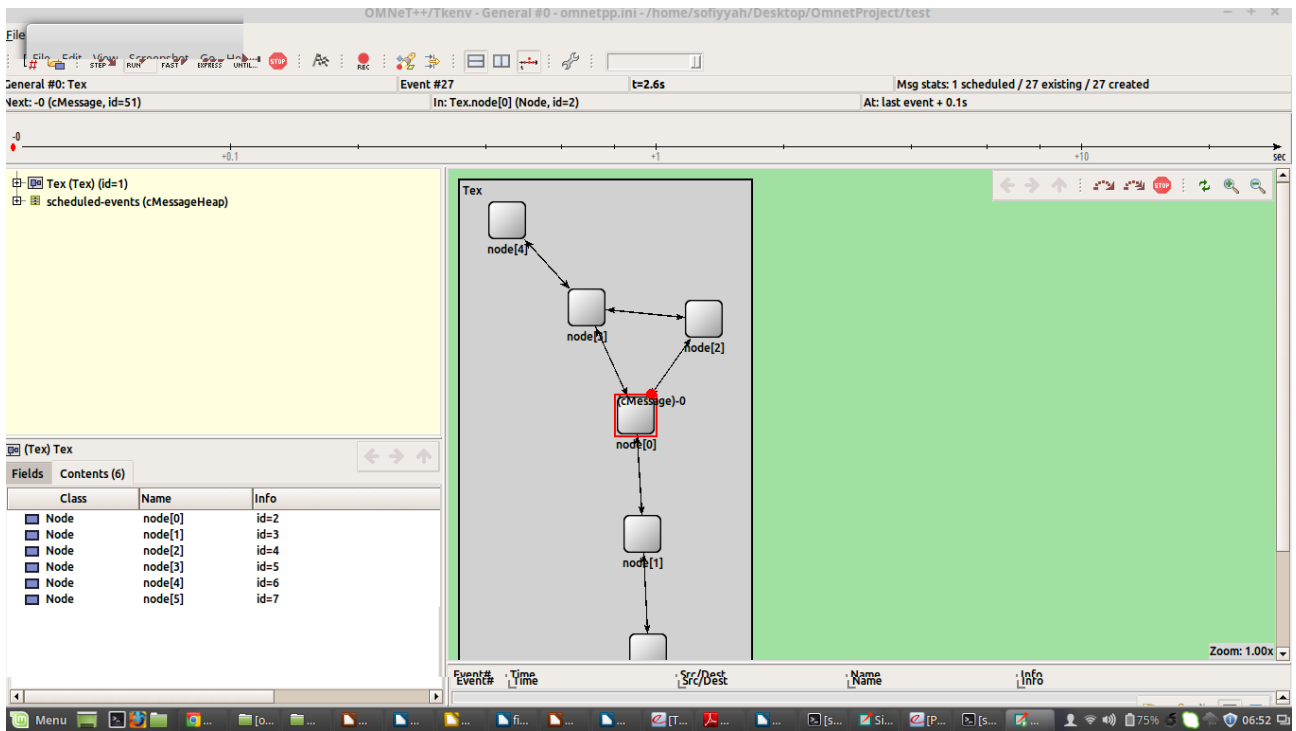


Figure 4.6: Acknowledgement message

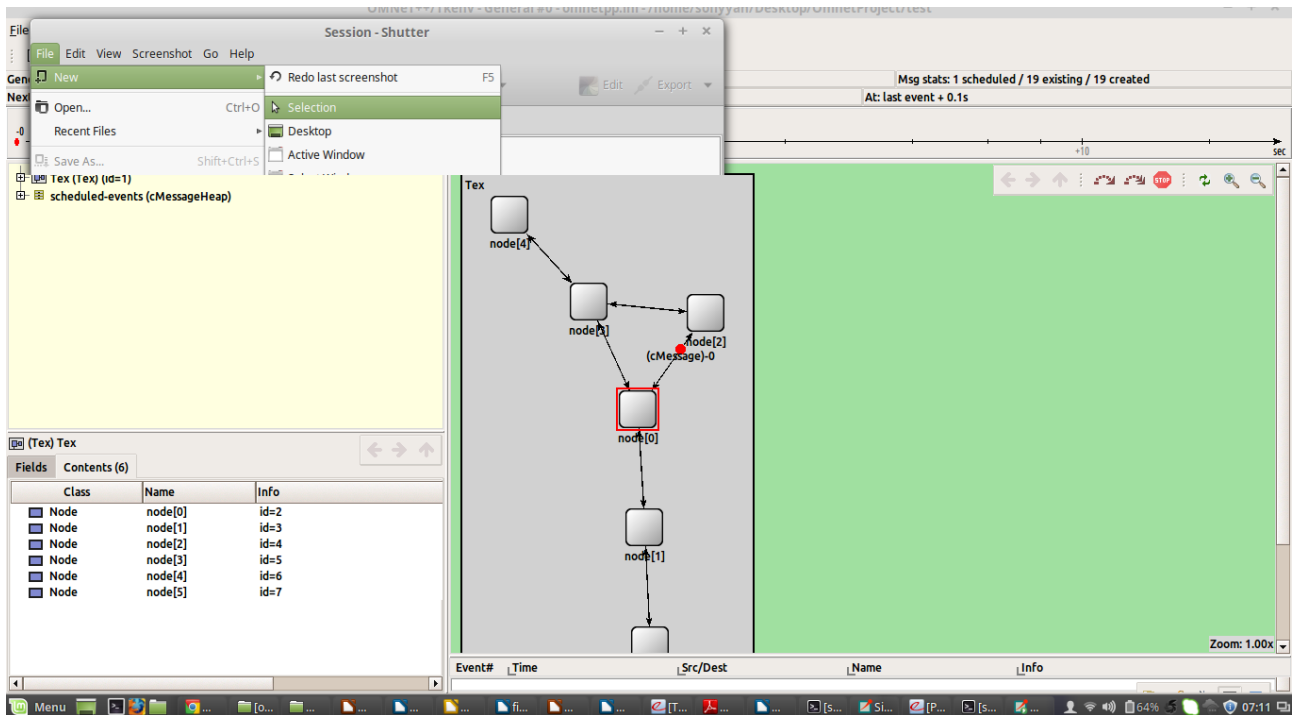


Figure 4.7 : showing movement of acknowledgement

Chapter Five

5.0 RECOMMENDATION AND CONCLUSION

5.1 Resolved Issue

In our improved distributed algorithm we were able to solve the problem of nodes getting wrong detection about it self which in turns leads to wrong detection status about other nodes in it neighbors. We achieved this be enforcing in our proposed improved algorithm by first getting the detection status of nodes with larger neighbors before the ones with fewer neighbors until the detection status of the whole nodes in the network is achieved. We also solve the problem of multi-hop and multi-channel communication, in this we stated clearly in our algorithm that any node whose detection status is good (GD) and has no good neighbor (I.e all neighbors faulty (FT)) is itself faulty (FT). Since it can not communicate it sensed data to it two hops count neighbor and it has no alternative channel to use in passing the details of it sensed data. Finally we were able to state each theorem of our proposed improved algorithm and we proved it.

5.2 Unresolved Issues

The simulation result was not fully achieved because Castalia 3.2 which was suppose to be our bases for simulation in the Omnetpp4.2 environment was unable to be imported to the platform of the Omnetpp4.2. This was the reason for proving the algorithm and part of our algorithm was some what implemented in Omnetpp4.2 with out the Castalia

5.3 Conclusion

In a faulty sensor node distributed detection algorithm where each node detect it own status to be either good or faulty and this claim will be supported by its neighbors because they also check the behaviour of the node themselves The probability of faulty sensor nodes being diagnosed as good and good nodes not been diagnosed as good is very low. The proposed algorithm ensure that sensor nodes with larger neighbors are been diagnosed first before the one with fewer neighbors.

5.4 Challenges

In our improved algorithm the main problem we faced is in the aspect of simulating the algorithm using omnetpp with castalia. Even though we were able to successfully install omnetpp4.2 and castalia 3.2 but we were not able to import castalia in omnetpp as described in the manual. So we were just able to deploy few sensors in the network.

5.5 Future work / Extension of Work

In our future work we want to have a real testbed where our proposed improved algorithm will be tested in real life scenario. We also wish to calculate the detection accuracy for each nodes in wireless sensor network. The detection accuracy is the ratio of the number of faulty sensors detected to the total number of faulty sensors in the whole network.

Reference

1. J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, System Architecture Directions for Networked Sensors, ASPLOS, November 2000.
2. K. R. Chowdhury, N. Nandiraju, P. Chanda, D. P. Agrawal and Q. A. Zing, "Channel allocation and medium access control for wireless sensor networks", Ad Hoc Networks, vol. 7, pp.307 -321, 2009.
3. I. Saleh, M. Eltoweissy, A. Agbaria and H. El-Sayed, "Fault Tolerance Management Framework for Wireless Sensor Networks", Journal of Communications, Vol. 2, No.4, June 2007, pp. 38-48
4. S. Chouikhi, I. El Korbi, Y. Ghamri-Doudane and L. A. Saidane, "A Survey on Fault Tolerance in Small and Large Scale Wireless Sensor Networks" Computer Communications 69 (2015) 22-37

5. D. Thakur, and A. Nayer, "Multichannel Communication- A Need for Wireless Sensor Networks", *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 4, Issue 3, March- 2014, pp. 786-794.
6. X. Han, X. Cao, E. L. Lloyd and C-C. Shen, Fault-Tolerant Relay Node Placement in Heterogeneous Wireless Sensor Networks. Unpublished, Accessed January 2016.
7. M. Zahid Khan, M. Asim and I. Muhammad Khan, "Centralized Schemes of Fault Management in Wireless Sensor Networks. *Computer Science and Telecommunications*, No. 4 (36), 2012. pp. 66-74.
8. Y. Wu, J. Stankovic, T. He and S. Lin, Realistic and efficient multichannel communications in wireless sensor networks. In *INFOCOM 2008, The 27th Conference on Computer Communications. IEEE*. IEEE. pp.1867-1875.
9. I. Saleh, A. Agbaria, and M. Eltoweissy. In-Network Fault Tolerance in Networked Sensor Systems. In *2nd ACM/SIGMOBILE Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks DIWANS06*, California, September 2006
10. M. Panda and P.M. Khilar, "Distributed Byzantine Fault Detection Techniques in Wireless Sensor Networks Based on Hypothesis Testing". Elsevier, *Computer and Electrical Engineering*, 48 (2015), pp. 270-285
11. S. Chouikhi, I. El Korbi, Y. Ghamri-Doudane and L. A. Saidane, "A Survey on Fault Tolerance in Small and Large Scale Wireless Sensor Networks" *Computer Communications* 69 (2015) 22-37
12. P. Chanak, T. Samanta and I. Banerjee, "Fault-Tolerance Multipath Routing Scheme For Energer Efficient Wireless Sensor Networks". *International Journal of Wireless & Mobile Networks* Vol. 5, No. 2, April 2013 pp. 33-45.
13. J. Choi, S. Yim, Y. Jae Huh and Y. Choi, "A Distributed Scheme for Detecting Faults in Wireless Sensor Networks". *WSEAS Transactions on Communications*, Issue 2, Volume 8,

February 2009, pp. 269-278.

14. Hakki Bagci, Ibrahim Korpeoglu, Adnan Yazici, "A Distributed Fault-Tolerant Topology Control Algorithm for Heterogeneous Wireless Sensor Networks", *IEEE Transactions on Parallel & Distributed Systems*, vol.26, no. 4, pp. 914-923, April 2015, doi:10.1109/TPDS.2014.2316142.
15. M. Hezaveh, Z. Shirmohammdi, N. Rohbani and S. G. Miremadi, "A Fault-Tolerant and Energy-Aware Mechanism for Cluster-based Routing Algorithm of WSNs" Unpublished, accessed February 2015.
16. T. Du, S. Qu and Q. Wang, "An Energy Aware Ladder Diffusion Routing Algorithm for WSNs", *Lecture Notes on Information Theory*, Vol. 2, No. 1, March 2014.
17. P. Jiang,; A New Method for Node Fault Detection in Wireless Sensor Networks, *Sensor Nodes* 2009, 9, pp. 1282-1294; doi:10.3390/s90201282. www.mdpi.com/journal/sensor.
18. A. M S and D. Patil, "Implementation of Recovery Algorithm for Faulty Nodes in a Wireless Sensor Network" *International Journal of Research in Engineering and Technology*, 03(03), May 2014. pp. 414-418
19. A. Devi, S. Kumar, and Prerna, Distance Based Fault Detection for Wireless Sensor Network Based on Weighted Average Value. *International Journal of Advanced Technology & Engineering Research*, 1st international Conference on Research in Science, Engineering & Management, 2014 pp. 206-210
20. B. Krishnamachari and S. Iyengar, "Distributed Algorithms for Fault-Tolerance Event Region Detection in Wireless Sensor Networks", *IEEE Transactions on Computers*, Vol. 53, No. 3, March 2004, pp. 241-250.
21. J. Chen, S. Kher, and A. Somani, Distributed Fault Detection of Wireless Sensor Networks, Proc. 2006 workshop on dependability issues in wireless ad hoc networks and sensor networks, pp. 65-72, 2006.

22. Er. Saurabh and R. R. Aggarwal, "A Review of Fault Detection Technique for Wireless Sensor Networks". *IJCSI International Journal of Computer Science*. Vol.10, Issue4, No-1, July 2013.
23. H. Liu, A. Nayak, and I. Stojmenovic, "Fault tolerant algorithms/protocols in wireless sensor networks," in *Handbook of Wireless Ad Hoc and Sensor Networks*. Springer-Verlag London Limited, 2009.
24. F. Koushanfar, M. Potkonjak, and A. Angiovanni-Vincentell, "Fault tolerance techniques for wireless ad hoc sensor networks", *Sensors 2002, Proceedings of IEEE*, pp. 1491-1496, 2002.
25. Manesh and D. Nandal, "Fault Detection in wireless sensor networks", *IPASJ International Journal of Computer Science*, Volume 2, Issue 3, March 2015. pp.6-10.
26. Heena and S. Kapoor, "Survey of Fault Detection Algorithm in WSN", *SSRG International Journal of Computer Science and Engineering*, ISSN 2348-8387, April 2015, pp.78-81.
27. Er. J. Kaur and Er. P. Kaur, A Survey on Fault Detection and Recovery Techniques in Wireless Sensor Networks, *International Journal of Engineering Research and General Science* Volume 3, Issue 2, Part 2, March-April, 2015 ISSN 2091-2730, pp. 638-642.
28. A. S. Arun Raj, K. Ramalakshmi and C. Priyadharsini, "A Survey on Classification of Fault Tolerance Techniques Available in Wireless Sensor Networks", *International Journal of Engineering Research and Technology*, ISSN: 2278-0181, Vol. 3 Issue 1, January – 2014, pp. 688-691.
29. X. Wang, X. Wang, X. Fu, G. Xing, and N. Jha. "Flow-based real-time communication in multi-channel wireless sensor networks." In *Wireless Sensor Networks*, pp. 33-52. Springer Berlin Heidelberg, 2009.
30. J. So, and N. H. Vaidya. "Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver." In *Proceedings of the 5th ACM international*

- symposium on Mobile ad hoc networking and computing*, pp. 222-233. ACM, 2004.
31. M. Ramakrishnan, and P. V. Ranjan. "Multi channel MAC for wireless sensor networks." *International Journal of Computer Networks & Communications (IJCNC)* 1, no. 2 (2009): 47-54.
 32. T. Luo, M. Motani, and V. Srinivasan. "CAM-MAC: A cooperative asynchronous multi channel MAC protocol for ad hoc networks." In *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*, pp. 1-10. IEEE, 2006.
 33. A. Akbari, A. Dana, A. Khademzadeh, and N. Beikmahdavi. "Fault detection and recovery in wireless sensor network using clustering." *International Journal of Wireless & Mobile Networks (IJWMN)* 3, no. 1 (2011): 130-138
 34. A. Sharma and V. Richariya, "Intermittent fault diagnosis on WSNs using an energy efficient clustering Techniques", *International Journal of Emerging Technology and Advanced Engineering*, Vol.1, Issue 2, 2011, ISSN 2250-2459.
 35. M. H. Lee, and Y. H. Choi. "Fault detection of wireless sensor networks." *Computer Communications* 31, no. 14 (2008): 3469-3475.
 36. S. Yuvaraj, and K. R Prasanna Kumar. "Fault Node Detection in Wireless Sensor Networks Based on Round Trip Delay and Paths", *International Journal of Advanced Research Trend in Engineering and Technology*, 2(8), Febuary 2015.
 37. A. Siddiqua, S. Swaroop, P. Krishan and S. Mandal, "Distance Based Fault Detection in Wireless Sensor Network", *International Journal on Computer Science and Engineering*, ISSN: 0975-3397, Vol. 5 No. 05, May 2013.
 38. Castalia, A simulator for Wireless Sensor Networks and Body Area Network Version 3.2 User's Manual by thanassios Boulis, March 2011, NICTA.
 39. OMNeT++, Discrete Event Simulation System Version 2.3b1 User Manual, by András Varga Last updated: February 11, 2003.

