PROJECT REPORT

ON

"MALICIOUS NODE DETECTION IN WIRELESS SENSOR NETWORKS USING WEIGHTED TRUST EVALUATION"

WIRELESS NETWORKING DOMAIN

Submitted in partial fulfilment of the requirements for the award of degree of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM

SUBMITTED BY:

CHANNABASAVA	**1BM10EC023**
MOHAMMED MUDDASSER	**1BM10EC062**
NAVEEN KUMAR K B	**1BM10EC067**
PRAMOD K	**1BM10EC075**

Under the Guidance of,

Mrs. G POORNIMA

Assistant Professor

Department of Electronics and Communication Engineering

B.M.S COLLEGE OF ENGINEERING

(Autonomous College Affiliated to Visvesvaraya Technological University, Belgaum)

Bull Temple Road, Basavanagudi, Bangalore-560019

B.M.S COLLEGE OF ENGINEERING

(Autonomous College under VTU)

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the project entitled **"Malicious Node detection in Wireless Sensor Networks using Weighted Trust Evaluation"** is a bonafide work carried out by **Channabasava (1BM10EC023), Mohammed Muddasser (1BM10EC062), Naveen Kumar K B (1BM10EC067) and Pramod K (1BM10EC075)** in partial fulfillment for the award of Bachelor of Engineering degree by VTU Belgaum, during the academic year 2013-2014.

Internal guide	**HOD**	**Principal**
Mrs. Poornima.G	(Dr. D. Seshachalam)	(Dr. K.Mallikharjuna Babu)
(Associate Professor)		

External Examination **Signature with date**

1.

2.

BMS COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
(Autonomous College under VTU),
Bangalore-560019

DECLARATION

We hereby declare that the project report entitled "**Malicious Node detection in Wireless Sensor Networks using Weighted Trust Evaluation**" is the bonafide record of the project carried out at **B.M.S COLLEGE OF ENGINEERING** in partial fulfillment of the requirement for the award of degree **Bachelor of Engineering** in **Electronics and Communication Engineering, Visvesvaraya Technological University, Belgaum** in the academic year **2013-2014**.

We further declare that the project report is not submitted to any other university in fulfillments of the requirements for the award of any degree.

1. **Channabasava** **1BM10EC023**
2. **Mohammed Muddasser** **1BM10EC063**
3. **Naveen Kumar K.B** **1BM10EC067**
4. **Pramod.K** **1BM10EC075**

ABSTRACT

Wireless Sensor Networks (WSNs) present unique opportunities for a broad spectrum of applications such as industrial automation, situation awareness, tactical surveillance for military applications, environmental monitoring, chemical or biological detection etc., Wireless Sensor Networks (WSNs) consist of hundreds of tiny nodes having the capability of sensing, computation and wireless communications.

Deployed in a hostile environment, individual nodes of a wireless sensor network (WSN) could be easily compromised by the adversary due to the constraints such as limited battery lifetime, memory space and computing capability. It is critical to detect and isolate the compromised nodes in order to avoid being misled by the falsified information injected by the adversary through compromised nodes. However, it is challenging to secure the flat topology networks efficiently because of the poor scalability and high communication overhead. Since security and performance issues are a big concern in the case of wireless sensor networks, emphasis has been given to the scheme based on *Weighted-Trust Evaluation* (WTE). Extensive simulation is performed using MATLAB, to verify performance and efficiency of WTE by varying various parameters.

ACKNOWLEDGEMENT

The joy of accomplishing a task on hand will last a lifetime only if we humbly acknowledge the people who have played a significant part in its fulfillment . Hence, we hereby express our gratitude to all who have contributed to my work.

With profound reverence, we sincerely thank Dr. K Mallikharjuna Babu, Principal, BMSCE Bangalore, and Dr. D Seshachalam, Head, Department of Electronics and Communication Engineering, BMSCE, Bangalore, for giving us an opportunity to undertake this project work at BMSCE, Bangalore.

We would like to express our deepest gratitude to our guide, Mrs. Poornima.G. She introduced us to the field of Wireless Sensor Networks with keen interest and encouragement. We are greatly indebted to her for her valuable advice and moral support and supervision till the completion of the project.

A special word of thanks to all the staff members of Department of Electronics and Communication Engineering, BMSCE, my friends, parents, and well-wishers for their moral support, help with care and concern which helped this work in a big way.

Team Members

TABLE OF CONTENTS

ABSTRACT	**I**
ACKNOWLEDGEMENT	**II**
TABLE OF CONTENTS	**III**
LIST OF FIGURES	**IV**
ACRONYMS	**V**

Chapter	Title	Page no
1	Introduction	**2**
	1.1 Motivation	**3**
	1.2 Problem Statement	**4**
	1.3 Objective	**4**
	1.4 Report Recognition	**5**
2	Background Information In WSN	**6**
	2.1 WSN Model	**7**
	2.2 Architecture of Wireless Sensor Node	**8**
	2.3 WSN Requirements	**10**
	2.4 Characteristics Of WSN	**14**
	2.5 Application Of WSN	**15**
	2.6 Clustering in WSN	**16**
	2.7 WSN Topology	**19**
	2.8 Literature Survey	**20**
3	Network Model	**26**

	3.1 Hierarchical Network Architecture	**26**
	3.2 Grid Based Sensor Network	**27**
	3.3 Data Aggregation and Byzantine Problem	**28**
	3.4 Weighted Trust Evaluation	**29**
	3.5 Extended WTE	**33**
4	Simulation	**42**
	4.1 Weighted trust Evaluation	**43**
	4.2 Extended WTE	**59**
5	Conclusion	**67**
6	Reference	**70**

LIST OF FIGURES

FIGURE NAME	PAGE
2.1.1 Architecture of WSN …….	8
2.2.1 Architecture of Wireless Sensor Node ……..……	9
2.6.1 Clustering in WSN ……..……...…….....	17
2.6.2 Intra and Inter Cluster Communication ……..…….....	18
2.7.1 WSN Topologies ……..…….…….....	20
3.1.2 Hierarchical Network ……..…..	26
3.2.1 Sensor Network with 9 Grids ……...……...	27
3.4.1 A Weighted Based Network for Hierarchical Sensor Network …….	29
3.4.2 Flowchart for Weight Depreciation ……...…….	31
3.4.3 Flowchart for Weight Recovery……...……	33
3.5.1 Localized Event in WSN with Grids ……...……	34
3.5.2 WSN in Localized Event ……	37
3.5.3 WTE in No Event Region ……	40
3.5.4 WTE in an Event Region …….…..	41
4.1.1 Random Deployment of Nodes ……...……	44
4.1.2 Transfer of Data in One Cycle Before the change of CH ……..	45
4.1.3 Selection of New CH ……..	45
4.1.4 Simulation set values are shown …….	46
4.1.5 Initial Deployment of Nodes ……..	47
4.1.6 Change of CHs ……	47
4.1.7 Malicious Node Detected ……	48
4.1.8 Average Response Time, MDR and MR ……	48
4.1.9 Node Setting ……	49
4.1.10 Output ……	49

4.1.11 Deployment of Nodes..... 50

4.1.12 Detected Malicious Nodes and change of CH…… 50

4.1.13 Detected Malicious Nodes…….. 50

4.1.14 MDR vs P_{ma} ……...……. 51

4.1.15 MR vs P_{ma} ……...……..... 52

4.1.16 RT vs P_{ma} ……... 52

4.1.17 MDR vs α……...…… 53

4.1.18 MR vs α……... 54

4.1.19 RT vs α ……. …..... 54

4.1.20 MDR vs β ……...…… 55

4.1.21 MR vs β ……...……...……... 56

4.1.22 RT vs β……...…… …… 57

4.1.23 Nodes set as Malicious in Each Cluster……...…… 58

4.1.24 Deployment of Nodes in Grid Based Clustering…… 58

4.1.25 Deployment of Nodes in Grid Based Clustering after change in CH. 59

4.1.26 Malicious Node Detection……...…… 59

4.2.1 Localized Event Deployment ……... 60

4.2.2 Initial Deployment……...…… 61

4.2.3 Localized Event Occurring at Transmission 3……...…….. 61

4.2.4 End of Localized Event at Transmission 8……...…… 62

4.2.5 No Localized Event at Transmission 9……...……. 62

4.2.6 Detected Malicious Nodes……...……. 63

4.2.7 Initial Set up……...…….. 63

4.2.8 Output ……...……...…… 64

4.2.9 Pma vs MDR ……..... 65

4.2.10 α vs MDR ……. 65

4.2.11 β vs MDR…… ……... 66

ACRONYMS

WSN	Wireless Sensor Network
FN	Forwarding Node
SN	Sensor Node
AP	Access Point
BS	Base Station
WTE	Weighted Trust Evaluation
AODV	Ad-Hoc On demand Distance Vector
QOS	Quality of Service
MAC	Media Access Control
TDMA	Time Division Multiple Access
CSMA	Carrier Sense Multiple Access

CHAPTER 1

INTRODUCTION

Wireless Sensor Networks

Motivation

Objective

Problem Statement

Report Organisation

INTRODUCTION

Wireless Sensor Networks (WSNs) consist of very small devices, called sensor nodes, that are battery powered and are equipped with integrated sensors, a data-processing unit, a small storage memory, and short-range radio communication. Typically, these sensors are randomly deployed in the field. They form an unattended wireless network, collect data from the field, partially aggregate them, and send them to a sink that is responsible for data fusion.. It is critical to detect and isolate the compromised nodes in order to avoid being misled by the falsified information injected by the adversary through compromised nodes. Sensor networks have applications in emergency-response networks, energy management, medical monitoring, logistics and inventory management, and battlefield management.

In contrast to traditional wireless networks, special security and performance issues have to be carefully considered for sensor networks. For example, due to the unattended nature of sensor networks, an attacker could launch various attacks and even compromise sensor devices without being detected. Therefore, a sensor network should be robust against attacks, and if an attack succeeds, its impact should be minimized. In other words, compromising a single sensor node or few sensor nodes should not crash the entire network.

Another concern is about energy efficiency. In a WSN, each sensor node may need to support multiple communication models including unicast, multicast, and broadcast. Therefore, due to the limited battery lifetime, security mechanisms for sensor networks must be energy efficient [1] .Especially, the number of message transmissions and the amount of expensive computation should be as few as possible.

In fact, there are a numbers of attacks an attacker can launch against a wireless sensor network once a certain number of sensor nodes have been compromised [4]. In literature, for instance, HELLO flooding attacks, sink hole attacks, Sybil attack, black hole attack, wormhole attacks, or DDoS attacks are options for an attacker. These attacks lead to anomalies in network behaviors that are detectable in general. There are some reported solutions to detect these attacks by monitoring the anomalies.

In this work, a complex scenario has been addressed. When an adversary has gained control over certain sensor node(s), he/she does not launch direct attacks against the network. Since once the misbehavior is detected, the operator may forsake these compromised nodes and turn to other data sources. Instead, the attacker let those compromised nodes behave normally but report false data to the data collector. The purpose of the adversary is to mislead the operator with falsified data. This may lead to more serious consequences; for instance, in the battlefield a false report regarding the operations of the enemy may lead to extra casualties.

1.1 MOTIVATION:

Wireless sensor networks have recently come into prominence because they hold the potential to revolutionize many segments of our economy and life, from environmental monitoring and conservation, to manufacturing and business asset management, to automation in the transportation and health-care industries. A wireless network consists of thousands of tiny devices which monitor physical or environmental conditions such as temperature, pressure, motion or pollutants etc. at different areas. All nodes in a network communicate with each other and base station via wireless communication. Sensor nodes are usually battery operated devices, and hence energy saving of sensor nodes is a major design issue. To prolong the network's lifetime and to minimize the energy consumption a new technique called clustering [2] is used, where all sensor nodes are divided into some groups called clusters.

In cluster –based routing, special nodes are called cluster heads [3]. Each cluster heads collect data from the sensors belonging to its cluster and forwards it to the sink. So cluster head needs to evaluate the trust of each of its nodes, it is essential to avoid transmission of false data. Hence, the role of cluster head should be changed. The development of clustered sensor networks increases trust, decreases system delay, save energy while performing data aggregation and increase system lifetime.

1.2 PROBLEM STATEMENT:

Just as the wireless sensor networks provide a large number of advantages it also comes with few drawbacks. One among them is security. While the deployment of sensor nodes in an unattended environment makes the networks vulnerable to a variety of potential attacks, the inherent power and memory limitations of sensor nodes makes conventional security solutions unfeasible. In the network if few nodes get compromised the whole network can be toppled unless the compromised nodes are removed from the network soon. In order to do so the compromised nodes should be identified using suitable algorithm and be removed. Sometimes the normal nodes might send the wrong data in case of temporary interruption in communication channel. Suitable algorithm should be used to ensure the proper detection of malicious nodes and avoid misdetection.

1.3 OBJECTIVE:

- The project aims to present a detailed description of weight trust evaluation (WTE) algorithm used to detect the compromised nodes in any wireless sensor network.
- To study the performance of the algorithm used using extensive simulation and analysis.
- To also study various parameters used in the algorithm and find their optimal values.

1.4 REPORT ORGANIZATION:

We have organized the report into 4 chapters which include

CHAPTER 1 describes the Wireless Sensor Network in general in terms of motivation followed by problem statement and objective and finally the whole report outline.

CHAPTER 2 describes the background of WSN and also Literature Survey done for the project.

CHAPTER 3 describes the Network Model. This includes complete details of the algorithm 'Weighted Trust Evaluation (WTE)' being implemented. The Extended version of WTE is also discussed and implementation details are discussed. The various flowcharts for the algorithm are discussed which depict the process flow.

CHAPTER 4 describes the Simulation part of the work done. Extensive Simulation using '**MATLAB'** is done and the performance of 'WTE' and 'Extended WTE' is observed.

CHAPTER 2

BACKGROUND INFORMATION OF WSN

WSN model

Sensor Node Architecture

WSN Requirements

Characteristics of WSN

Application of WSN

Clustering in WSN

WSN Topology

Literature Survey

Advances in wireless networking, micro-fabrication and integration (for example, sensors and actuators manufactured using micro-electromechanical system technology, or MEMS), and embedded microprocessors have enabled a new generation of massive-scale sensor networks suitable for a range of commercial and military applications. . The autonomous devices(nodes) combine with routers and a gateway to create a typical WSN system. The distributed measurement nodes communicate wirelessly to a central gateway, which provides a connection to the wired world. The position of Sensor Nodes need not be engineered or predetermined. This allows random deployment in accessible terrains or disaster relief operations.

Sensor networks are used for wide range of applications like health, military, and home. Especially in military applications the rapid deployment, self-organization, and fault tolerance characteristics of sensor networks make them a very promising sensing technique for control, communications, computing, intelligence, surveillance, reconnaissance and targeting systems. In health system, sensor nodes can also be deployed to monitor and assist disabled patients. Some other commercial applications include managing inventory, monitoring product quality, and monitoring disaster areas.

2.1 WSN MODEL:

The Wireless Sensor Network model was described in below figure.

The architecture has the following components

- Sensor Node
- Cluster Head
- Base Station.

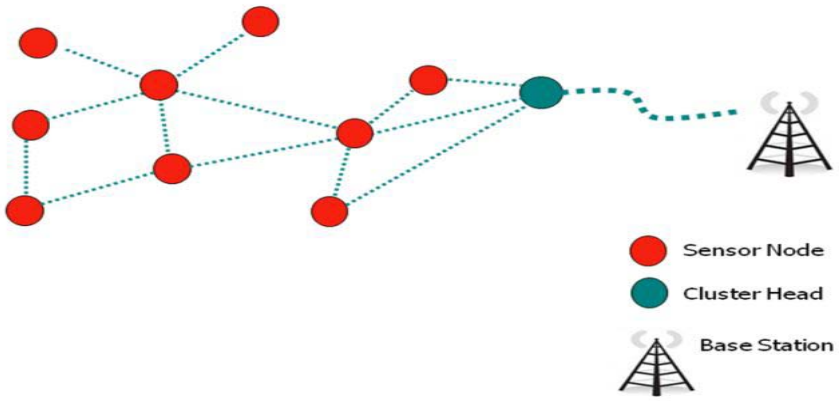


Fig 2.1.1: Architecture of WSN

2.1.1 Sensor Node:

A sensor node is a node in a wireless sensor network that is capable of performing some processing, gathering sensory information and communicating with other connected nodes in the network.

2.1.2 Cluster Head:

Sensor nodes are portioned into clusters and a cluster head is elected using a distributed algorithm. All nodes in the communication range of the cluster head belong to its cluster. Cluster head acts as interface between sensor nodes and base station.

2.1.3 Base Station:

Base station is transceiver station for communication.

2.2 ARCHITECTURE OF A WIRELESS SENSOR NODE:

The basic block diagram of a wireless sensor node is presented in Figure 2. It is made up four basic components: a sensing unit, a processing unit, a transceiver unit and a power unit.

There can be application dependent additional components such as a location finding system, a Power generator and a Mobilizer.

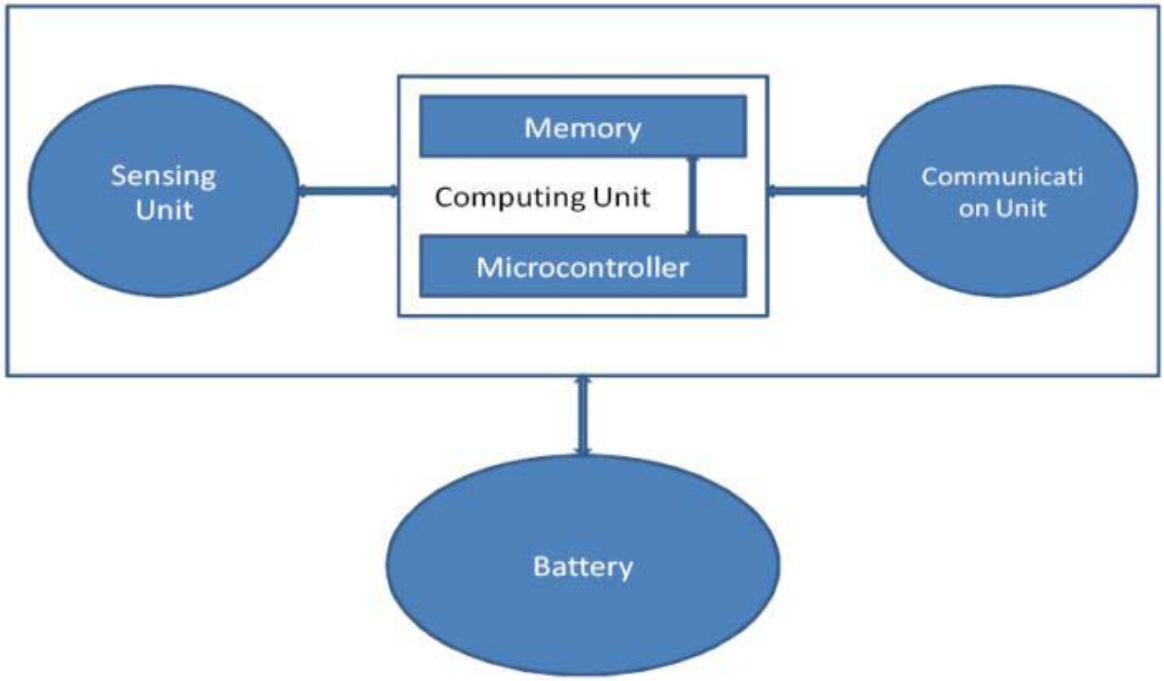


Fig 2.2.1: Architecture of a Wireless Sensor Node

2.2.1 The Sensing Unit:

It consists of the sensor deployed at the node which collects data at the ground level. This data is the physical or the raw data which is sampled and converted to the analog domains and then into the digital form which is then converted into digital forms which is then sent to the processing unit. Sensing units are usually composed of two subunits: sensors and analog to digital converters. Sensor is a device which is used to translate physical phenomena to electrical signals. Sensors devices can be classified as either analog or digital devices.

There exists a variety of sensors that measure environmental parameters such as temperature, light intensity, sound, magnetic fields, image, etc.

2.2.2 The Processing Unit:

The processing unit mainly provides intelligence to the sensor node. The processing unit consists of a microprocessor, which is responsible for control of the sensors, execution of communication protocols and signal processing algorithms on the gathered sensor data. Commonly used microprocessors are Intel's Strong ARM microprocessor, Atmel's AVR microcontroller and Texas Instruments'

MP430microprocessor. In general, four main processor states can be identified in a microprocessor: off, sleep, idle and active. In sleep mode, the CPU and most internal peripherals are turned on, and can only be activated by an external event (interrupt). In idle mode, the CPU is still inactive, but other peripherals are active.

2.2.3 Transmission Unit:

Similar to microcontrollers, transceivers can operate in Transmit, Receive, Idle and Sleep modes. An important observation in the case of most radios is that, operating in Idle mode results in significantly high power consumption, almost equal to the power consumed in the Receive mode. Thus, it is important to completely shut down the radio rather than set it in the idle mode when it is not transmitting or receiving due to the high power consumed. Another influencing factor is that, as the radio's operating mode changes, the transient activity in the radio electronics causes a significant amount of power dissipation. The sleep mode is a very important energy saving feature in WSNs.

2.2.4 Battery: The battery supplies power to the complete sensor node. It plays a vital role in determining sensor node lifetime. The amount of power drawn from the battery should be carefully monitored. Sensor nodes are generally small, light and cheap, the size of the battery is limited. Furthermore, sensors must have a lifetime of months to years, since battery replacement is not an option for networks with thousands of physically embedded nodes. This causes energy consumption to be the most important factor in determining the sensor node lifetime.

2.3 WSN REQUIREMENTS:

A sensor network is composed of a large number of sensor nodes, which are densely deployed. Due to the dense deployment of sensor nodes in remote areas, the successful design of WSNs poses unique challenges and requirements that need to be addressed carefully. Wireless Sensor Node are having limited energy supply, limited computing power, and limited bandwidth of the wireless links connecting sensor nodes. One of the main design goals of WSN is to carry out data communication while trying to prolong the lifetime of the network and prevent connectivity degradation by employing aggressive energy management techniques. In order to increase the reliability of the network, several challenging factors needs to be

addressed meticulously. The main design challenges in WSNs can be categorized into the following areas:

2.3.1 Node deployment:

Node deployment in WSN is application dependent and affects the performance of the routing protocol. The deployment can be either deterministic or randomized. In deterministic deployment, the sensors are manually placed and data is routed through pre-determined paths; but in random node deployment, the sensor nodes are scattered randomly creating an infrastructure in an ad hoc manner. Hence, random deployment raises several issues as coverage, optimal clustering etc. which need to be addressed.

2.3.2 Node/Link Heterogeneity:

Some applications of sensor networks might require a diverse mixture of sensor nodes with different types and capabilities to be deployed. Data from different sensors, can be generated at different rates, network can follow different data reporting models and can be subjected to different quality of service constraints. Such a heterogeneous environment makes routing more complex.

2.3.3 Network Dynamics:

Most of the network architectures assume that sensor nodes are stationary. However, mobility of both BS's and sensor nodes is sometimes necessary in many applications. Routing messages from or to moving nodes is more challenging since route stability becomes an important issue, besides energy, bandwidth etc. Moreover, the sensed phenomenon can be either dynamic or static depending on the application, e.g., it is dynamic in a target detection/tracking application, while it is static in forest monitoring for early fire prevention. Monitoring static events allows the network to work in a reactive mode, simply generating traffic when reporting. Dynamic events in most applications require periodic reporting and consequently generate significant traffic to be routed to the BS.

2.3.4 Scalability:

The number of sensor nodes deployed in the sensing area may be in the order of hundreds or thousands, or more. Any routing scheme must be able to work with this huge number of sensor nodes. In addition, sensor network routing protocols should be scalable

enough to respond to events in the environment. Until an event occurs, most of the sensors can remain in the sleep state, with data from the few remaining sensors providing a coarse quality.

2.3.5 Channel estimation:

Channel estimation plays a critical role in WSNs, since sensor nodes communicate over wireless channels and have to overcome the effects of wireless link, such as noise, multipath effect, intentional jamming and internodes interference. Estimating the wireless link between a specific transmitter and receiver pair provides directionality and reliable data transfer between the nodes. In a multi-hop sensor network, communicating nodes are linked by a wireless medium. The traditional problems associated with a wireless channel (e.g., fading, high error rate) may also affect the operation of the sensor network. As the transmission energy varies directly with the square of distance therefore a multi -hop network is suitable for conserving energy. But a multi-hop network raises several issues regarding topology management and media access control. One approach of MAC design for sensor networks is to use CSMA-CA based protocols of IEEE 802.15.4 that conserve more energy compared to contention based protocols like CSMA (e.g. IEEE 802.11). So, Zigbee which is based upon IEEE 802.15.4 LWPAN technology is introduced to meet the challenges.

2.3.6 Energy consumption:

The nodes in wireless sensor networks have limited energy resources so to extend the lifetime of the entire network, power conservation in individual nodes is of significant importance. In a multi hop WSN, each node plays a dual role as data sender and data router. The malfunctioning of some sensor nodes due to power failure can cause significant topological changes and might require rerouting of packets and reorganization of the network. In WSNs, radio communications is the major consumer of energy. Hence, minimizing the radio transmission power or avoiding the unnecessary communications can considerably save power in sensor nodes.

2.3.7 Connectivity:

The connectivity of WSN depends on the radio coverage. If there continuously exists a multi-hop connection between any two nodes, the network is connected. Connectivity is intermittent if WSN is partitioned occasionally, and sporadic if the nodes are only occasionally in the communication range of other nodes.

2.3.8 Data Aggregation:

Sensor nodes usually generate significant redundant data. So, to reduce the number of transmission, similar packets from multiple nodes can be aggregated. Data aggregation is the combination of data from different sources according to a certain aggregation function, e.g., duplicate suppression, minima, maxima and average. It is incorporated in routing protocols to reduce the amount of data coming from various sources and thus to achieve energy efficiency. But it adds to the complexity and makes the incorporation of security techniques in the protocol nearly impossible.

2.3.9 Data Reporting Model:

Data sensing and reporting in WSNs is dependent on the application and the time criticality of the data reporting. In wireless sensor networks data reporting can be continuous, query-driven or event-driven. The data-delivery model affects the design of network layer, e.g., continuous data reporting generates a huge amount of data therefore, the routing protocol should be aware of data-aggregation.

2.3.10 Coverage:

The coverage of a WSN node means either sensing coverage or communication coverage. Typically with radio communications, the communication coverage is significantly larger than sensing coverage. For applications, the sensing coverage defines how to reliably guarantee that an event can be detected. The coverage of a network is either sparse, if only parts of the area of interest are covered or dense when the area is almost completely covered. In case of a redundant coverage, multiple sensor nodes are in the same area.

2.3.11 Self-organization:

With the large number of sensor nodes deployed in remote environments, the ability of individual sensor nodes to self-organize is vital. Self-organization should be done in a way to improve the performance while reducing the power consumption of the entire sensor network.

2.3.12 Fault Tolerance:

Some sensor nodes may fail or be blocked due to lack of power, physical damage, or environmental interference. The failure of sensor nodes should not affect the overall task of the sensor network. If many nodes fail, MAC and routing protocols

must accommodate formation of new links and routes to the data collection base stations. require actively adjusting transmit powers and signaling rates on the existing links to reduce energy consumption, or rerouting packets through regions of the network where more energy is available. Therefore, multiple levels of redundancy may be needed in a fault-tolerant sensor network.

2.3.13 Quality of Service:

In some applications, data should be delivered within a certain period of time from the moment it is sensed; otherwise the data will be useless. Therefore bounded latency for data delivery is another condition for time-constrained applications. However, in many applications, conservation of energy, which is directly related to network lifetime, is considered relatively more important than the quality of data sent. As the energy gets depleted, the network may be required to reduce the quality of the results in order to reduce the energy dissipation in the nodes and hence lengthen the total network lifetime. Hence, energy-aware routing protocols are required to capture this requirement.

2.4 CHARACTERISTICS OF WSN:

WSNs have some unique characteristics. These are:

Sensor nodes are small-scale devices with volumes approaching a cubic millimeter in the near future. Such small devices are very limited in the amount of energy they can store or harvest from the environment. Nodes are subject to failures due to depleted batteries or, more generally, due to environmental influences. Limited size and energy also typically means restricted resources (CPU performance, memory, wireless communication bandwidth and range).Node mobility, node failures, and environmental obstructions cause a high degree of dynamics in WSN. This includes frequent network topology changes and network partitions. Despite partitions, however, mobile nodes can transport information across partitions by physically moving between them. The resulting paths of information flow might have unbounded delays and are potentially unidirectional. Communication failures are also a typical problem of WSN.

Another issue is heterogeneity. WSN may consist of a large number of rather different nodes in terms of sensors, computing power, and memory. The large

number raises scalability issues on the one hand, but provides a high level of redundancy on the other hand. Also, nodes have to operate unattended, since it is impossible to service a large number of nodes in remote, possibly inaccessible locations.

2.5 APPLICATIONS OF WSN:

Wireless Sensor Networks (WSN) offers a rich, multi-disciplinary area of research, in which a number of tools and concepts can be applied to address a whole diverse set of applications. Sensor networks may consist of many different types of sensors such as magnetic, thermal, visual, seismic, infra-red and radar, which are able to monitor a wide variety of conditions. These sensor nodes can be put for continuous sensing, location sensing,motion sensing and event detection. The idea of micro-sensing and wireless connection of these sensor nodes promises many new application areas. A few examples of their applications are as follows.

2.5.1 Area Monitoring Applications:

Area monitoring is a very common application of WSNs. In area monitoring, the WSN is deployed over a region where some physical activity or phenomenon is to be monitored. When the sensors detect the event being monitored (sound, vibration), the event is reported to the base station, which then takes appropriate action (e.g., send a message on the internet or to a satellite). Similarly, wireless sensor networks can be deployed in security systems to detect motion of the unwanted, traffic control system to detect the presence of high-speed vehicles. Also WSNs finds huge application in military area for battlefield surveillance, monitoring friendly forces, equipment and ammunition, reconnaissance of opposing forces and terrain, targeting and battle damage assessment.

2.5.2 Environmental Applications:

A few environmental applications of sensor networks include forest fire detection, greenhouse monitoring, landslide detection, air pollution detection and flood detection. They can also be used for tracking the movement of insects, birds and small animals, planetary exploration, monitoring conditions that affect crops and livestock and facilitating irrigation.

2.5.3 Structural Monitoring Applications:

Wireless sensors are used to monitor the movement within large buildings and infrastructure such as bridges, flyovers, embankments, tunnels etc.

2.5.4 Other Applications:

Sensor networks now find huge application in our day-to-day appliances like vacuum cleaners, micro-wave ovens, VCRs and refrigerators. Other commercial application includes observation of constructing large size buildings, monitoring product quality, managing inventory, factory instrumentation and many more.

2.6 CLUSTERING IN WSN:

In most wireless sensor network (WSN) applications nowadays the entire network must have the ability to operate unattended in harsh environments in which pure human access and monitoring cannot be easily scheduled or efficiently managed or it's even not feasible at all. Based on this critical expectation, in many significant WSN applications the sensor nodes are often deployed randomly in the area of interest by relatively uncontrolled means (i.e., dropped by a helicopter) and they form a network in an ad hoc manner. Moreover, considering the entire area that has to be covered, the short duration of the battery energy of the sensors and the possibility of having damaged nodes during deployment, large populations of sensors are expected; it's a natural possibility that hundreds or even thousands of sensor nodes will be involved. In addition, sensors in such environments are energy constrained and their batteries usually cannot be recharged. Therefore, it's obvious that specialized energy-aware routing and data gathering protocols offering high scalability should be applied in order that network lifetime is preserved acceptably high in such environments.

Naturally, grouping sensor nodes into clusters has been widely adopted by the research community to satisfy the above scalability objective and generally achieve high energy efficiency and prolong network lifetime in large-scale WSN environment [6]. The corresponding hierarchical routing and data gathering protocols imply cluster-based organization of the sensor nodes in order that data fusion and aggregation are possible, thus leading to significant energy savings. In the hierarchical network structure each cluster has a leader, which is also called the cluster head (CH) and usually performs the special tasks referred above (fusion and aggregation), and several common sensor nodes (SN) as members.

The cluster formation process eventually leads to a two-level hierarchy where the CH nodes form the higher level and the cluster-member nodes form the lower level.

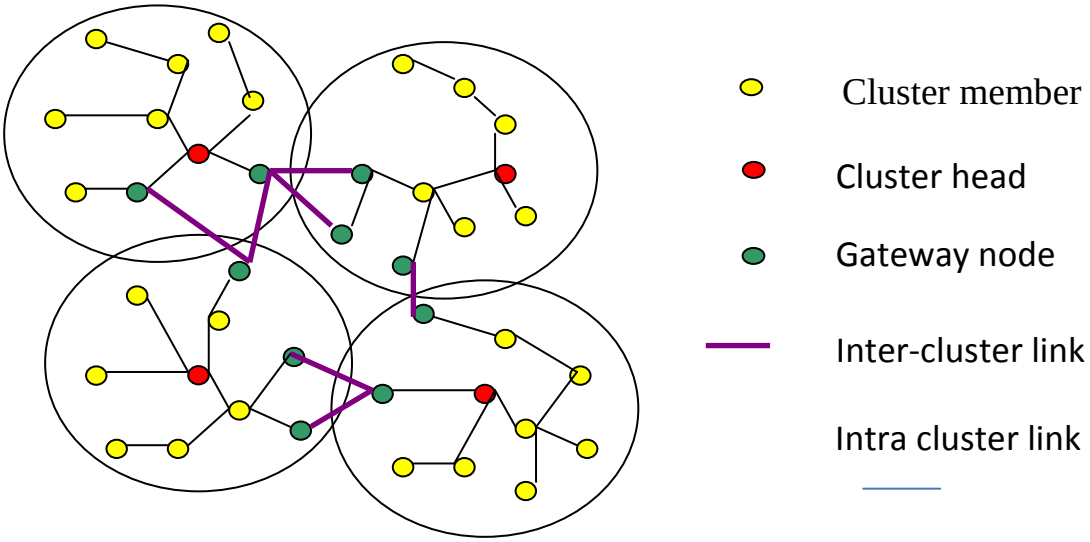


Fig 2.6.1: Clustering in WSN

The sensor nodes periodically transmit their data to the corresponding CH nodes. The CH nodes aggregate the data (thus decreasing the total number of relayed packets) and transmit them to the base station (BS) either directly or through the intermediate communication with other CH nodes. However, because the CH nodes send all the time data to higher distances than the common (member) nodes, they naturally spend energy at higher rates. A common solution in order balance the energy consumption among all the network nodes is to periodically re-elect new CHs (thus rotating the CH role among all the nodes over time) in each cluster. A typical example of the implied hierarchical data communication within a clustered network (assuming single hop intra-cluster communication and multi-hop inter-cluster communication) is further illustrated in Figure 2.6.1.

The BS is the data processing point for the data received from the sensor nodes, and where the data is accessed by the end user. It is generally considered fixed and at a far distance from the sensor nodes. The CH nodes actually act as gateways between the sensor nodes and the BS. The function of each CH, as already mentioned, is to perform common functions for all the nodes in the cluster, like aggregating the data before sending it to the BS. In some way, the CH is the sink for the cluster nodes, and the BS is

the sink for the CHs. Moreover, this structure formed between the sensor nodes, the sink (CH), and the BS can be replicated as many times as it is needed, creating (if desired) multiple layers of the hierarchical WSN (multi-level cluster hierarchy).The approach used in this project is the grid based clustering approach. From a routing perspective, clustering allows to split data transmission into intra-cluster(within a cluster) and inter-cluster (between cluster heads and every cluster head and the sink) communication. This separation leads to significant energy saving since the radio unit is the major energy consumer in a sensor node. In fact, member nodes are only allowed to communicate with their respective cluster head, which is responsible for relaying the data to the sink with possible aggregation and fusion operations. Moreover, this separation allows reduce Routing tables at both member nodes and cluster heads in addition to possible spatial reuse of Communication Bandwidth.

2.6.1 Intra-cluster communications:

All the members of clusters are communicate with that cluster head and other cluster members only. It won't communicate with other cluster members and cluster head. Communication in an Intra –cluster mechanism is done by two hops from member node to base station .During first hop member node communicate with cluster head and then during second hop from cluster head to base station. This phenomena show in figure below.

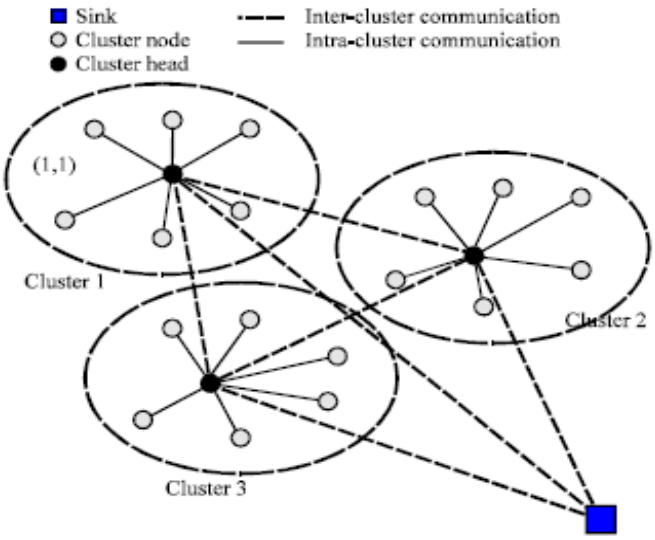


Fig 2.6.2: Intra and Inter cluster communication

2.6.2 Inter-cluster Routing:

In an Inter-cluster routing members of cluster are communicate with cluster head by using either single hop or multiple hop communication then cluster head is communicate with base station via another cluster head [7]. During Inter-cluster routing two or more cluster heads involve. Single hop Inter-cluster communication is easy to communicate sink or base station. Although simple, this approach is not only inefficient in terms of energy consumption, it is based on unrealistic assumption. The sink is usually located far away from the sensing area and is often not directly reachable to all nodes due to signal propagation problems.

2.7 WSN TOPOLOGIES:

Depending on the parameters like load balancing, network scalability several approaches are used for topology control. The clustering types used mainly are:

- Static : local topology control
- Dynamic : changing network parameters
- Single-hop and Multi-hop

The sensor nodes are immobile in case of static WSN topology whereas they change their positions in case of dynamic topology.

In the single-hop models, all sensor nodes transmit their data to the sink node directly. These architectures are infeasible in large-scale areas because transmission cost becomes expensive in terms of energy consumption and in the worst case, the sink node may be unreachable.

In the multi-hop models, we can consider the flat model and the clustering model. In the multi-hop flat model, because all nodes should share the same information such as routing tables, overhead and energy consumption can be increased. On the other hand, in the multi-hop clustering model, sensor nodes can maintain low overhead and energy consumption because particular cluster heads aggregate data and transmit them to the sink node. In the multi-hop clustering model, resources can be allocated orthogonally to each cluster to reduce collisions between clusters and be reused cluster by cluster. As a result, the multi-hop clustering model is appropriate for the sensor network

deployed in remote large-scale areas. The working of each of the types can be briefly understood from the figure below.

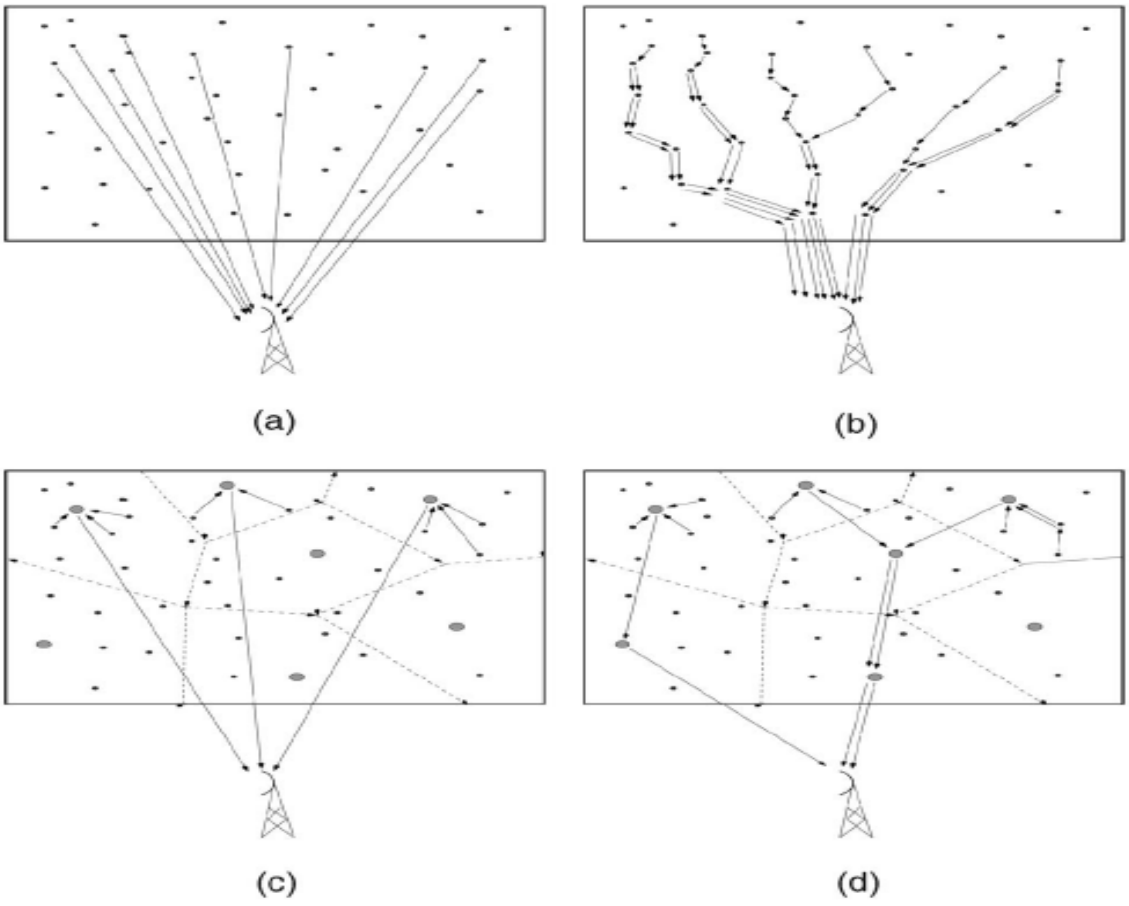


Fig. 1. Sensor information forwarding with and without clustering and aggregation. (a) Single hop without clustering. (b) Multihop without clustering. (c) Single hop with clustering. (d) Multihop with clustering.

Fig 2.7.1: WSN Topologies

2.8 Literature Survey

A literature survey talks about collecting the information required for implementation of any algorithm, behind any successful research work or any good implementation of an algorithm, there will be definitely good amount of time would be spent on the literature survey. It is more about collecting research papers, journals, and documents related to the work which was done before, it really helps in understanding what is being worked and done on the field or domain, and future work supposed to be done. After a very good literature survey one will be having a clear idea of what is the next step to do.

Ganeriwal et al. [1] proposed a reputation-based framework for data integrity in WSNs. The proposed reputation system takes information collected by each node using a *Watchdog* mechanism (for direct monitoring and observations) to detect invalid data and uncooperative nodes.

Yao et al. [2] proposed a parameterized and localized trust management scheme for WSN security, particularly for secure routing, where each node only maintains highly abstracted parameters to evaluate its neighbors.

Aivaloglou and Gritzalis [3] proposed a hybrid trust and reputation management protocol for WSNs by combining *certificate-based* and *behavior-based* trust evaluations. However, [1-3] cited above only considered a node's QoS property in trust evaluation. Also the analysis was conducted based on a flat WSN architecture which is not scalable.

Liu et al. [4] and Moraru et al. [5] proposed trust management protocols and applied them to geographic routing in WSNs. However, no hierarchical trust management was considered for managing clustered WSNs. Their work again evaluated trust based on QoS aspects only such as packet dropping and the degree of cooperativeness, while our work considers both QoS and social trust for trust evaluation of a SN.

Capra et al. [6], [7] discussed the notion of human trust which could be formed from three sources: direct experiences, credentials and recommendations. In particular, recommendations are trust information coming from other nodes in the social context. We consider only two sources in our notion of trust, namely, direct experiences and recommendations, since it is hard for SNs with limited resources to carry credentials. A significant difference of Capra's work from our work is that we specifically consider individual QoS and social trust property, say, X, and devise specific trust aggregation protocols using direct experiences and recommendations to form trust property X, while

Capra used the three sources of information to form human trust. Moreover, because different trust properties have their own intrinsic trust nature and react differently to trust decay over time, we identify the best way for each trust property *X* to take in direct experiences and recommendations information so that the assessment of trust property *X* would be the most accurate against actual status in trust property *X*. Another significant difference is that we consider trust formation as the issue of forming the overall "trust" out of individual social and QoS trust properties, while Capra considered it as the issue of forming human trust out of the three sources of trust information

Shaikh et al. [8] proposed a group-based trust management scheme for clustered WSNs in which each SN performs peer evaluation based on direct observations or recommendations, and each cluster head (CH) evaluates other CHs as well as SNs under its own cluster. This work is similar to ours in that a hierarchical structure is employed for scalability. However, trust in their case is assessed only based on past interaction experiences in message delivery, which in our case is just one possible trust component along with other social and QoS trust components comprising the overall trust metric. Furthermore, we address the trust formation issue (i.e., how a peer-to-peer trust value is formed) to maximize application performance.

Zhang et al. [9] followed the same hierarchical trust architecture and considered multi-attribute trust values instead of just one as in [8]. They also considered a decay function that captures the changing nature of trust in trust calculations. However, their work is theoretical in nature without addressing what trust attributes should be used (a trust composition issue), how trust is aggregated accurately (a trust aggregation issue), or what weights should be put on trust attributes to form trust (a trust formation issue). Intrusion detection is the last defense to cope with malicious nodes for WSNs in which SNs can be compromised due to capture or virus infection

S. Rajasegarar et al. [10], and V. Bhuse et al. [11], proposed a rule-based anomaly detection, typically rules based on QoS metrics are being setup to detect suspected attack behaviors, e.g., if a SN does not forward a packet within a time limit, if a SN forwards the

same packet multiple times without suppression, or if a packet is received directly from a non-neighbor SN or from a neighbor SN who is not supposed to send a packet during a particular time interval, then the SN in question is suspected of maliciousness. When a SN's "maliciousness count" exceeds a tolerance limit, the SN is diagnosed as compromised. The main drawback of rule-based anomaly detection is that it cannot cope with anomalies not covered by rules, thus leading to high false negatives when unknown anomalies appear.

H. Hu et al. [12], proposed weighted trust approach here each SN has a weight associated with it representing the trustworthiness of its sensor reading output. The system periodically calculates the average sensor reading output by taking a weighted summation out of all sensor reading outputs. The weight associated with a SN is dynamically updated according to the deviation of the SN's output from the average output. A larger deviation results in a lower weight. Once the weight of a SN falls below a threshold, the SN is considered a malicious node. The main drawback of this approach is a high false positive probability may result.

In the clustering based approach C. E. Loo et al. [13], SNs reporting similar sensor reading data out of selected data features are clustered together. Consequently, a SN that does not belong to any cluster or belong to a small cluster is considered an outlier or a compromised SN. The effectiveness of this approach hinges on the accuracy of the underlying clustering algorithm achievable only through heavy learning and computation which may impede its use for real time operation.

Theodorakopoulos et al. [14] modeled trust evaluation as a path problem and used path semiring and distance semiring operators to combine opinions such that two nodes can establish an indirect trust relation without previous direct interactions. Here we note that most trust-based intrusion detection mechanisms employed for MANETs cannot be directly implemented in WSNs due to limited battery power and resources in SNs. In this paper, we propose hierarchical trust management leveraging clustering to implement light-weight trust-based intrusion detection for WSNs.

Jun-Won Ho et al. [15] proposed a zone-based node compromise detection and revocation scheme in wireless sensor networks. The main idea behind their scheme is to use sequential hypothesis testing to detect suspect regions in which compromised nodes are likely placed. In these suspect regions, the network operator performs software attestation against sensor nodes, leading to the detection and revocation of the compromised nodes. Through quantitative analysis and simulation experiments, they showed that their scheme detects the compromised nodes with a small number of samples while reducing false positive and negative rates, even if a substantial fraction of the nodes in the zone are compromised.

CHAPTER 3

NETWORK MODEL

Hierarchical Network Architecture

Grid-Based Sensor Network

Data Aggregation and Byzantine Problem

Weighted Trust Evaluation

Extended Weighted Trust Evaluation

Wireless Sensor Networks (WSN) promise researchers a powerful instrument for observing sizable phenomena with fine granularity over long periods. Since the accuracy of data is important to the whole system's performance, it is critical to detect and isolate the compromised nodes in order to avoid being misled by the falsified information injected by the adversary through compromised nodes [8].

Here we use Cluster based Hierarchal routing so that, compromising a single sensor node or few sensor nodes should not crash the entire network, along with the Weighted Trust Evaluation (WTE) scheme to detect malicious nodes.

3.1 HIERARCHICAL NETWORK ARCHITECTURE:

Figure 3.1.1 depicts the network architecture in which the WTE based detection algorithm is implemented. This architecture is based on a three-layer hierarchical topology, consisting of the following three types of nodes:

1. *Sensor Nodes* (SNs): Common nodes in a WSN with limited functionality. These nodes sense and send data to the forwarding node.
2. *Forwarding Nodes* (FNs): More powerful nodes that forward the data obtained from SNs to the upper layer. Also they sense data as the normal SN's.
3. *Access Points* (APs): Nodes that route data between wireless networks and the wired infrastructure.

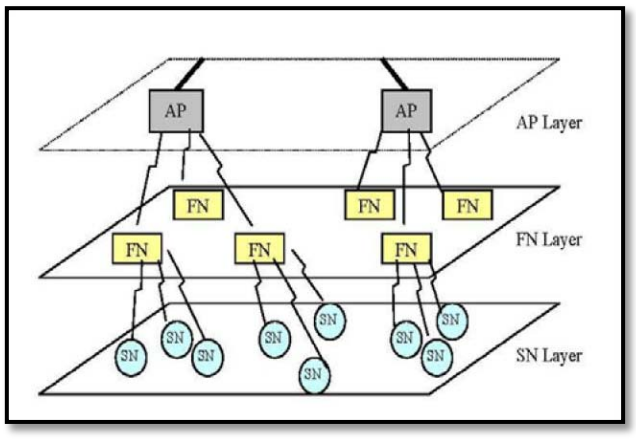


Figure 3.1.1 Architecture of the hierarchical WSN

A number of SNs are organised as a group and are controlled by a higher layer node called 'FN'. SNs here in this hierarchical network are not equipped with

multi-hop routing function to their neighbour nodes. Each SN communicates only with its FN including sending information to and receiving information from FNs. FNs in the middle layer on top of the SN layer offer multi-hop routing function. Each FN has two wireless interfaces: one communicates with lower layer SNs belonging to its management and the other connects to a higher layer node AP.

APs located on the highest layer have wireless and wired interfaces, providing multi-hop routing for all sensor and FNs within the radio range as well as routing data to the wired network. Moreover, APs have the functionality of forwarding control information from the wired network to forwarding and SNs. *Here, we assume FNs are trustful and not compromised by any attack. We also assume APs are trustful, otherwise the adversary can inject any data without been detected.*

3.2 GRID-BASED SENSOR NETWORK:

Grid-based sensor networks have been proposed for energy efficient data aggregation and routing [21]. Our malicious node detection scheme is developed to conform to the protocol of the hierarchical networks. The sensor field in a grid based sensor network is assumed to be divided in M X N square shaped grids as illustrated in *Figure 4.2* , where there are nine grids, A through I, and '*l*' is the side of a grid.

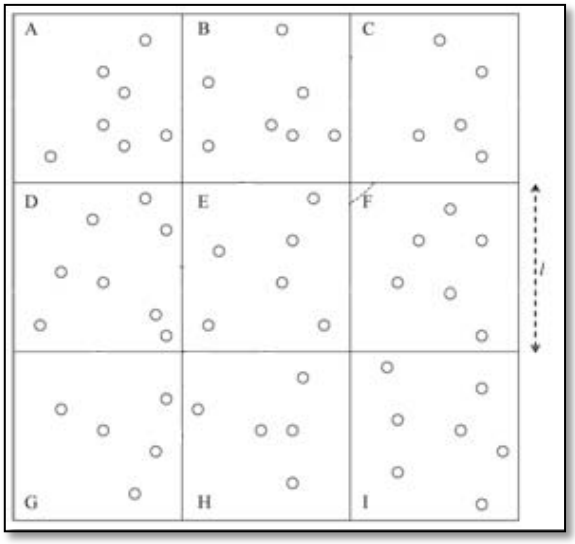


Figure 3.2.1: A sensor network with nine grids

Sensor nodes are assumed to be deployed randomly. Each sensor node is also assumed to know its own location. Immediately after deployment, the sensor network carries out grid construction process, and each sensor node figures out the grid it belongs

to. Sensor nodes in each grid form a cluster, where a cluster head is selected dynamically. All other nodes in the cluster communicate directly with the cluster head.

Two types of communication are defined here for malicious node detection: *one for communication between the cluster head and cluster members and the other for communication between neighboring cluster heads.*

Round Robin Selection of Cluster Heads:

The cluster head is selected dynamically and the selection is based on round robin fashion, where each sensor node (SN) is made the Cluster Head in a circular fashion provided it's not detected as malicious. This type of cluster head selection provides for equitable use of energy in all nodes .Contrary, if a single node is taken as cluster head it loses all of its energy eventually due to continuous functions of aggregation, evaluation and transmission, and hence the round-robin selection improves the lifetime of the node and the network.

3.3 DATA AGGREGATION AND BYZANTINE PROBLEM:

Here the hierarchical network can be considered as a distributed information aggregation network [10]. Based on the information reported by SNs, FNs compute *Aggregation Information* and commit the information to APs. Since SNs may be compromised and report falsified information, it is important for FNs to verify the correctness of the information. Similarly, it is also desired that APs possess the ability of verifying the committed information. Consequently a SN may be compromised or out of function and then provides incorrect information to mislead the whole network. *This problem of detecting incorrect data from malicious node which may lead to detection of falsified information to the BS is called as the Byzantine problem.* For instance, a compromised SN may frequently report incorrect information to higher layers. The aggregator (FN) is thus not able to obtain correct aggregation information due to the effect of this malicious node. Thus, the detection of malicious nodes is an important issue.

3.4 WEIGHTED TRUST EVALUATION:

To overcome this Byzantine problem we implement a Weighted Trust Evaluation Scheme in the above proposed **Grid –Based Hierarchical WSN**.

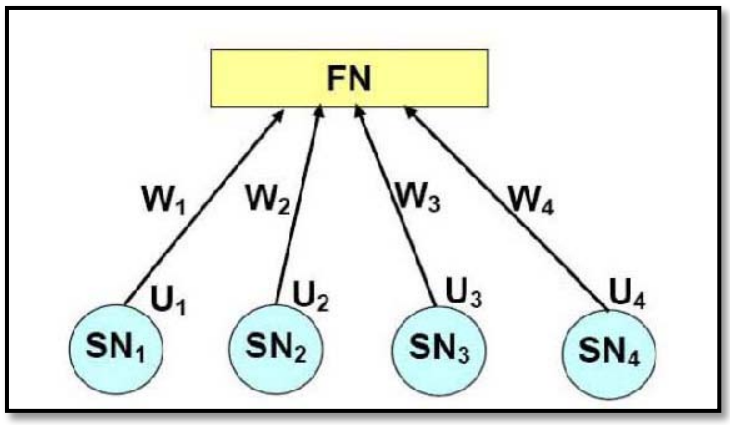


Figure 3.4.1: A weight based network for hierarchical sensor network

At the first step, a weight based network that applies WTE is adapted for a group of SNs and its FN. As shown in Figure 4.3, a weight W_n is assigned to each SN to represent the reliability of the node. In the process aggregating the information sent by the SNs, the FN utilizes the weights and calculates the aggregation result (E) as follows:

$$E = \frac{\sum_{n=0}^{N} W_n \, X \, U_n}{\sum_{n=0}^{N} W_n} \quad \ldots\ldots (1)$$

Where,
E = Aggregation result
W_n = Weight of each SN
U_n = Data sensed and sent by each SN

One concern is about the output U_n definition of SNs. In practice, the output information U_n may be binary information (*e.g.*, 'false' or 'true') or continuous numerical values (*e.g.*, temperature reading). Thus the definition of the output U_n is application dependent.

3.4.1 Malicious Node Modeling:

Here as an example assume that all SN are *sensing temperature of the environment and sending it to the FN's*. The 'E' or aggregation value gives the *weighted average of the temperature* sensed by the nodes. The malicious nodes will send the wrong

or incorrect data so as to confuse the FN to transmit wrong data.

A small acceptable variation 'var' from E (i.e. values of Un between [E + var] and [E – var]) are allowed in the data readings for the normal nodes to account for slight temperature variations in local environment. The normal nodes may too send wrong data i.e. values outside the range [E + var] and [E – var], due to faults in its sensing but these are temporary, and we assume the normal nodes send this incorrect data with the probability of 'Pt'.

We assume that all the sensor nodes become malicious randomly and independently with the same probability P_m. We also assume the malicious nodes send wrong data with probability of 'P_{ma}'. If, for example P_{ma} = 0.4, malicious nodes report 1(0) with a probability of 0.4 when the actual reading is 0(1).

3.4.2 Updating Weights:

Malicious nodes are assumed to arbitrarily modify their readings without being easily detected. To monitor their behavior we define confidence level of a sensor node to represent its reliability in form of WEIGHTS, measuring its past behavior in reporting sensor readings. For a grid with *n* sensor nodes, N_1……N_n, the cluster head maintains w_1…..w_n, as their weights (confidence levels), respectively, where, ($0 < w_i < 1$), and updates them each time a decision on the correctness of their reports is made. Initially all the weights are set to 1. At the time the weight reaches a *predefined lower bound (θ)*, the corresponding node is determined to be malicious and logically isolated thereafter.

If the value(Un) lies outside the range [E + var] and [E – var] it is treated as false value and hence the weight of the node sending this data is reduced by a value 'α', where 'α' is called *weight depreciating factor*.

$$W_n = \begin{cases} W_n - \alpha, & if\,(U_n > [E + var]\ or\ U_n < [E - var]) \\ W_n, & elsewise \end{cases} \quad \ldots\ldots\ldots\ldots (2)$$

The flowchart showing the weight reduction of nodes sending wrong data is shown in figure 4.4. If the data sent by a node (N_n), U_n is outside the acceptable range the nodes

weight is reduced by a factor α. The initial condition of Wn > θ is done so that the procedure is applicable only to nodes which are above threshold θ. Once θ is reached the node is termed as malicious and is removed from network. No data coming from that SN is taken into consideration. The procedure is executed for each SN and for each transmission the weight reduction is performed so that once 'W_n' reaches 'θ', it is declared malicious.

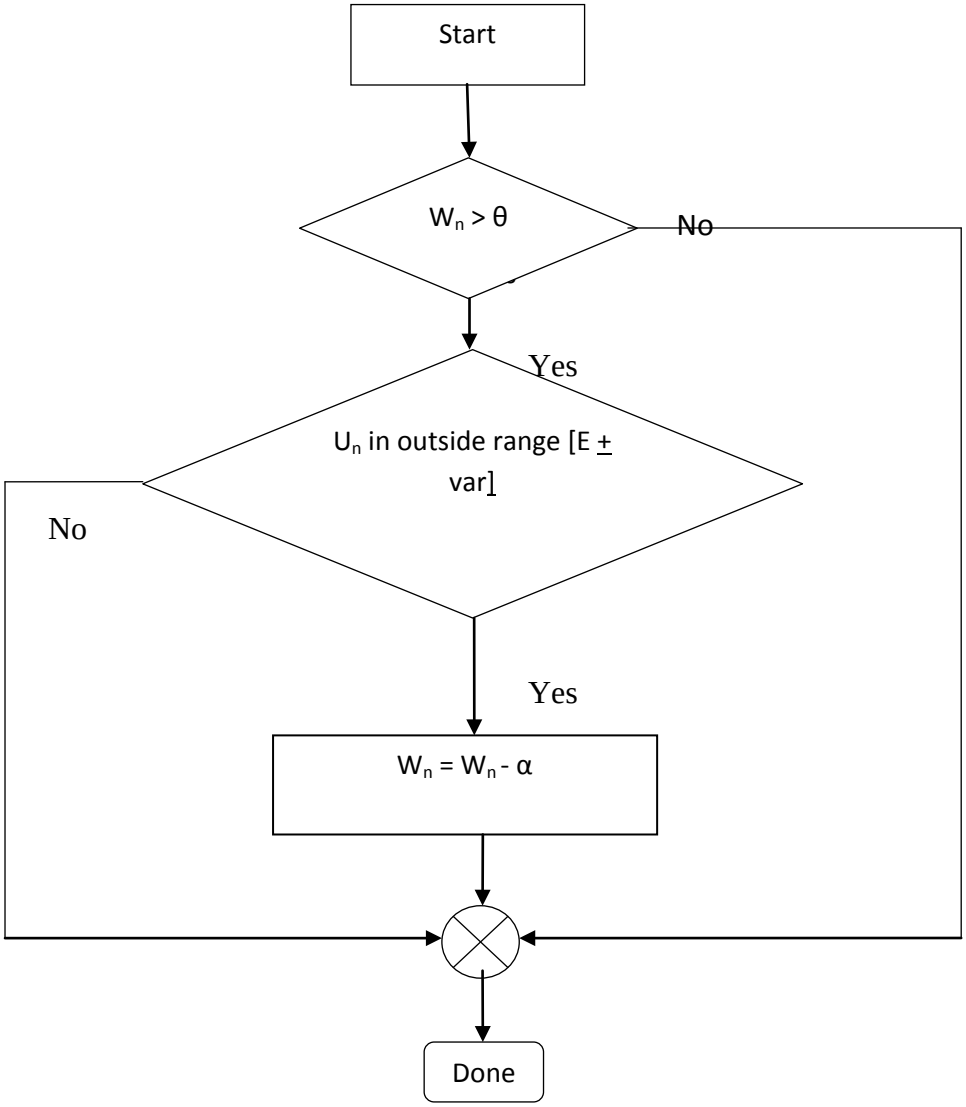


Figure 3.4.2 Flow Chart for weight depreciation

3.4.3 Weight Value Recovery:

As shown in Section 4.2.2, the weight value of a SN is decreased once it is detected reporting incorrect information. However, this incorrect information may be

merely due to a *temporary interruption* in communication channel, the SN is neither compromised nor out-of-function. It is not desired to keep the weight values of such nodes low permanently. Thus, a mechanism is needed to recover weight values of SNs if they work normally after that disturbance. For this purpose, an adaptive weight value recovery algorithm is proposed in this section.

The rationales of this algorithm are considered as follows. If a node has been compromised by the adversary, at least it needs to report falsified information for certain length of time if it intends to mislead the operator of the sensor network. Therefore, whether it is time to recover the weight value depends on the behavior of a node during the past certain period of time.

Only if the SN has been behaving correctly longer than the recovery time t_h, its weight value is increased. In experiments reported in this paper, we assume that each SN reports to the FN periodically, both t_h and t_c are defined in number of period cycles. Once a node's weight value become lower than θ, then no weight recovery is done since the node is detected malicious. If the node has worked normally long enough, its weight value is recovered according to Equation (3).

$$W_n = \begin{cases} W_n + \beta, & if(U_n = E \ and \ t_c \geq t_n) \\ W_n, & elsewise \end{cases} \quad \ldots\ldots\ldots\ldots (3)$$

Where:

β = a weight recovery rate

t_C = the time in which the SN behaves correctly

t_H = (preset threshold) the required length of time for weight value recovery, named

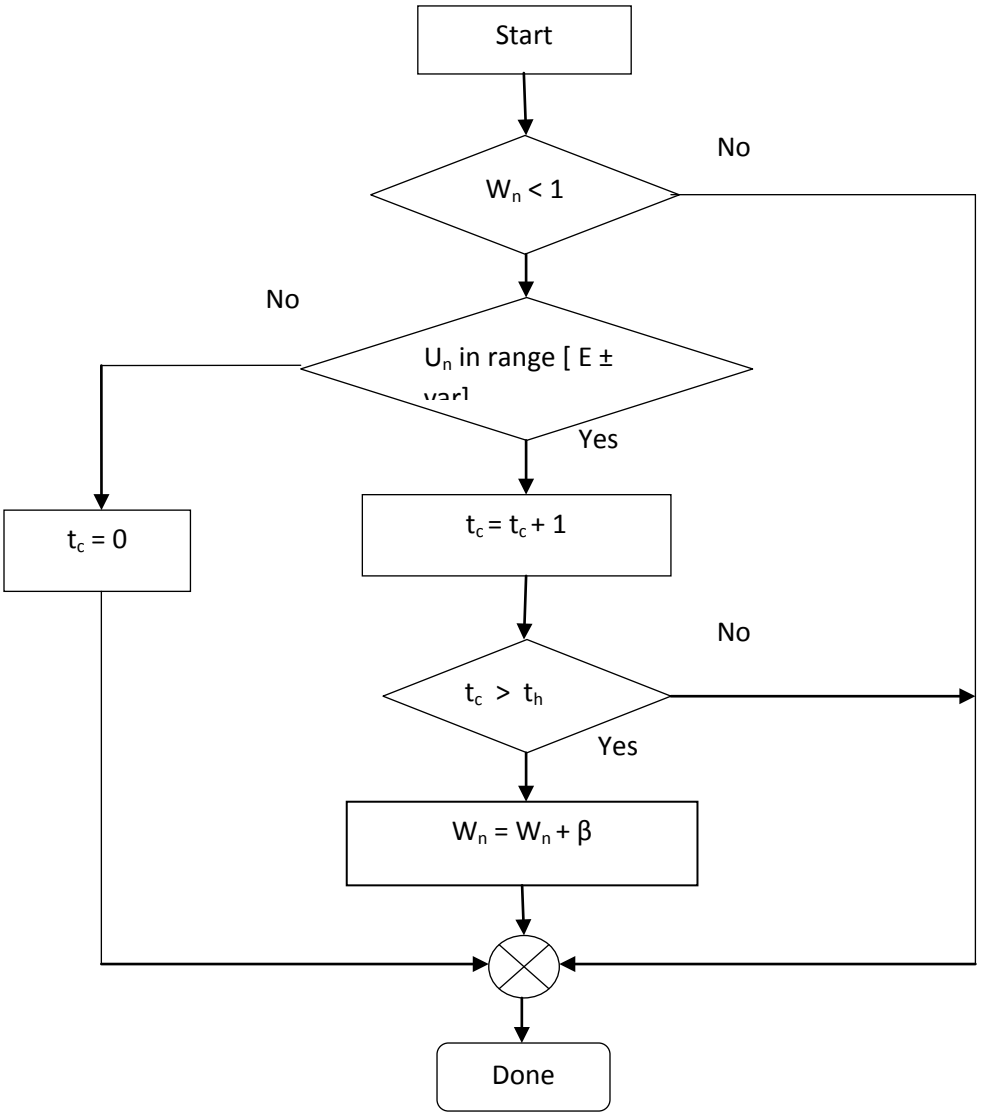


Figure 3.4.3 : Flow Chart for weight recovery

So that after recovery the weight does not go beyond 1(the max limit for weight of the node), min (1, W_n) is taken as the updated weight. The flowchart of the weight recovery algorithm is shown in the figure 4.5. Here the initial condition $W_n < 1$ is done so that only the weight of normal nodes is recovered.

3.5 EXTENDED WEIGHTED TRUST EVALUATION:

The WTE discussed until now works well for situations like temperature detection in an environment since here, the parameter sensed (temperature) varies throughout the area evenly and all sensor nodes need to send the same value with slight local variations. What if in a scenario there is a localized event between multiple grids as shown in the figure 4.6 below?

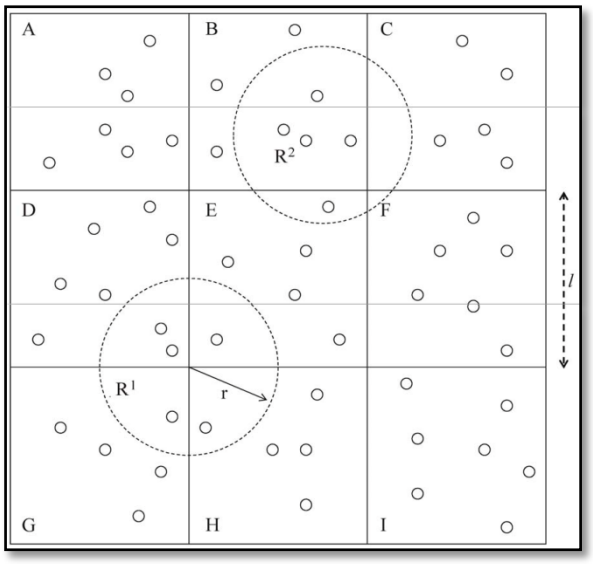


Figure 3.5.1 localized events in WSN with 9 grids

For example consider a scenario where temperature is initially uniform throughout the grid as explained in normal WTE. Now event regions such R^1, shown in figure 4.6 are formed, where assume temperature changes locally due to some activity, the rest of the region now is called as no event region. For clarity, we name acceptable sensor data in case of no event as "normal" readings. Any readings outside the normal range are called "unusual" readings for convenience. In other words, in a normal WTE application correct sensor readings sent by normal nodes due to localized event, in the local event region are also sensed as unusual readings since their value does not match with the aggregation result. Hence a binary decision is made for each sensor reading, where a "1" indicates an unusual reading. Sensor nodes in an event region are expected to report a 1, unless the nodes are faulty. Hence we need to provide some modification to the WTE so that these kind of situations are dealt in appropriate manner.

Also, we assume that malicious nodes can change the sensor readings arbitrarily. In addition, they have some intelligence to report 0's and 1's alternately, to break down the network while remaining undetected, unless some sophisticated techniques are used to detect them.

No event Region: In this case SN detect and send a value in range of aggregation result so that all nodes sending correct data and are represented as binary '0'. The cluster head computes and finds if the sensed node value is within aggregation result and is represents the sensed data of node as binary 0.(in the cluster head).

Event Region: Now the normal SN's in this region send different data since there is an activity going on in the region where in this event region a different temperature is sensed by the nodes. But this when compared to aggregation result is out of range and is represented as binary '1'.

Problem - Now in these event/no event situations malicious nodes keep sending opposite or wrong data in any of these regions with probability 'P_{ma}' as discussed before. Hence detection of these malicious nodes becomes a very complex task. We discuss and implement an extended WTE to detect malicious nodes in these environments

Example considered for Simulation:

For simplicity of simulation we consider another example such as '*Border Monitoring Area'* where the field or region is filled with IR sensors to detect any human presence. Now here event region is region where the human is actually sensed which occurs locally in regions like R^1. Now the nodes in event region send '1' directly indicating alarm. The other nodes send no alarm i.e. '0'. Since the localized events lie across multiple grids then the decision made at a cluster head alone based on the sensor readings of its member nodes might not be accurate since the cluster head is confused as to whether a event has occurred or not. Here we use a '*Threshold Test*' to identify if event region occurs. And a *suitable WTE is applied in 'event' and 'no event' region* which will be discussed in the next sections.

No event Region: In this case SN detect no alarms in the area all normal SN's send '0' to the FN.
Event Region: Now the normal SN's send '1', i.e. alarm in the region, to the FN's.

This example is considered and used for generating a simulation model, and the performance of WTE is seen.

3.5.1 Threshold Test:

Let's consider each sensor node 'v_j' send a data s_j such that $s_j = 0$ or 1 based on no event/event condition. We need to find out if there is an event or not in a grid, to do this we perform the **threshold test** using two variables M_0 and M_1 as shown below:

ALGORITHM FOR THRESHOLD DETECTION

1. Each sensor node v_j sends a 1 (alarm) to the cluster head if $s_j = 1$.

2. Each cluster head computes $M_0 = \sum_{j=1}^{d} w_j (1 - s_j)$ and

$$M_1 = \sum_{j=1}^{d} w_j s_j .$$

3. If $\frac{M_1}{M_1 + M_0} > \theta_1$, then $E = 1$(*i.e.*, an event) and update confidence levels accordingly

If $\frac{M_1}{M_1 + M_0} \le \theta_1$ and $\frac{M_1}{M_1 + M_0} > \theta_2 \left(\theta_2 \le \theta_1\right)$, then estimate the center of alarms using inter-grid communication, and apply weighted majority voting to the estimated event region. If $E = 1$, update confidence levels accordingly

If $\frac{M_1}{M_1 + M_0} \le \theta_2$, then $E = 0$ (*i.e.*, no-event) and update confidence levels accordingly.

4. Determine the nodes with $w_j = 0$ to be malicious.

As shown above we have 3 cases:

Case 1: **Event in the grid** (called **Majority Voting**) – occurs when majority (depending on θ_1) number of nodes sends an event or alarm signal i.e. when:

$$\frac{M_1}{M_1 + M_0} > \theta_1$$

Case 2: **Localized Event** may occur, and now conformation from neighboring cluster head is needed, when:

$$\frac{M_1}{M_1 + M_0} \le \theta_1 \quad \text{and} \quad \frac{M_1}{M_1 + M_0} > \theta_2 \left(\theta_2 \le \theta_1\right)$$

Case 3: **No event** in the grid, when:

$$\frac{M_1}{M_1 + M_0} \le \theta_2 ,$$

Where,

θ1 - Upper threshold : Ex. if θ1 = 0.6, then if more than 60% of the nodes send '1' i.e. the presence of an event then WTE is performed throughout the grid assuming event occurring in full grid.

θ2 - Lower threshold : Ex. if θ2 = 0.2, then if more than 20% and less than 60% of the nodes in a grid send '1' we need to find the event region which may lie between multiple grids and apply WTE in that event region only i.e. a localized event has occurred.

Note – The percentages shown above are weighted percentages where node weight is considered in evaluating, i.e. the weights or confidence levels of each node will be taken into account.

3.5.2 Determining the Event Region in case of Localized Event:

In order to cope with the expected poor performance, we estimate the event region, if necessary, with **inter-grid communication**, by finding the center of the nodes reporting an alarm, and then apply WTE to the estimated event region. To estimate this event region we use the following procedure across the grids.

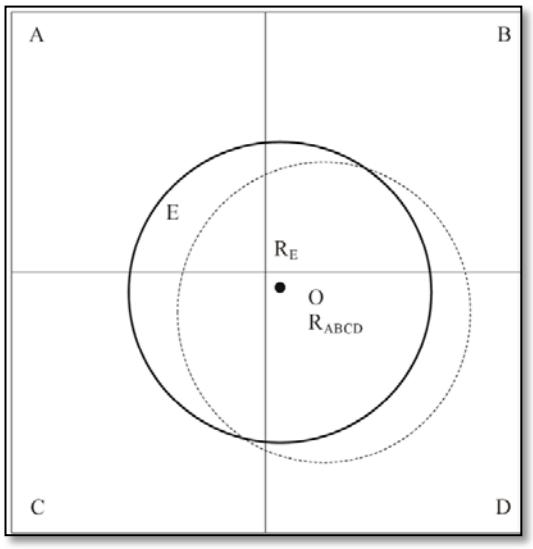


Figure 3.5.2 WSN with localized events

In Figure 3.5.2**,** an event region *E* is located across four grids, A, B, C, and D, such that each grid has insufficient number of event nodes to pass majority voting. The event can possibly be detected if the threshold is lowered. It, however, causes significant false alarms, especially for a fault-prone sensor network with a large number of malicious nodes. The center of the alarms in the grid A, R_A, is defined here as the weighted average of the positions, r_i's, of the nodes reporting a '1' (*i.e.*, an alarm), and it thus can be expressed as

$$R_A = \frac{\sum w_i r_i}{\sum w_i} \qquad \text{…….. (6)}$$

Where, each alarm node in the grid A contributes to the estimated center as long as its weight is not zero.

Similarly, the overall center of the alarms from the four grids, R_{ABCD}, can be written as

$$R_{ABCD} = \frac{w_A R_A + w_B R_B + w_C R_C + w_D R_D}{w_A + w_B + w_C + w_D} \quad \text{…….. (5)}$$

Where, 'w_A' represents 'Σw_i' of the sensor nodes reporting a '1' in the grid A. Once the center is computed, '*Weighted Trust Evaluation*' is performed accordingly, within the circle centered at with radius *r'*, such that r'<r.

Note:

- As the event region increases, however, at least one grid may have sufficient number of event nodes to pass a threshold test, such as the well- known majority voting.
- In order for localized event to occur all surrounding grids must have a localized event so that area under the circle of R_{ABCD} is equal to area of each grid, else no event region is detected. It is assumed that when activity occurs minimum area affected is equal to the grid area is affected. In such a scenario the algorithm works optimally. For simplicity we consider a single local event occurring in between 4 grids and the event region is obtained for simulation purpose.

3.5.2 Updating Weights WTE:

In the proposed detection, the decision on an event is made at the cluster head based on threshold test as discussed before. Now the event region is found and the following steps are followed for weight reduction and recovery.

In **NO-EVENT** region, the cluster head updates the weights as follows. Malicious nodes reporting a 1 i.e. $s_j = 1$ in the case of no-event lose their weights, w_j by α. The flowchart shown in figure 4.8 represents the algorithm for the same. Otherwise; they gain weights by β if the node has sent correct data for periods more than 't_n' as shown by equation 6 and 7.

$$w_j = w_j - \propto \quad for\ s_j = 1 \qquad \text{.....(6)}$$

If $\mathbf{t_c \geq t_n}$, then;

$$w_j = \min(1, w_j + \beta) \quad for\ s_j = 0 \qquad \text{..... (7)}$$

Flowchart for WTE in NO - EVENT Region

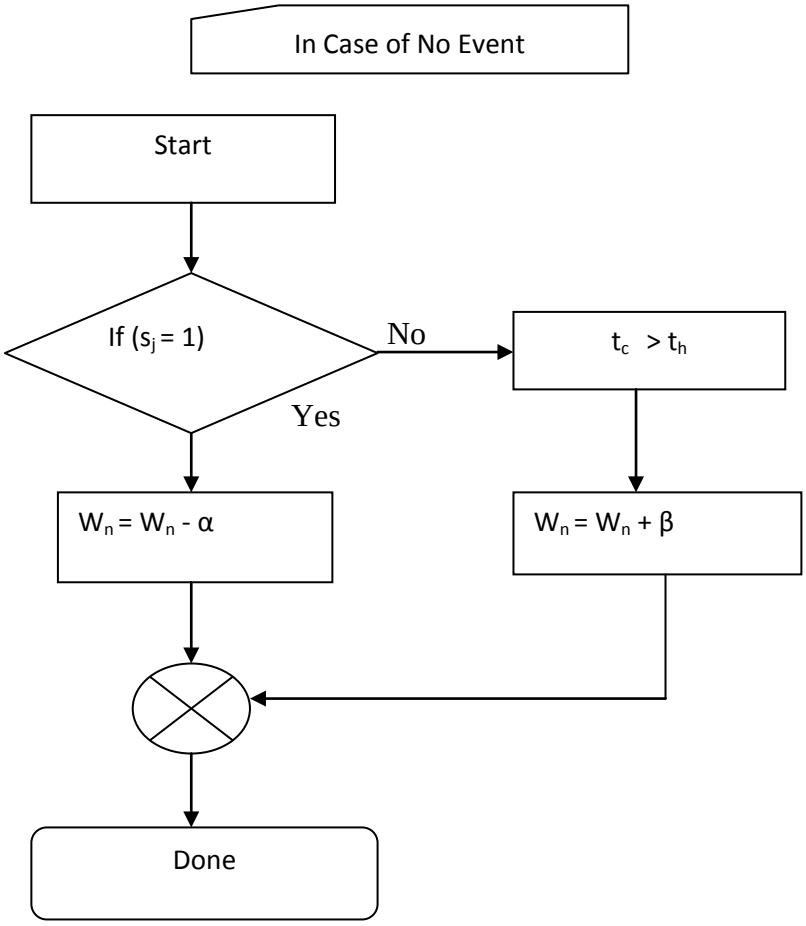


Figure 3.5.3: WTE in no event region

The flowchart represents the steps for weight reduction and recovery in event region for WTE.

The two parameters, α and β play an important role in distinguishing between malicious and normal nodes. If α = 0.2, β = 0.05 and t_n = 2, for example, a sensor node reporting a 1 every ten cycles recovers its weight to 1.0. That is, for the chosen values of α, β and t_n and a normal sensor node with some transient faults remain in the network unless the probability 'P_t' is greater than 0.2. Malicious nodes reporting alarms more frequently than this gradually lose their weights, and will eventually be detected at the time the weights reach 0.

In the **case of an EVENT REGION**, the weights of the nodes within the event region need to be lowered if they have reported a '0' i.e. $s_j = 0$. Due to the inaccuracy of the center estimation, however, we apply the updates only to sensor nodes within a circle of radius r', centered at the estimated center, not to sacrifice normal nodes. The following up- dates are done at the cluster head.

$$w_j = w_j - \propto \qquad for\ s_j = 0 \qquad \text{..... (8)}$$

If $\mathbf{t_c \geq t_n}$, then;

$$w_j = \min(1, w_j + \beta) \qquad for\ s_j = 1 \qquad \text{..... (9)}$$

Note: The weight recovery occurs only if $t_c > t_n$ for both event and no event region.

Flowchart for WTE in EVENT Region :

The flowchart represents the steps for weight reduction and recovery in the event region for WTE in figure 4.9

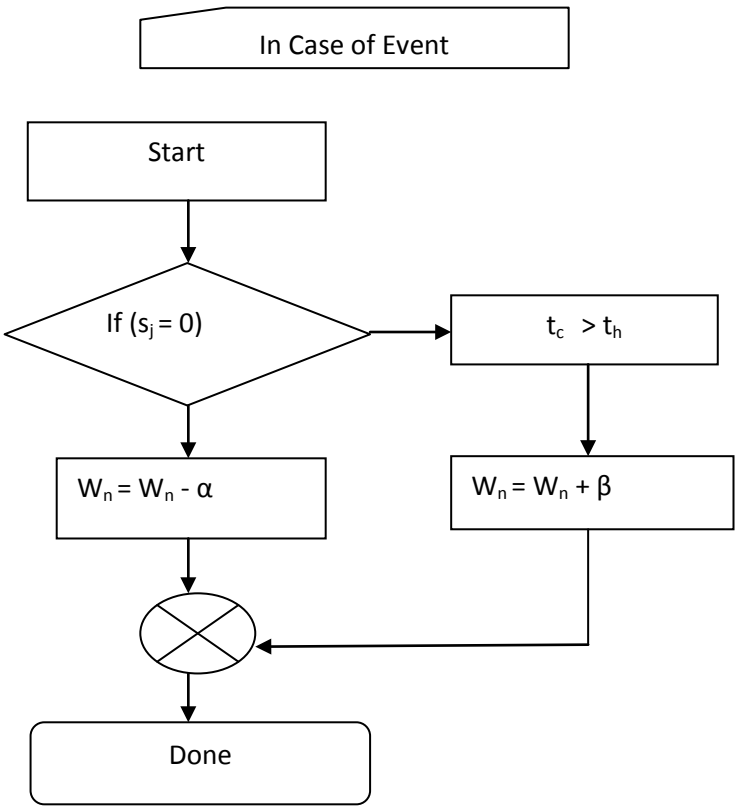


Figure 3.5.4: WTE when in an event region

Limitations of Algorithm in Localized Sensing:

- Cases where there is slight interference with event occurring in adjacent grid, such that a no event scenario occurs in present grid may lead to misdetection of nodes.
- Centre estimation may not be accurate and may pose problems in the grid.
- The value of θ2 can be reduced but there is a possible chance of false alarms. Hence we assume the event minimally affects an area size is equal to one grid area.

CHAPTER 4
SIMULATION

Weighted Trust Evaluation

Extended Weighted Trust Evaluation

Computer simulation is performed in a sensor network using MATLAB, where sensor nodes are randomly deployed in a square area. The network area is divided into grids of the same size, each of which has 20 nodes on average. Three metrics, malicious node detection rate (MDR), misdetection rate (MR) and Response Time are used in the performance evaluation.

MDR is Malicious node Detection ratio defined to be the ratio between the number of detected malicious nodes and the total number of malicious nodes in the network set at the beginning of simulation.

$$MDR = \frac{No.of\ detetced\ malicious\ nodes}{Total\ number\ of\ malicious\ nodes}$$

(Should be ideally 1, so that all malicious nodes are detected.)

MR is misdetection ratio which is the ratio of misdetected nodes to all detected nodes including correctly detected and misdetected nodes. Note that these misdetected nodes actually consist of two categories: normal nodes being treated as malicious ones and malicious node being treated as normal nodes.

$$MR = \frac{No\ of\ misdeteted\ nodes(Malicious\ but\ not\ detected + normal\ detected\ as\ malicious)}{Total\ number\ of\ detected\ nodes}$$

(Ideally should be 1, but also all nodes detected should be malicious only.)

RT is average Response time defined as average number of cycles required to detect the malicious nodes.

4.1 WEIGHTED TRUST EVALUATION:

Here initially a single cluster is taken consisting of 15 SN's and the WTE is implemented in it. The basic WTE for a temperature sensor network where all nodes are sending temperature outputs is simulated. An area of 50 X 50 is taken and random deployment of nodes is done as shown in the figure 5.1.

Here α = 0.2, β=0.1, t_n = 2, $\theta 1$= 0.6, $\theta 2$ =0.2, P_t = 0.05. The fig 4.1.1 shows the sensed data and the updated weights after each transmission. Here initially a constant fixed cluster head is chosen to transmit data.

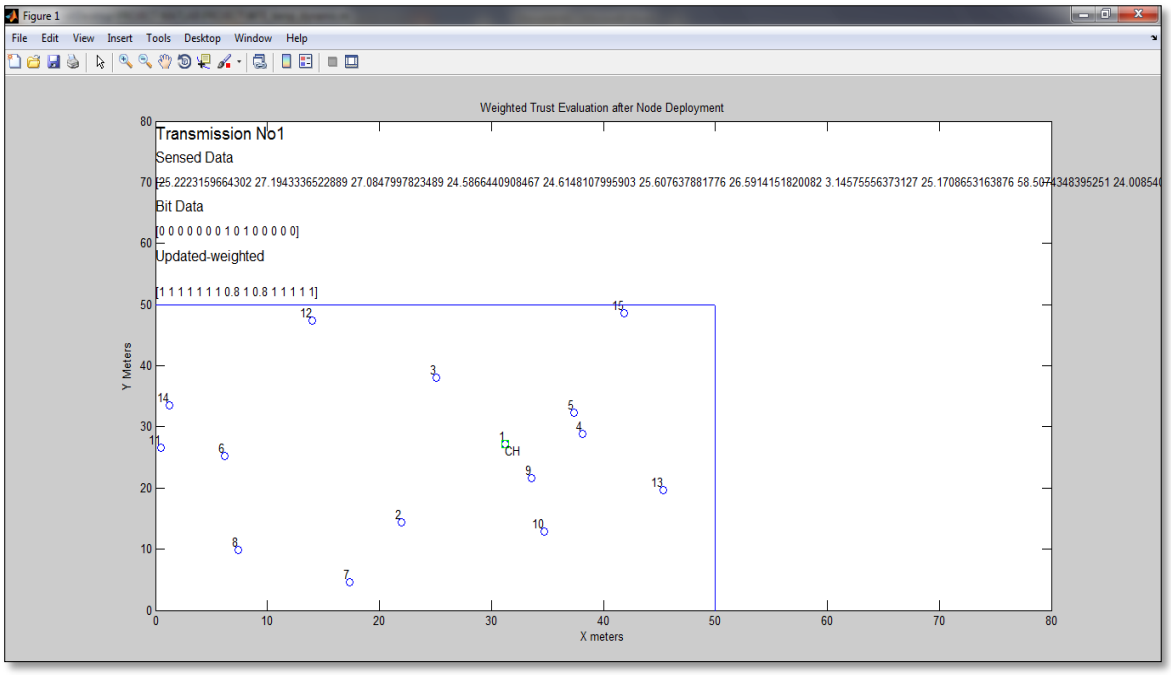


Figure 4.1.1 Random Deployment of nodes

Dynamically changing cluster head is introduced such that cluster head changes after every 5 cycles of operation. The weights of all nodes stored in the cluster head are transferred to the next node which is to become the cluster head in the round robin schedule, one cycle before the actual change occurs as shown in fig 4.1.2 and 4.1.3).

The number of cycles after which cluster head is to be changed is set to five. In figure 4.1.2 we can see how the at transmission 4 the data is being transferred to the next node, and also cluster head changes from node 1 to node 2 at cycle 5. The figures also show the updated weights and the region marked within the blue boundary shows the grid area. 15 nodes are created randomly. The sensed and sent temperature data is also shown in the figure.

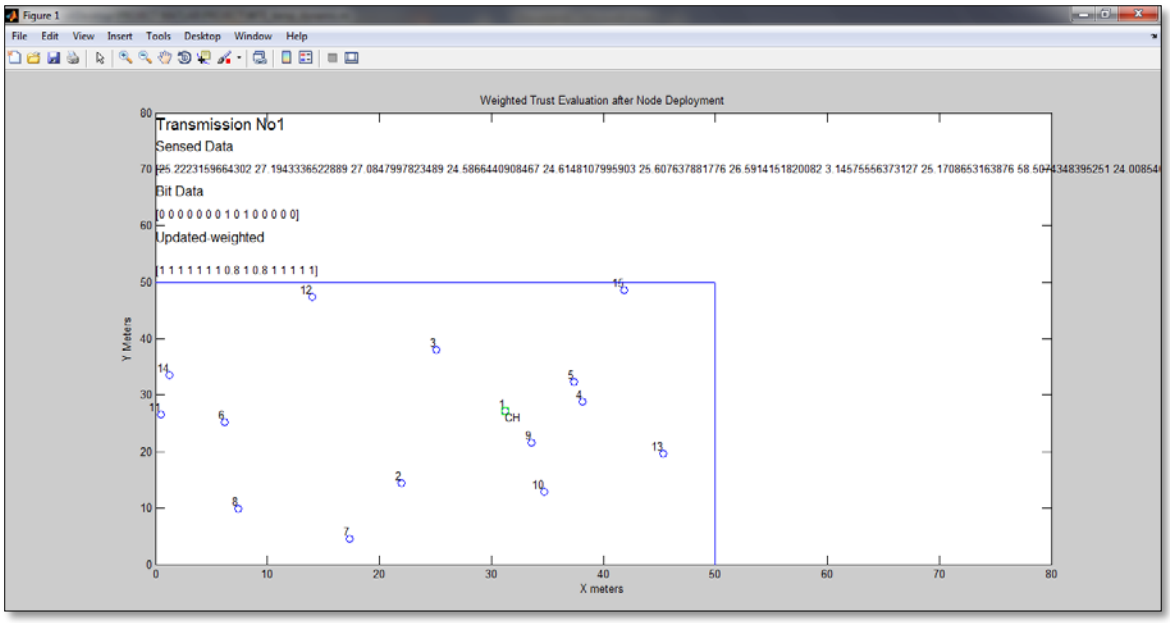


Figure 4.1.2 Transfer of data in one cycle before the change of cluster head

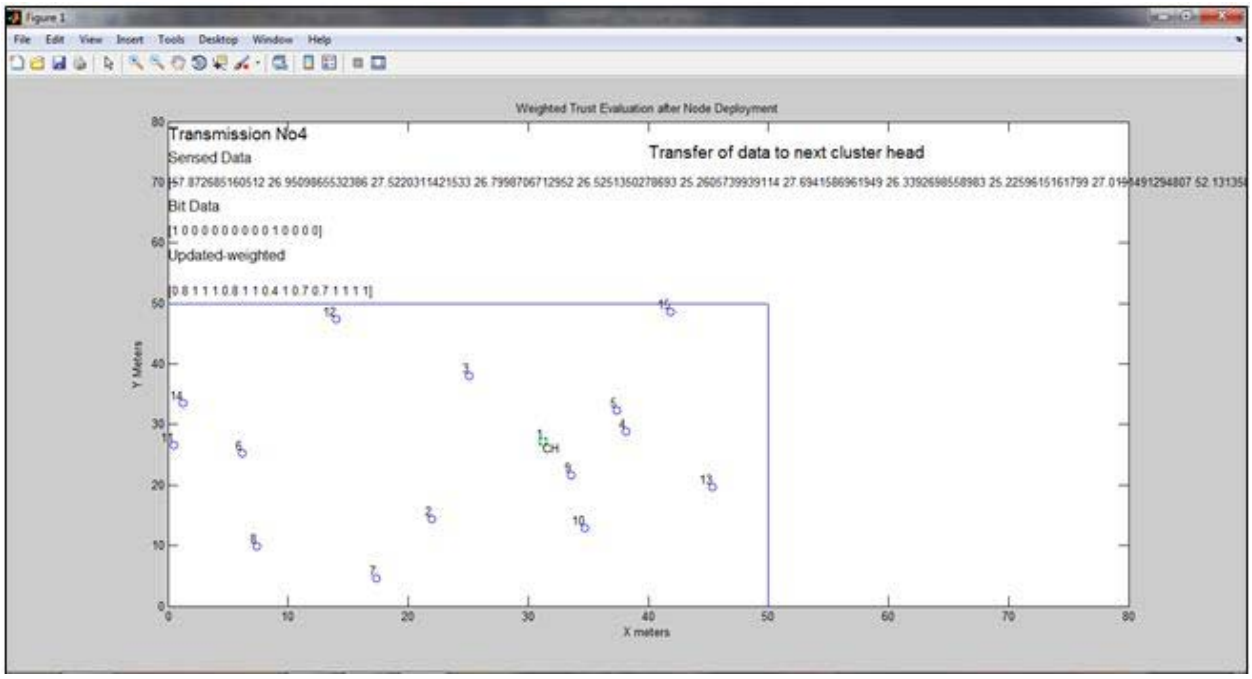


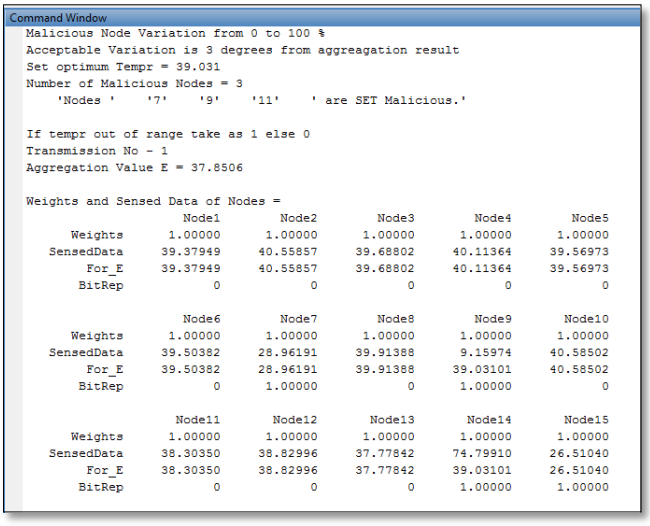
Figure 4.1.3 Selection of new cluster head

Since MAC protocol is being implemented and all nodes use AODV transmission as explained in Section 3, all nodes within the cluster maintain routes to the cluster head. When the cluster head changes new routes need to be found to the new cluster head, which is efficiently found using AODV protocol. The MAC protocol provides for TDMA transfer of data between various nodes and single cluster head. Also some versions of

MAC can be used to provide CSMD/CDMA transmission between the SN and The FN (cluster head). Practically ZIGBEE networks are used where TDMA is employed.

DETECTION OF MALICIOUS NODES:

The detection of malicious nodes is shown in figures below. We set nodes 7, 9, 11 as malicious before the execution and they are detected accurately with average response time of 9.5 transmissions. Also the aggregation result, Sensed data and the bit representation after each transmission cycle is shown too. Here the normal surrounding temperature is sensed by the nodes which for the area are set between 10^o to 50^o. So the temperature should lie between these values. The optimum temperature selected is 39.031^o for simulation. We can see that the aggregation value computed is 37.85^o. The local variation 'var' allowed was set for **5% of** (*max temp – min temp)* i.e.5% of (50 - 10). A maximum variation of 2^o which is between 37.031^o and 41.031^o is allowed variation. If the sensed value exceeds this variation *weight reduction* is done as explained in WTE section 3.



```
Command Window
  Malicious Node Variation from 0 to 100 %
  Acceptable Variation is 3 degrees from aggreagation result
  Set optimum Tempr = 39.031
  Number of Malicious Nodes = 3
      'Nodes '    '7'    '9'    '11'    ' are SET Malicious.'

  If tempr out of range take as 1 else 0
  Transmission No - 1
  Aggregation Value E = 37.8506

  Weights and Sensed Data of Nodes =
                      Node1        Node2        Node3        Node4        Node5
        Weights       1.00000      1.00000      1.00000      1.00000      1.00000
     SensedData      39.37949     40.55857     39.68802     40.11364     39.56973
          For_E      39.37949     40.55857     39.68802     40.11364     39.56973
         BitRep             0            0            0            0            0

                      Node6        Node7        Node8        Node9       Node10
        Weights       1.00000      1.00000      1.00000      1.00000      1.00000
     SensedData      39.50382     28.96191     39.91388      9.15974     40.58502
          For_E      39.50382     28.96191     39.91388     39.03101     40.58502
         BitRep             0      1.00000            0      1.00000            0

                     Node11       Node12       Node13       Node14       Node15
        Weights       1.00000      1.00000      1.00000      1.00000      1.00000
     SensedData      38.30350     38.82996     37.77842     74.79910     26.51040
          For_E      38.30350     38.82996     37.77842     39.03101     26.51040
         BitRep             0            0            0      1.00000      1.00000
```

Figure 4.1.4 The simulation set values are shown

Here $\alpha = 0.2$, $\beta = 0.05$, $t_n = 2$, $P_{ma} = 0.6$, $P_t = 0.05$. Hence the probability of malicious nodes ending wrong data is high and the probability of normal nodes sending wrong data is less. Hence now the system works very efficiently. In figures 4.1.5, 4.1.6, 4.1.7 we can see how the nodes are set initially. The cluster head changes every 5 cycles and also at transmission 8 all malicious nodes are detected.

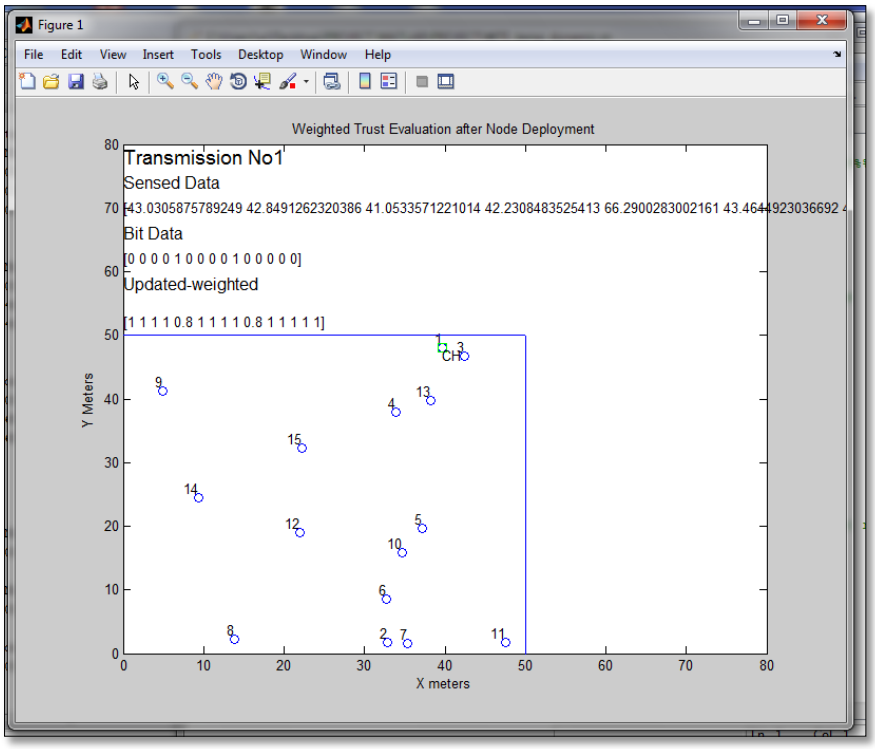


Figure 4.1.5 Initial deployment of nodes

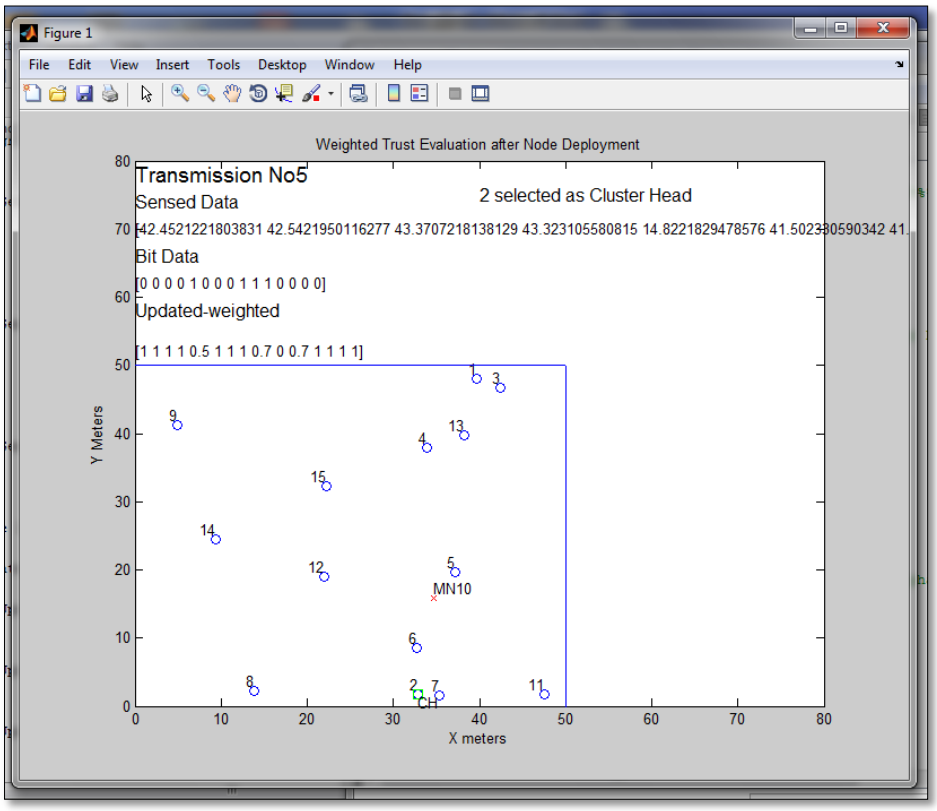


Figure 4.1.6 Change of cluster head

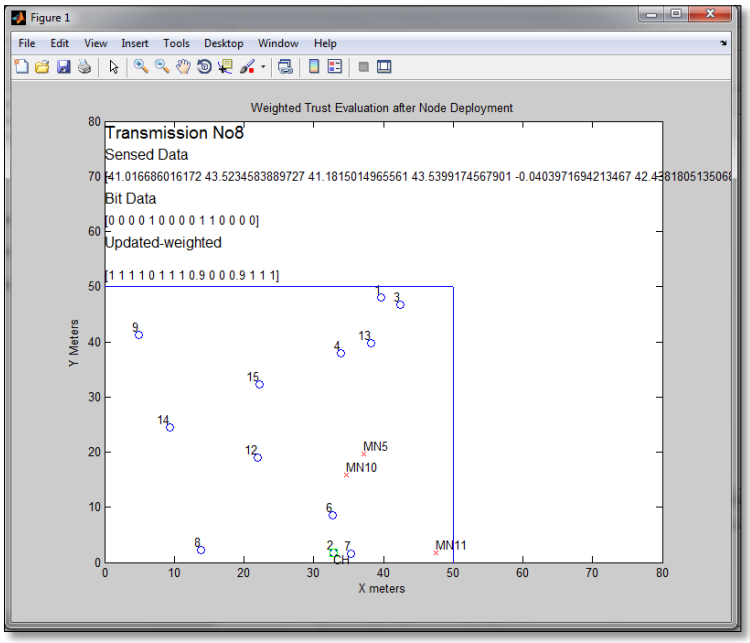


Figure 4.1.7 Malicious nodes detected

```
Node 7 is detetcted at 7 transmission
Node 9 is detetcted at 7 transmission
Node 11 is detetcted at 6 transmission
The Average response time is 6.6667Transmissions.

mdr =

     1

The malicious node detection ratio is 1
```

Figure 4.1.8 Average response time, MDR and MR

The average response time was found to be *6.667* transmissions. Also since there is no misdetection and all malicious nodes are detected MDR is found to be 1 and MR too is found to be 0.

Now we take another example where $\alpha = 0.2$, $\beta = 0.05$, $t_n = 2$, $\mathbf{P_{ma} = 0.4, P_t = 0.25.}$ Hence the probability of malicious nodes ending wrong data is *optimum* and the probability of normal nodes sending wrong data is *higher*. Now the system performance is seen. In figures 4.1.9, 4.1.10, 4.1.11 we can see how the nodes are set initially. The cluster head changes every 5 cycles and also at transmission 8 all malicious nodes are detected. The nodes set malicious are 5, 10, and 11.

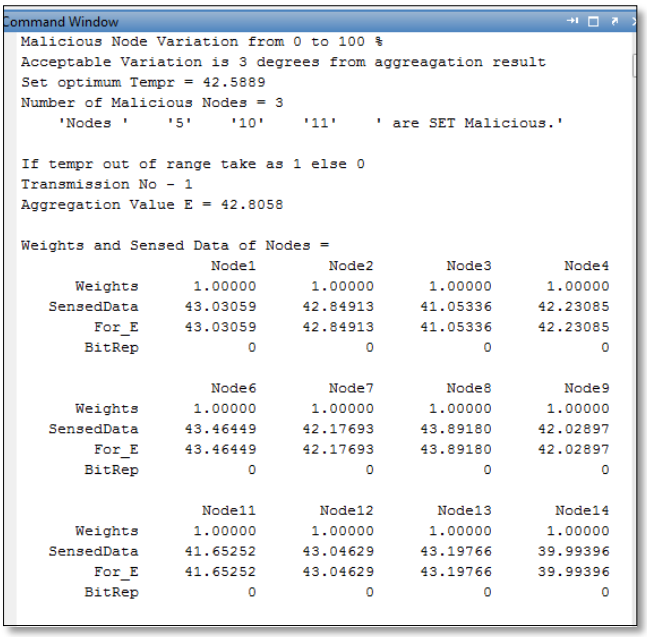


Figure 4.1.9 Node setting

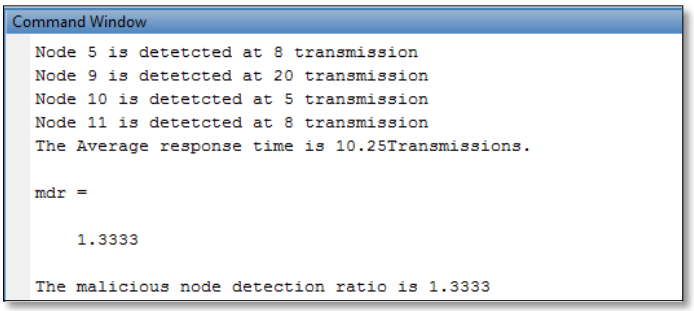


Figure 4.1.10 Output

Since the normal nodes send wrong data with higher probability here there is a misdetection as seen. Node 9 sends too much wrong data due to which it is detected as malicious. The MDR found to be 1.333 proving that there is misdetection and MR is found to be 0.333, indicating 1 misdetected node.

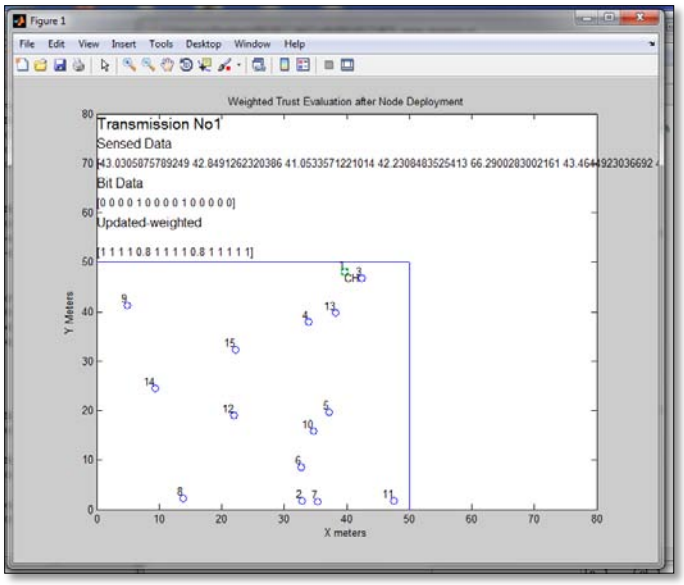


Figure 4.1.11 Deployment of nodes

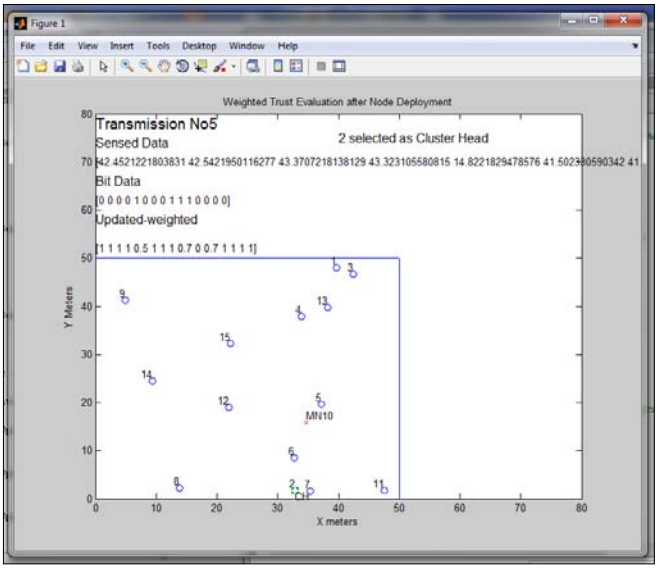


Figure 4.1.12 Detected malicious nodes and change of cluster head

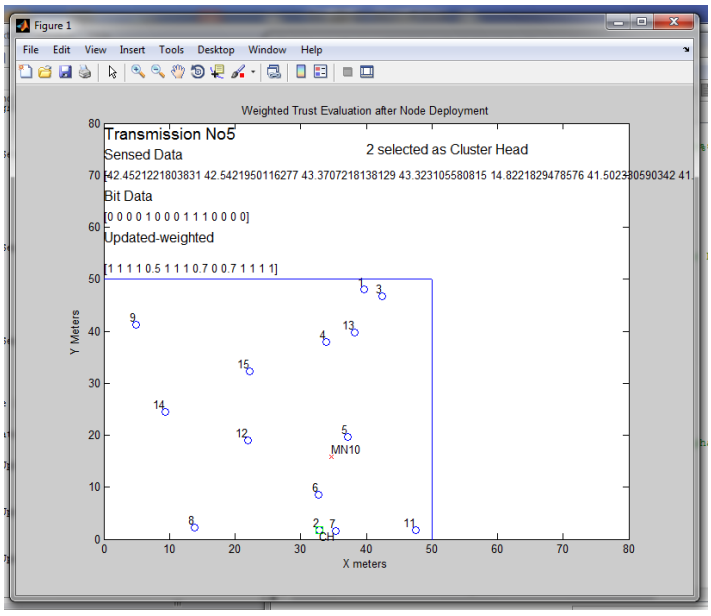


Figure 4.1.13 Detected malicious nodes

4.1.1 VARIATION OF P_{ma} WITH MDR MR AND RT:

We evaluate MDR, MR and RT for various values of P_{ma} *varying from 0 to 1 in steps of 0.1*, with $\alpha = 0.2$, $\beta = 0.1$, $t_n = 2$, $\theta1 = 0.6$, $\theta2 = 0.2$, $P_t = 0.05$ $\theta = 0.2$ for 50, 100, and 200 cycles of operation. The results are shown in *Figures 5.14, 5.15* and *5.16*, respectively.

- Figure 5.14 shows the variation of P_{ma} vs. MDR:

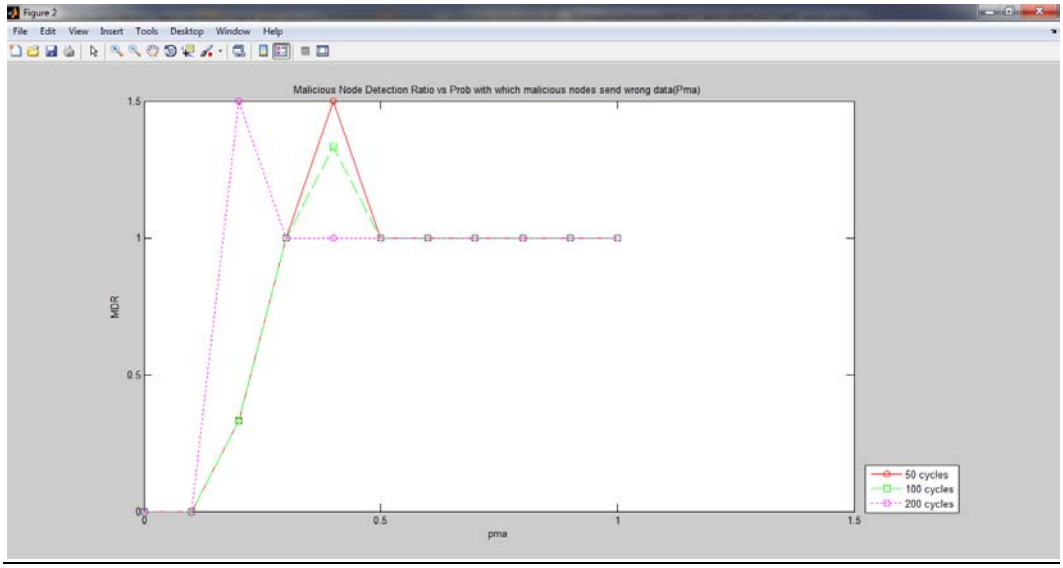


Figure 4.1.14 MDR vs P_{ma}

As we can see as P_{ma} - Probability, with which the malicious node generates wrong data, is varied from 0 to 1 the misdetection ratio is varying. MDR is low at lower values of P_{ma} because at these values the malicious nodes behave as normal nodes, not generating any wrong values. As P_{ma} is increased the MDR becomes 1, i.e. there is correct detection of malicious nodes since malicious nodes tend to give more wrong data at higher P_{ma}.

- Figure 4.1.15 shows variation of MR vs P_{ma}:

The MR variation is shown with respect to P_{ma} in figure 4.1.15. Here there is variation when the simulation occurs only for 50 cycles. But when the cycles increase the misdetection reduces such that the misdetected nodes are '0' hence making the MR zero. This happens since with more cycles the error is less, as weight recovery would prevent misdetection. Also at *low P_{ma} the ambiguity of node being malicious or not is too high and WTE cannot be performed if pma is too low*. More cycles gives the edge of more time for correct detection of the malicious nodes.

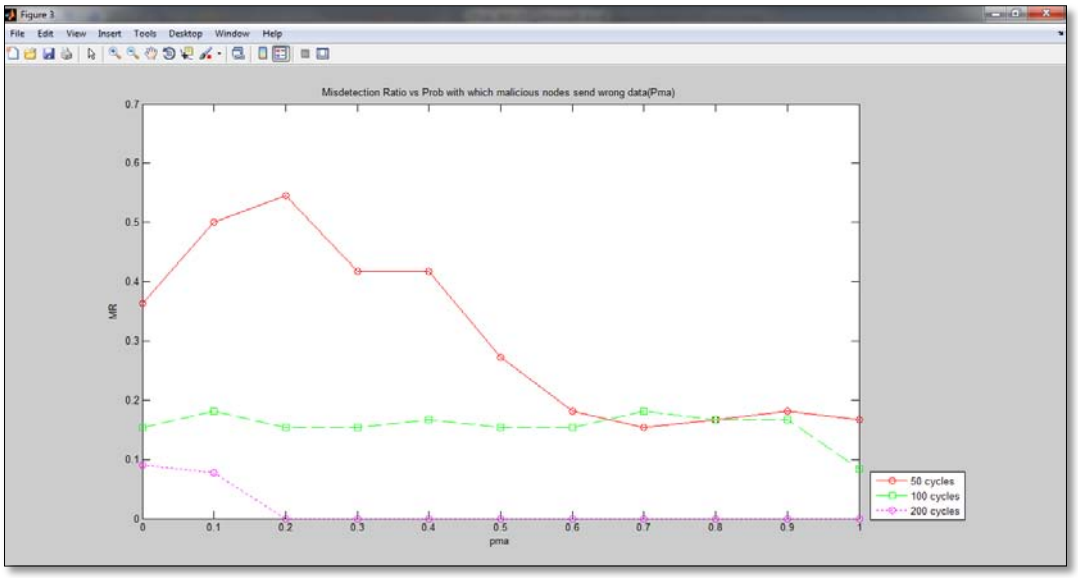


Figure 4.1.15 MR vs P_{ma}

❖ Figure 4.1.16 shows variation of RT vs P_{ma}:

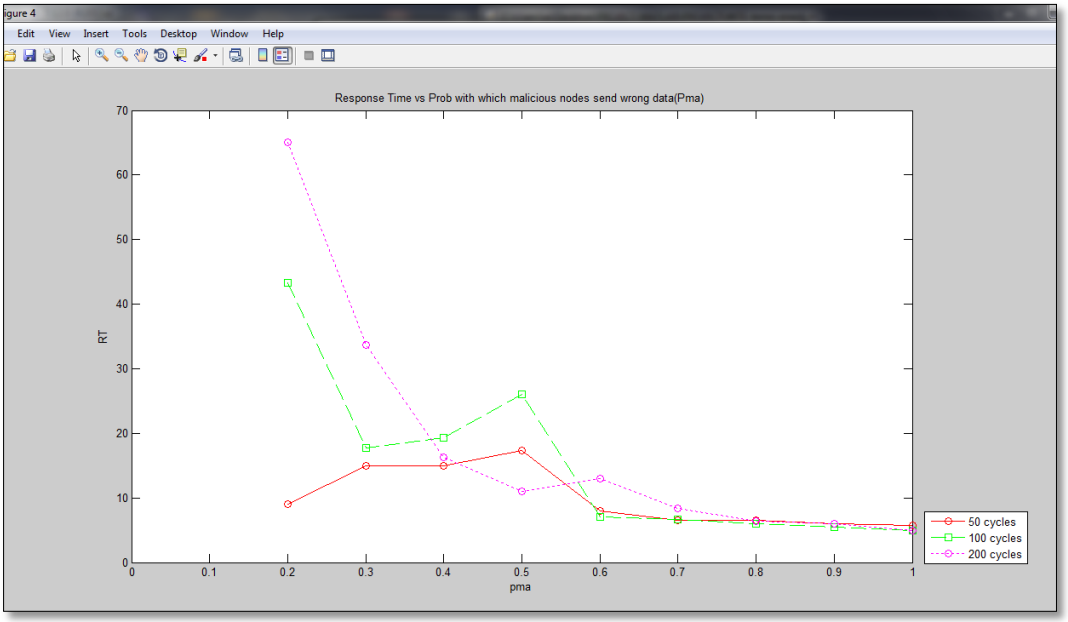


Figure 4.1.16 RT vs P_{ma}

Since when P_{ma} is small malicious nodes behave as normal nodes there is no detection hence response time is infinite since denominator becomes 0, shown as high values at low P_{ma}. As P_{ma} is increased the response time is reduced drastically since it becomes easier to detect malicious nodes at higher P_{ma}. The average RT comes to around 5 transmission cycles for any number of cycles simulated.

4.1.2 VARIATION OF ALPHA (α) WITH MDR MR AND RT:

We evaluate MDR, MR and RT for various values of *α varying from 0 to 1 in steps of 0.1,* with P_{ma} = 0.2, β = 0.1, t_n = 2, θ1= 0.6, θ2 =0.2, P_t = 0.05 θ = 0.2 for 50, 100, and 200 cycles of operation. The results are shown in *Figures 4.1.17, 4.1.18* and *4.1.19*, respectively.

- Figure 4.1.17 shows variation of MDR vs α:

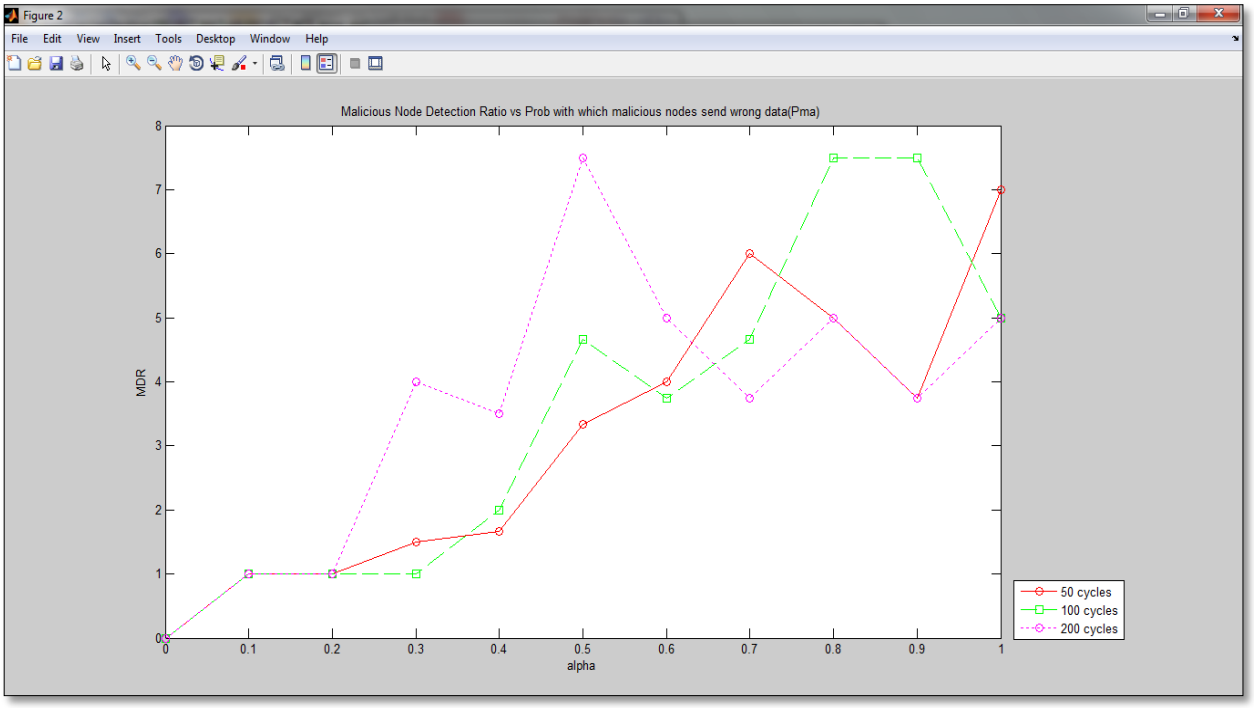


Figure 4.1.17 MDR vs α

As we can see as α is increased MDR increases. This is due to the fact that when α is too high when normal nodes send a wrong value due to faults the weight is reduced drastically which results in many normal nodes being misdetected as malicious.

- Figure 4.1.18 shows variation of MR vs α:

Here, we can see that alpha does **not have much effect** on the misdetection ratio. Smaller values of alpha need more simulation cycles to detect malicious node, hence we have higher MR's for simulation consisting of 50 cycles than with others. As the cycles increase the MR tends to become ideal.

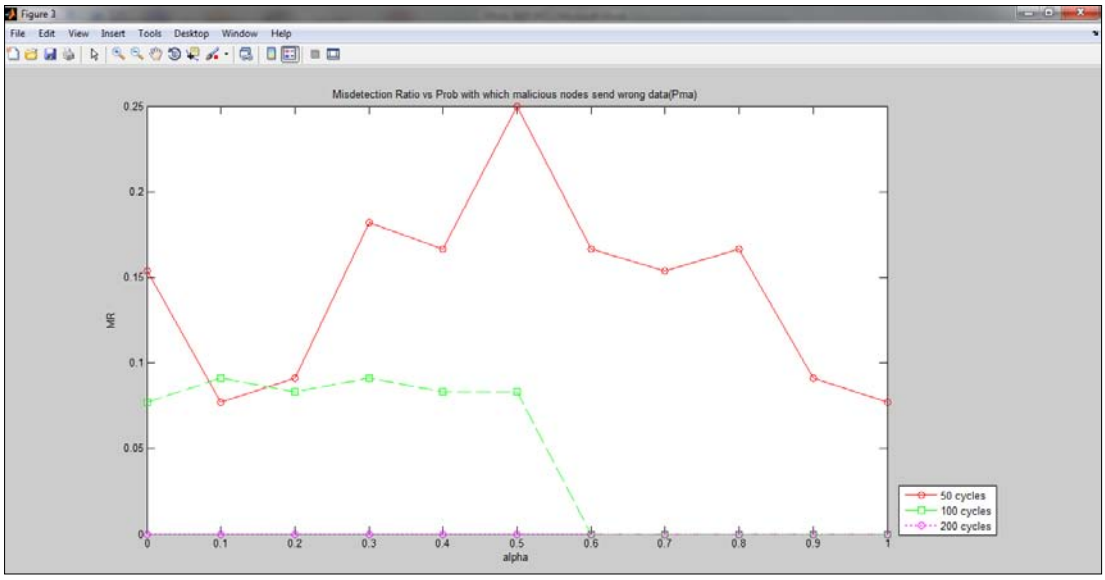


Figure 4.1.18 MR vs α

❖ Figure 4.1.19 shows variation of RT vs α:

Here we can see that as α is increased the response time decreases as the weight penalty is increased the weights of nodes sending wrong data decrease at a very fast rate providing a very good response time. On contrary it is seen that at smaller values of α response time should be high, but it is seen as low, because with α small malicious nodes are not detected at all and there is high misdetection.

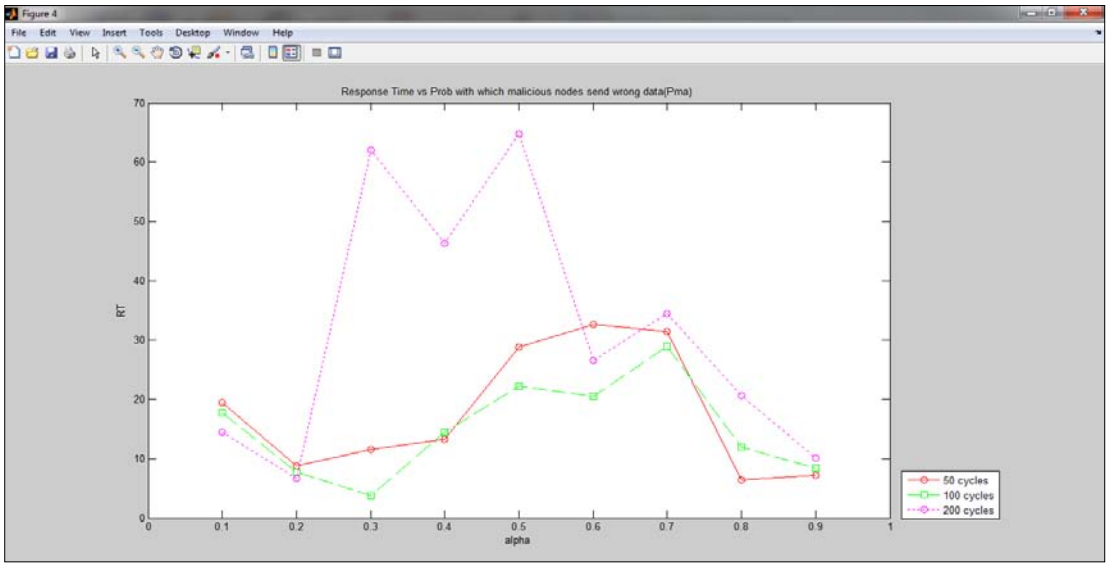


Figure 4.1.19 RT vs α

4.1.3 VARIATION OF BETA (β) WITH MDR MR AND RT:

We evaluate MDR, MR and RT for various values of β *varying from 0 to 1 in steps of 0.1,* with $\alpha = 0.2$, $P_{ma} = 0.6$, $t_n = 2$, $\theta1 = 0.6$, $\theta2 = 0.2$, $P_t = 0.05$ $\theta = 0.2$ for 50, 100, and 200 cycles of operation. The results are shown in *Figures 4.1.20, 4.1.21* and *4.1.22*, respectively.

- Figure 4.1.20 shows variation of MDR vs β:

Figure 4.1.20 shows that when β is small there is no weight recovery and hence misdetection occurs. As β slightly increases it provides stability and prevents misdetection.

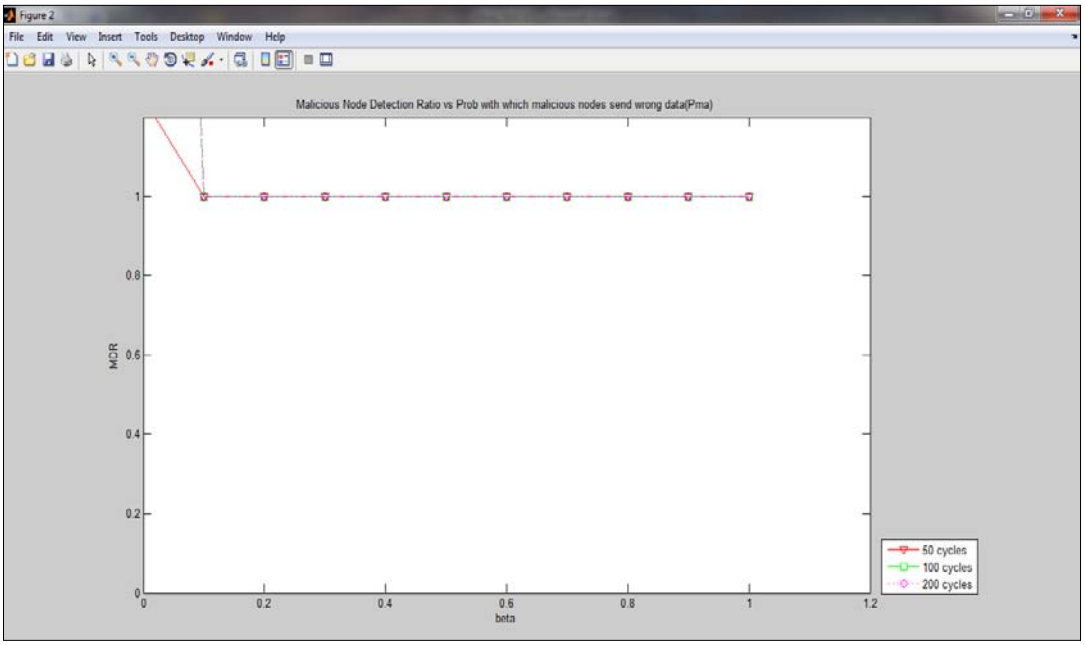


Figure 4.1.20 MDR vs β

- Figure 4.1.21 shows variation of MR vs β:

Figure 4.1.21 shows that β has very low impact effect on the MR. the misdetection due to β is very less and this occurs only at low β values where there is no weight recovery and normal nodes are detected malicious.

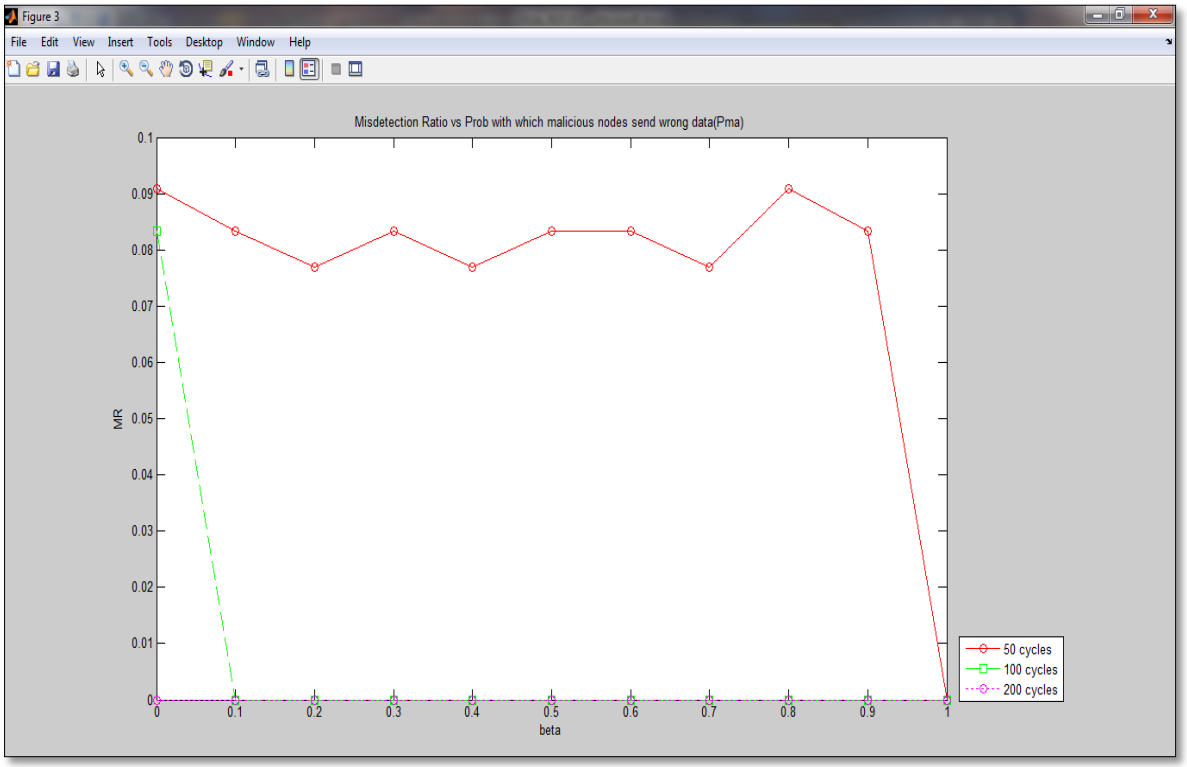


Figure 4.1.21 MR vs β

❖ Figure 4.1.22 shows variation of RT vs β:

Figure 4.1.21 shows that β has very low impact effect on the MR. the misdetection due to β is very less and this occurs only at low β values where there is no weight recovery and normal nodes are detected malicious.

As seen when β is high the response time is high this is because there is weight recovery increases the weight of malicious node even if it sends correct data with minimum probability. Even if the node sends correct data once in ten times the weight is recovered and hence response time to detect malicious node become too high.

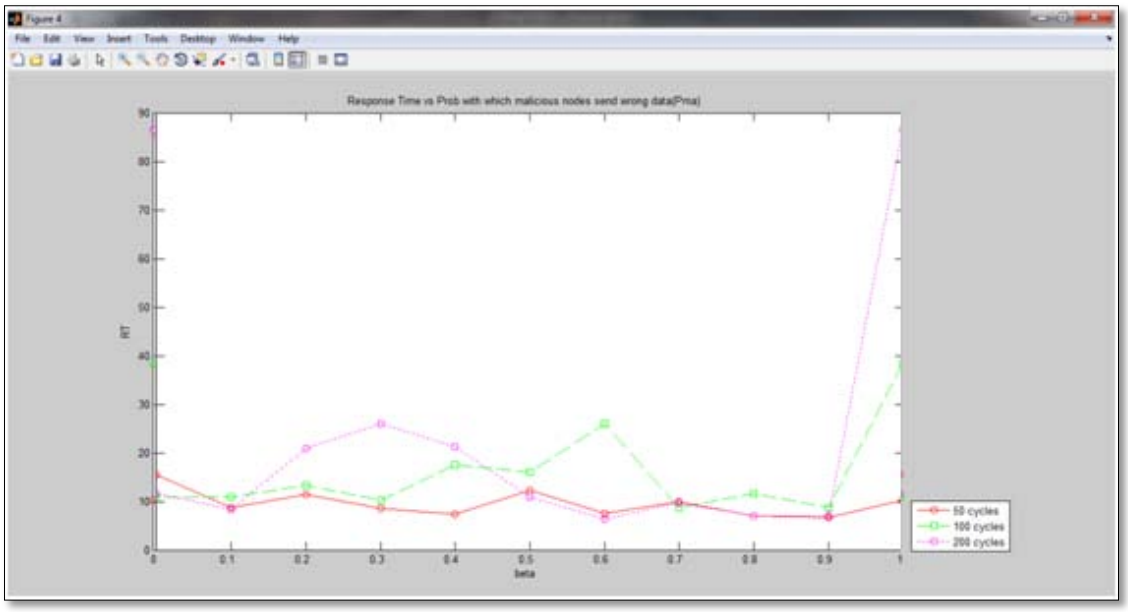


Figure 4.1.22 MR vs β

4.1.4 INTRODUCTION OF WTE IN A GRID BASED – HIERARCHICAL WSN:

In real life scenarios since the area is too large the area is divided in grids for easy handling of nodes. In the simulation we divide the area into grids, where each grid forms a cluster. In each cluster WTE is performed as done in above analysis. For simulation purpose we divide the area into 9 grids as shown in the figure 4.1.23. The cluster heads are dynamically changing in each grid for every 5 cycles. We set $\alpha = 0.2$, $\beta = 0.1$, $t_n = 2$, $\theta 1 = 0.6$, $\theta 2 = 0.2$, $P_t = 0.05$ and $P_{ma} = 0.6$ for 50 cycles of operation.

Figure 4.1.23 shows all the nodes set as malicious in each cluster with the above set parameters. Figure 4.1.24 shows random deployment of nodes in the region divided into grids. Figure 4.1.25 shows how the cluster head changes after 5 cycles of operation. Data is passed between the heads one cycle before the cluster head changes. Figure 4.1.26 shows the detection of the malicious nodes in each cluster.

We can see that all malicious nodes are detected correctly in each clustering figure 4.1.26 and hence no misdetection is observed.

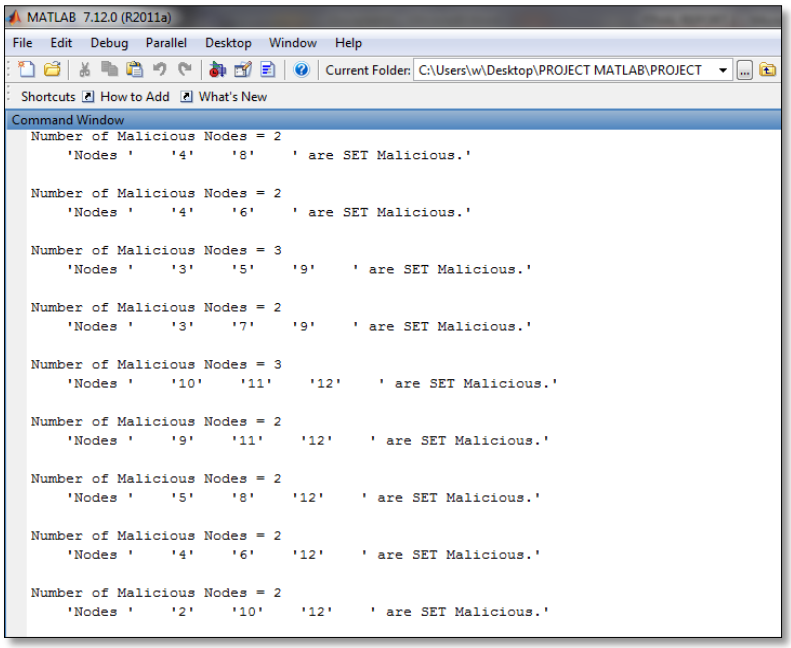


Figure 4.1.23 Nodes set as malicious in each cluster

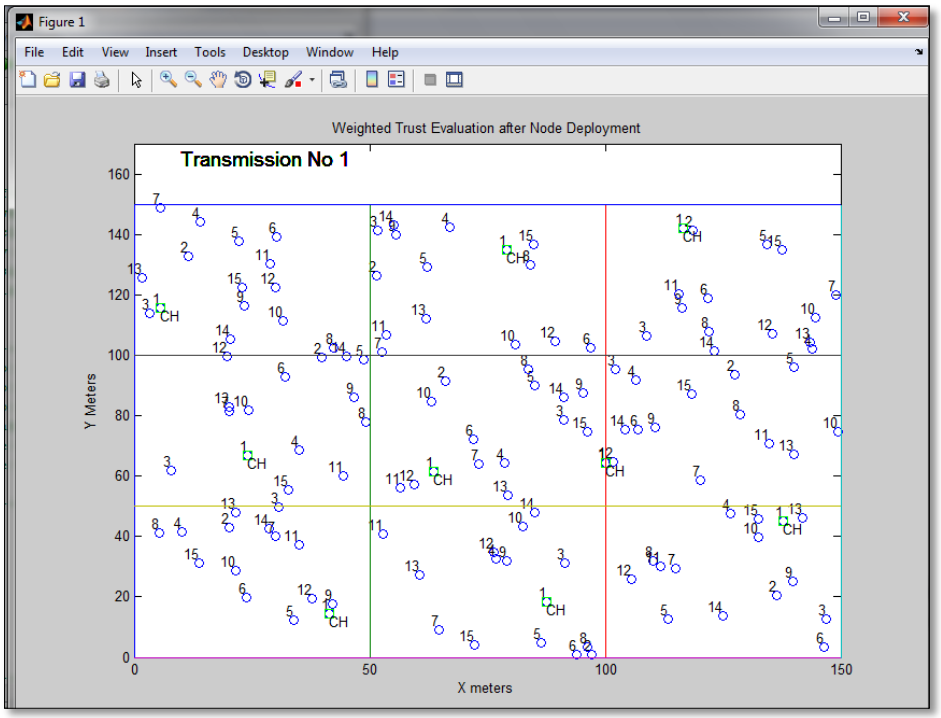


Figure 4.1.24 Deployment of nodes in grid based clustering

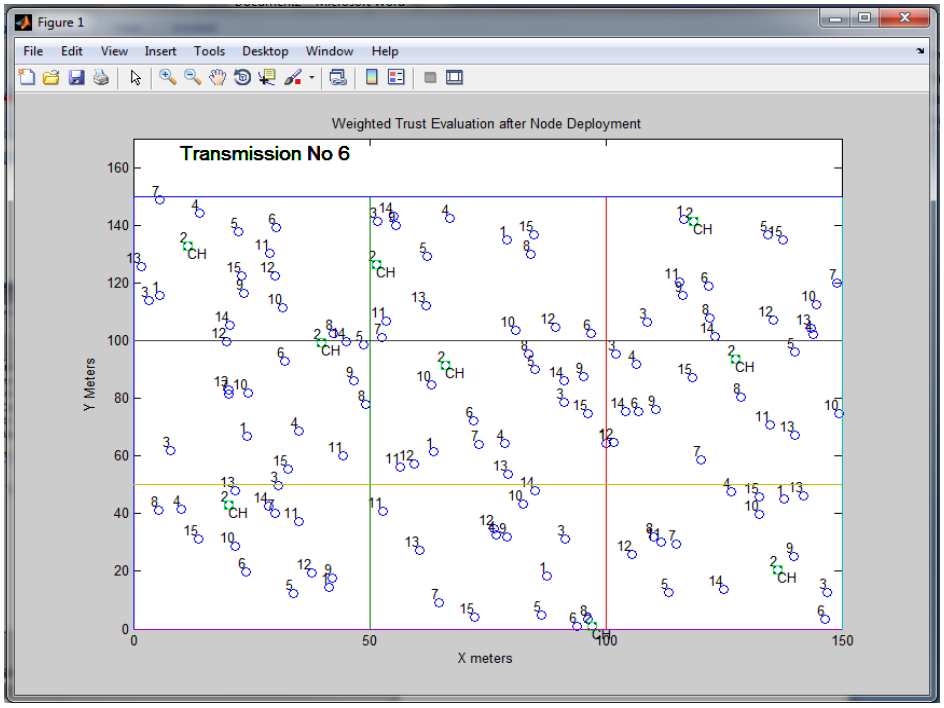


Figure 4.1.25 Deployment of nodes in grid based clustering

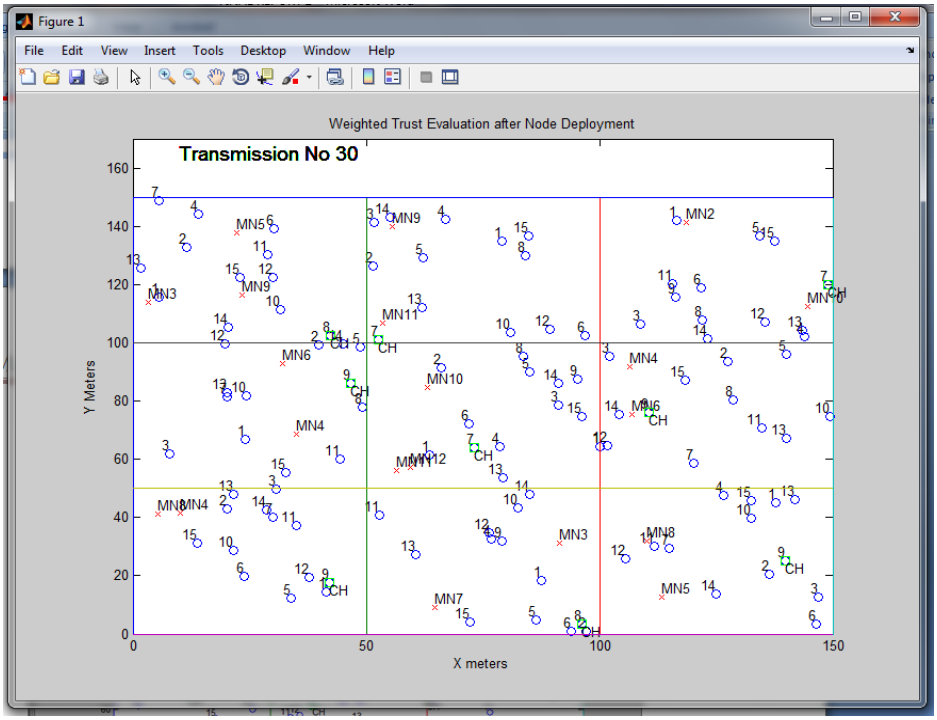


Figure 4.1.26 Malicious Node Detection

4.2 EXTENDED WEIGHTED TRUST EVALUATION

As discussed in section 3 the extended WTE involves regions in multiple clusters. We simulate an area with 4 grids. The event region appears to be located in all grids as shown

the simulation figure 4.2.1. The *blue circular region* shows the actual event region set for simulation.

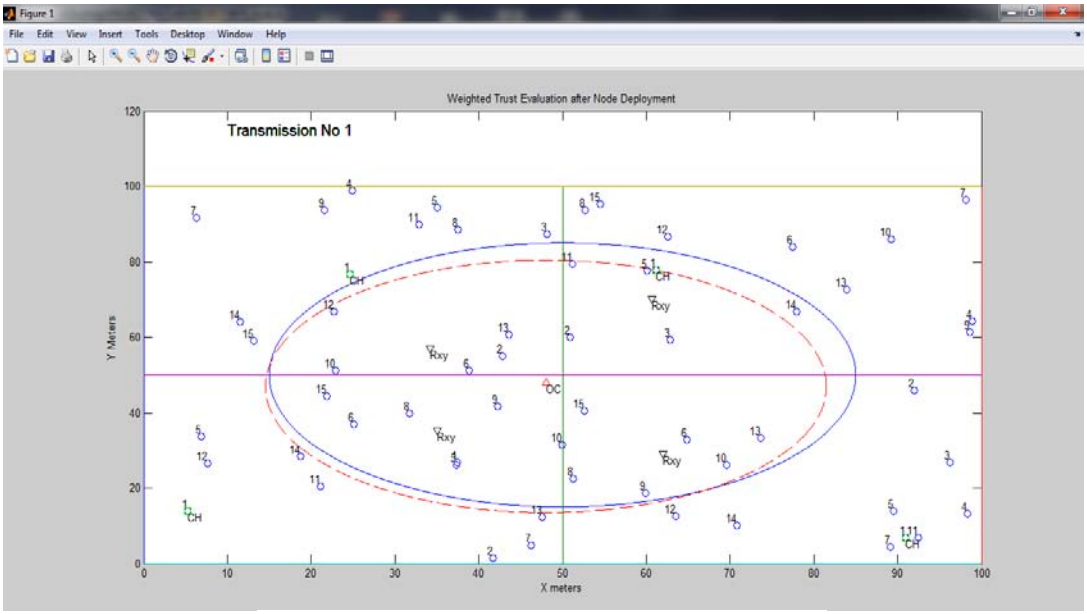


Figure 4.2.1 Localized Event Deployment

In the event region we have set the normal nodes to send '1' i.e. alarm and malicious nodes send wrong data based a probability P_{ma}. In no event region all normal nodes are sending '0's.

DETERMINATION OF EVENT REGION:

We can see R_{xy} defined in each grid which is the weighted positions of all nodes sending 1. The R_{ABCD} is the weighted position of all centres which gives us the '*OPTIMUM CENTRE'* . The event region is the region within the circular radius of r' $\leq$ r. Where 'r' is weighted mean distance between R_{ABCD} and R_{xy} in all the grids. We have set r' = 1.75(r). This is done because since the optimum centre is not accurate we perform WTE in the region slightly lesser. The determined region is the region within the *red circular radius*.

Usually, the event does not occur continuously it occurs in a region for a sppecified interval. In our simulation we have set so that the *event occurs during the transmission intervals i =3 to 8*, after which there is no localised event occuring. Figure 4.2.2 shows the basic deployment in grids, figure 4.2.3 shows the event region defined (within blue circle) and the optimum area found using the alforithm defined above (within

red circle). The appropraite WTE's are applied in the event / no event regions and we get the results as in figure 4.2.6. The grid numbers are as shown in figure below:

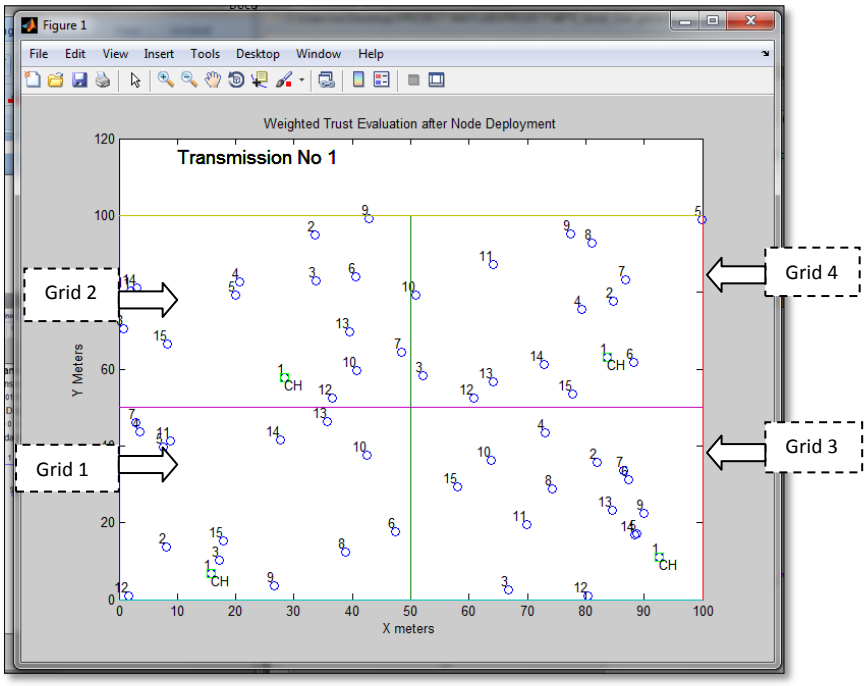


Figure 4.2.2 Initial Deployment

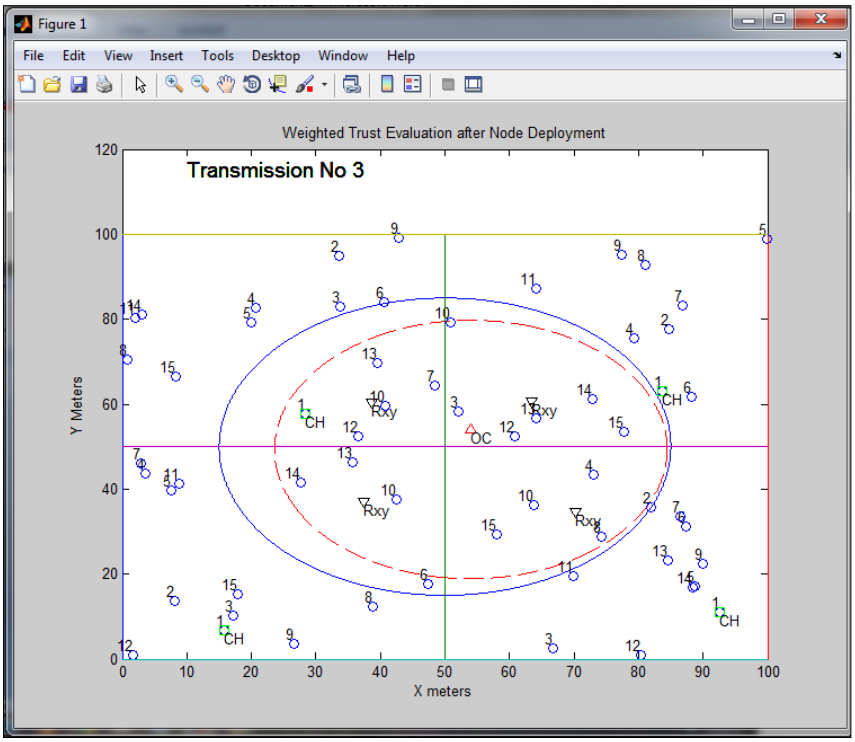


Figure 4.2.3 Localized event occurring at transmission 3

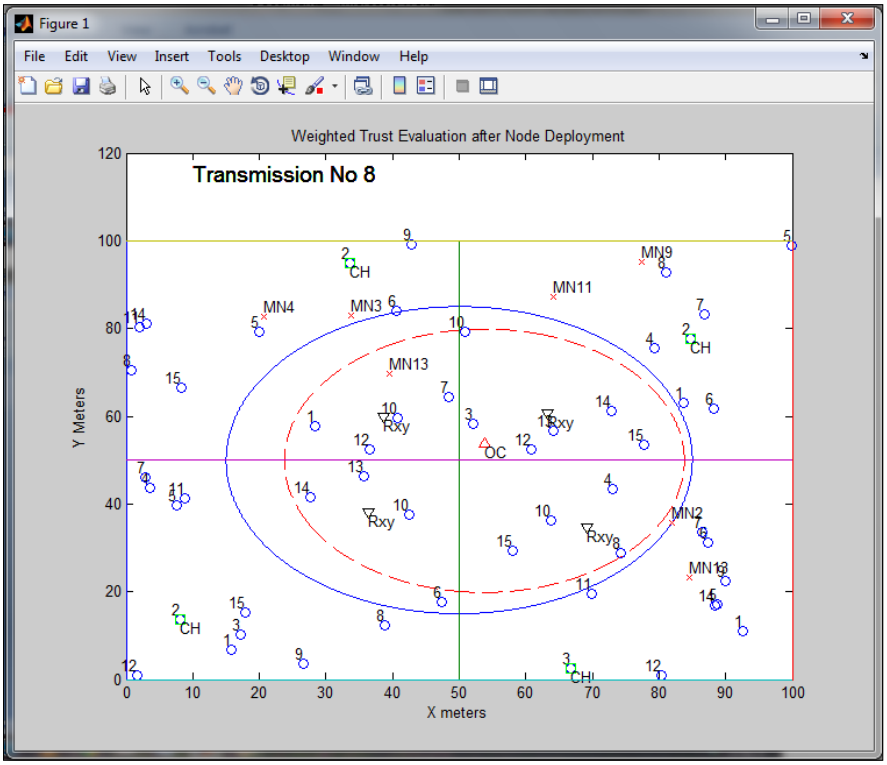


Figure 4.2.4 End of local event at transmission 8

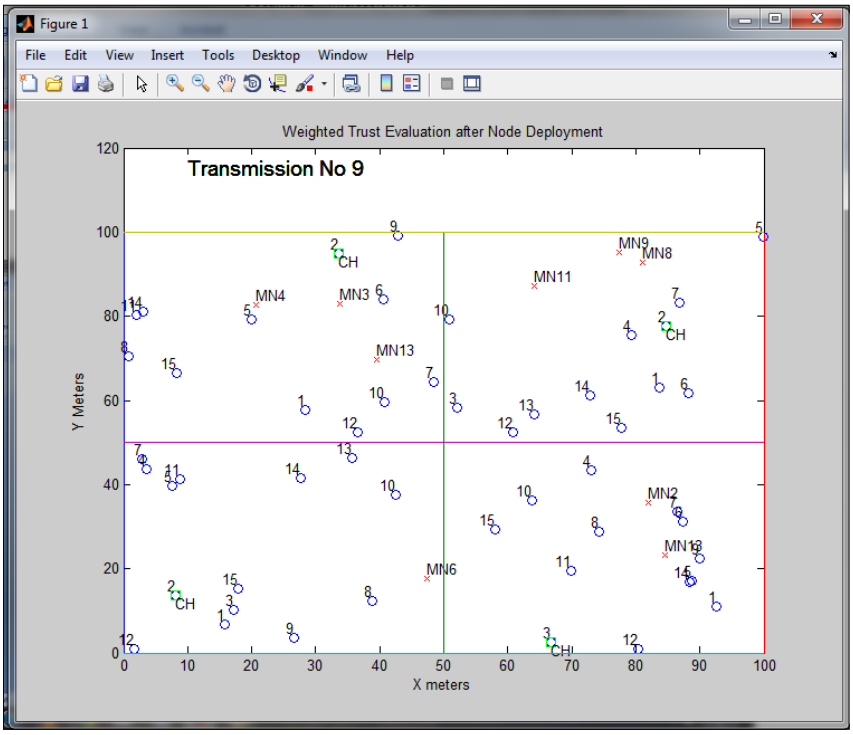


Figure 4.2.5 No localized event at transmission 9

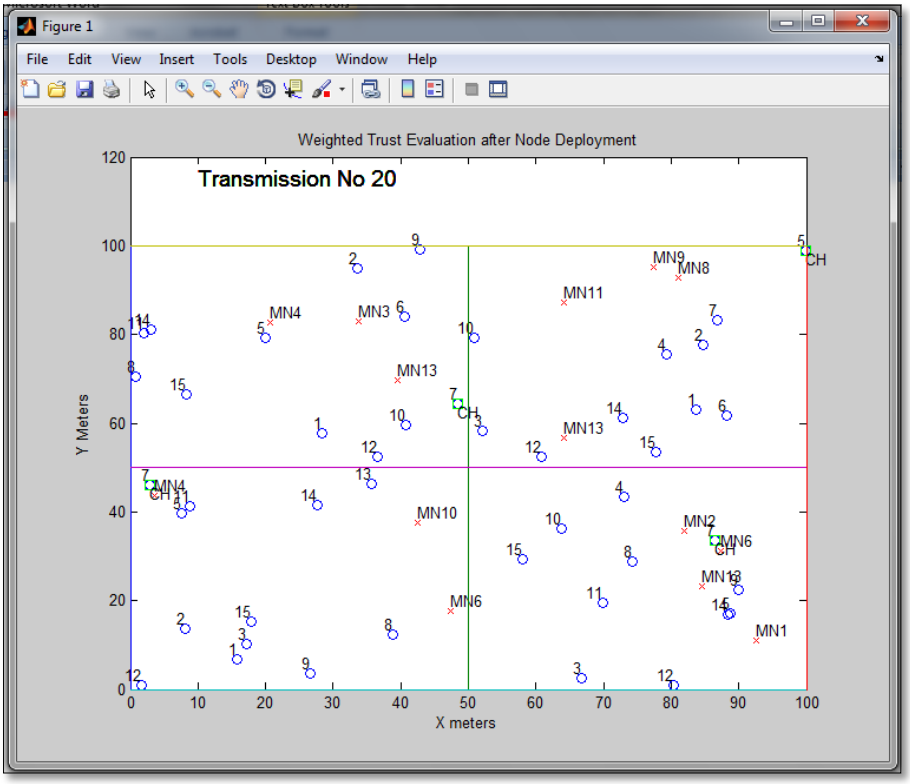


Figure 4.2.6 Detected malicious nodes

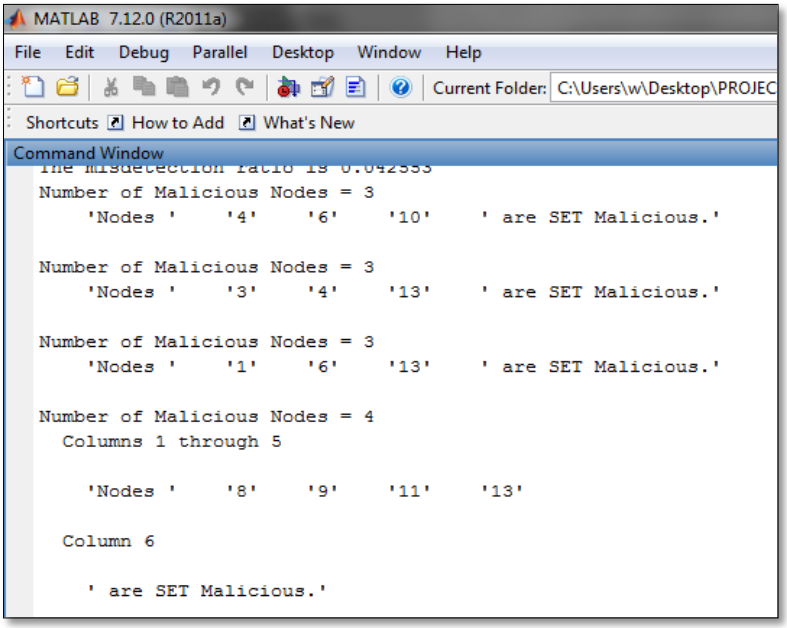


Figure 4.2.7 Initial Set Up

As we can see in each in grid 1 node 4, 6, 10 are set malicious, in grid 1 nodes 3, 4, 13 are set malicious, grid 1 nodes 1, 6, 13 are set malicious and grid 4 nodes 8, 9, 10, 11 are set malicious. Total set malicious nodes are 13.

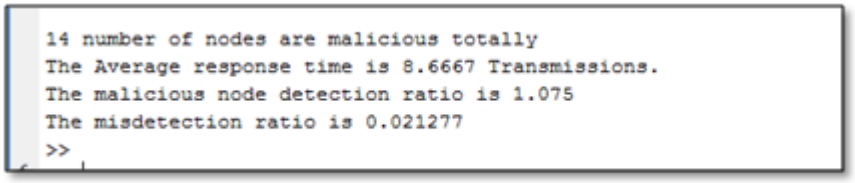


Figure 4.2.8 Output

As we can see 14 nodes are detected as malicious. *Node 2 in grid 3 is the misdetected node* as seen in figure 5.32. Hence due to misdetection we have Average response time is 8.667 transmissions, also MDR = 1.075 and MR = 0.0212. Hence we can see that if 'P_{ma}' is high and 'P_t' is lower we get an accurate result and exact detection of malicious nodes is possible.

VARIATION OF P_{ma}, ALPHA (A) AND BETA (B) TO MEASURE MISDETECTION:

More misdetection of nodes occurs in the localized sensing scenario due to imperfect estimation of the optimum centre, computed by the algorithm. Here we concentrate on to analyse this misdetection factor. Since the 'Malicious Detection Ratio (MDR)' is the main parameters which represent the amount of misdetection in the WSN's.

Here we vary the parameters P_{ma}, alpha (α) and beta (β) to find MDR and, find their effect on the WSN performance. The parameters are set as α = 0.2, P_{ma} = 0.6, t_n = 2, θ1= 0.6, θ2 =0.2, P_t = 0.05 for 50 cycles of operation. The results are shown in *Figures 5.20, 5.21* and *5.22*, respectively.

- *Variation of P_{ma} with MDR and MR:*

From the variation of P_{ma} with MDR in figure 4.2.9 we can see that it has the same impact as in normal WTE implementation where initially MDR is below 1 indicating malicious nodes are not detected. For higher P_{ma} we can see that the MDR is around 1 since here the malicious nodes always send wrong data ,there is easy detection of malicious nodes . But there is a lot of misdetection for Pma below 0.7. When P_{ma} is low

malicious nodes are not detected and when P_{ma} in mid region too many normal nodes tend to be misdetected.

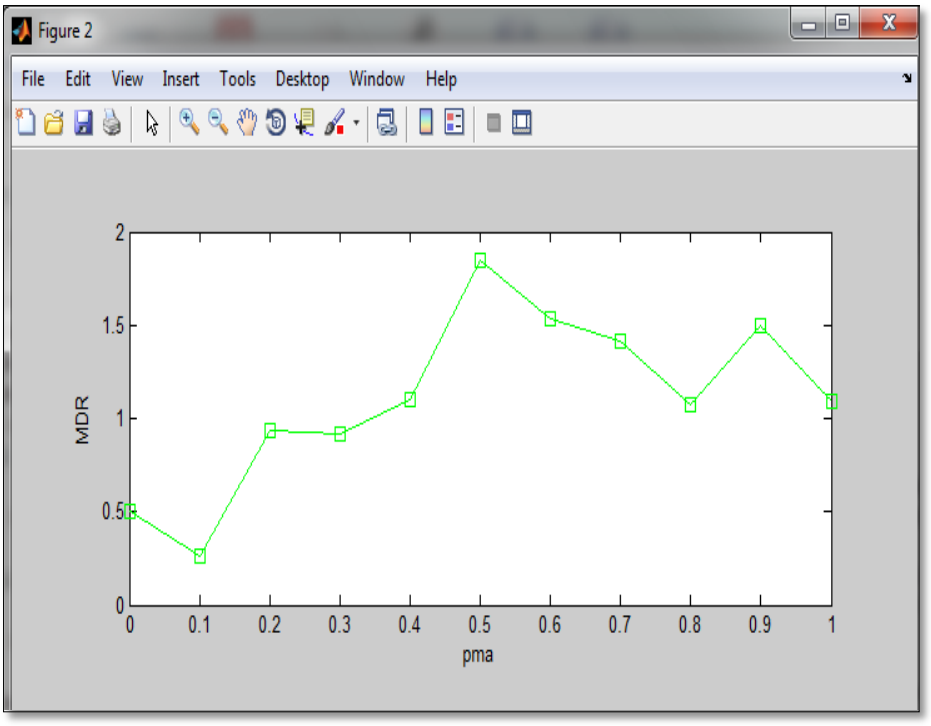


Figure 4.2.9 P_{ma} vs MDR

❖ *Variation of alpha(α) of with MDR and MR:*

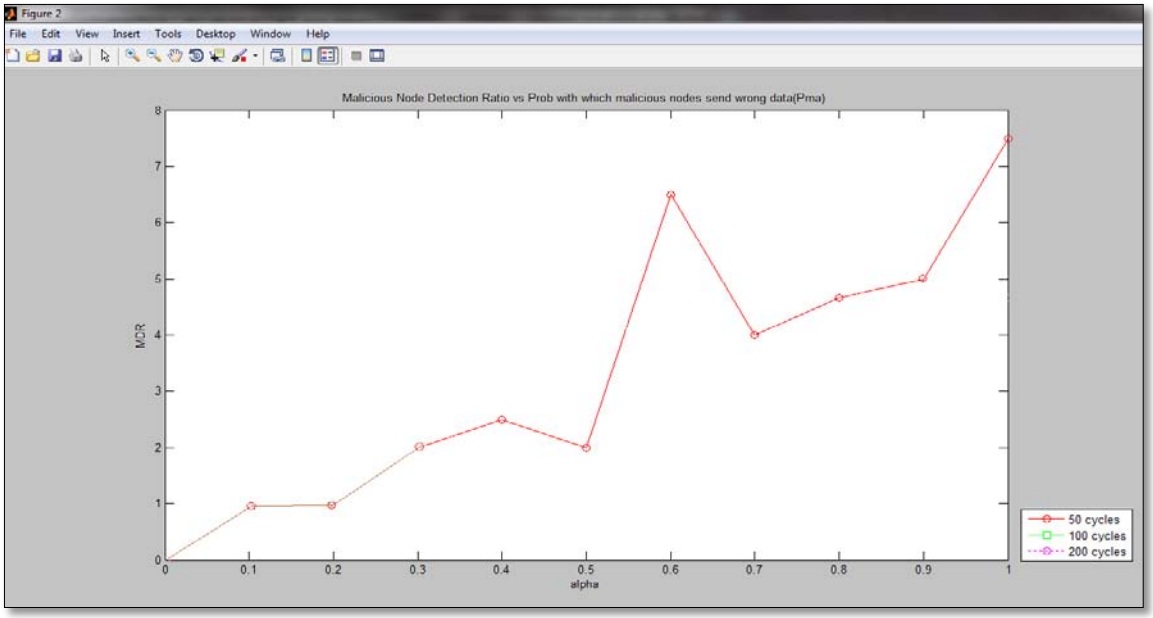


Figure 4.2.10 α vs MDR

In figure 4.2.10 we can observe that at low values of α there is misdetection where *malicious nodes are not detected* this happens because here is very less weight reduction and many nodes never pass the threshold θ below which they are detected as malicious. Due to high value of α, the weight of normal nodes is reduced drastically evn if they send wrong data due to temporary faults. Due to this there is high misdetection α are too high, where normal nodes as misdetected, as seen in the graph.

❖ *Variation of beta (β) of with MDR and MR:*

From figure 4.2.11 we can see that as β is increased MDR tends to stabilize and move towards its ideal value i.e.1. At low β there is no recovery for normal nodes when they send wrong data due to faults in network, but as β increases the performance improves.

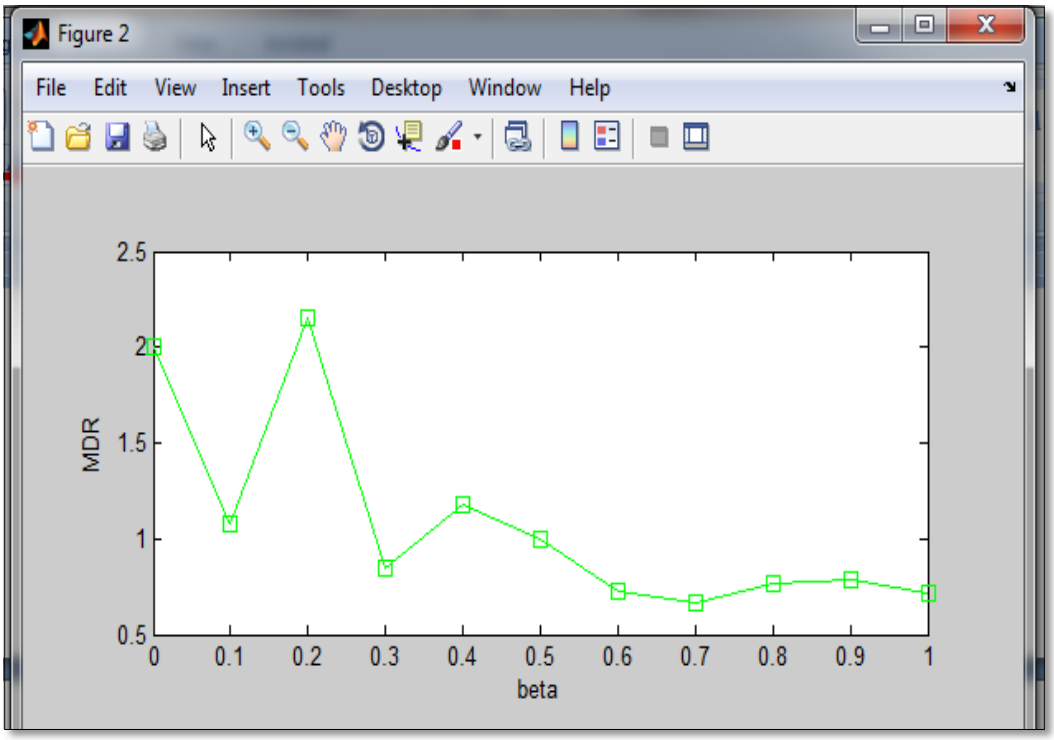


Figure 4.2.11 β vs MDR

CHAPTER 5

CONCLUSION

In this paper, we proposed a novel WTE based algorithm for the detection of malicious SNs in WSNs. Here, Weighted Trust Evaluation (WTE) is introduced for solving the Byzantine problem which occurs in Wireless Sensor Networks. The basic idea is that a weight representing the reliability of a node is assigned to each SN in the cluster under a FN. Since malicious nodes usually report falsified information to disrupt the network, if a node sends incorrect information, the FN gradually decreases the weight of the node and detect the node as a malicious node when its weight value becomes lower than a threshold. In addition, a weight recovery mechanism is incorporated in the algorithm to recover the weight of a node whose weight is accidentally decreased.

The network area is divided into square grids and malicious nodes are detected locally in a distributed manner. For a relatively small event region located across multiple adjacent grids, inter-grid communication is partially employed to enhance the event detection accuracy. Confidence levels (weights) are used to reflect the behavior of sensor nodes in reporting their readings in decision- making. Once the weights reach a predefined lower- bound, the corresponding nodes are logically isolated from the rest of the network. Thresholds are properly chosen to achieve high malicious node detection accuracy without sacrificing normal nodes.

Extensive simulation is performed using MATLAB. It can be seen that for low values of P_{ma} their high level of misdetection. Hence the algorithm is suitable for networks where is P_{ma} greater than 0.6. As for α, for very low values of α malicious nodes are not detected, since weight reduction is too low and also response time tends to become too high. High levels of α tend to misdetect normal nodes as malicious. Hence optimum $0.15 < \alpha < 0.4$ is found to be suitable based on the applications. WSN which is too sensitive may require a high α where even a single misdetection is not acceptable. Also β too shouls be maintained at an optimal level $0.15 < \alpha < 0.4$, It should always be seen that $\beta < \alpha$, so that recovery does not take away the efforts done by weight reduction. For sensitive WSN's β should be as small as possible.

Limitations:

- The proposed algorithm is based on the assumption that base stations (FNs and APs) are points of trust. In practice, if the adversary can gain control over the

base stations, it can launch any possible attacks against the WSN. Here we address only the Byzantine problem in WSN's.

- Another critical assumption is that the majority of the SNs are working properly. If the number of compromised nodes exceeds the number of normal nodes, normal nodes could be reported as malicious ones and malicious nodes are treated nice ones. If the compromised nodes are more than one-third of the total nodes this proposed is not applicable.
- No provisions or control is designed if the Forwarding Nodes (FN) itself is compromised during its operation as cluster head. We assume in our discussion when the cluster head is in action it cannot be compromised by any adversary.

The simulation experimental results have shown that the WTE algorithm is a promising solution to address the malicious nodes detection problem in WSNs. It achieves good scalability with reasonable detection delay, and is applicable to variant numbers of SNs deployed under the control of a FN, thus suitable to a flexible node deployment in WSNs. Note that the size of a cluster under a FN could be adjusted by setting more and less FNs for a WSN with fixed size. Essentially, it could be treated as a node-clustering problem. In both scalability and robustness simulations, the misdetection ratios in these cases could be largely reduced by introducing the weight recover mechanism.

In this paper we reported merely some preliminary results, which verified the correctness and effectiveness of our solution. More detailed analysis regarding the performance of our algorithm needs to be studied in the ongoing research for more questions to be answered.

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: a Survey", International journal on Computer Networks, vol. 38, no. 4, pp. 393-422, 2002.

[2] Ameer Ahmed,Mohamad Younis "A Survey on Clustering Algorithms for Wireless Sensor Networks", International journal on Computer Communications, vol. 30, no. 5, pp. 2826-2841, 2007.

[3]M. Ye, C. Li, G. Chen, J. Wu, "EECS: An Energy Efficient Cluster Scheme in Wireless Sensor Networks", in IEEE International Workshop on Strategies for Energy Efficiency in Ad Hoc and Sensor Networks, pp7–9, 2005.

[4]W.R. Heinzelman, A.P. Chandrakasan, H. Balakrishnan, "An application specific protocol architecture for wireless microsensor networks", IEEE Transactions on Wireless Communications, Vol. 1, No.4, pp. 660–670, 2002.

[5] Ganeriwal, L. K. Balzano, and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 3, pp. 1–37, May 2008.

[6] Z. Yao, D. Kim, and Y. Doh, "PLUS: parameterized and localized trust management scheme for sensor networks security," in *Proceedings of IEEE International Conference on Mobile Ad-hoc Sensor networks*, pp. 437–446, 2006.

[7] E. Aivaloglou and S. Gritzalis, "Hybrid trust and reputation management for sensor networks," *Wireless Networks*, vol. 16, no. 5, pp. 1493–1510, July 2010 \

[8]S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," *ACM Transactions on Sensor Networks*, vol. 4, no. 3, pp. 1–37, May 2008.

[9] Z. Yao, D. Kim, and Y. Doh, "PLUS: parameterized and localized trust management scheme for sensor networks security," in *Proceedings of IEEE International Conference on Mobile Ad-hoc Sensor networks*, pp. 437–446, 2006.

[10] E. Aivaloglou and S. Gritzalis, "Hybrid trust and reputation management for sensor networks," *Wireless Networks*, vol. 16, no. 5, pp. 1493–1510, July 2010

[11]S. Bo, G. Sui-Xiang, C. Rui, and H. Fei, "Algorithms for balancing energy consumption in wireless sensor networks" in Proceeding of the1st ACM international workshop on Foundations of wireless ad hoc and sensor networking and computing, vol.3,pp. 53–60,2008.

[12]Bijan Kumar Debroy, Muhammad Sheikh Sadi, Md. Al Imran"An Efficient Approach

to Select Cluster Head in Wireless Sensor Network", International Journal of Communications, vol. 6, no. 7, pp 529-539, 2011.

[13]Tan Ming-hao, Yu Ren-lai, Li Shu-jiang and Wang Xiang-dong "Interference Multipath Routing Protocol with Load Balancing in WSN Considering",IEEE 6th Conference on Industrial Electronics and Applications ,pp 1062-1067,2011.

[14]Sohail Jabbar , Ayesha Ejaz Butt, Najam us Sahar and Abid Ali Minhas "Threshold Based Load Balancing protocol for Energy Efficient Routing in WSN" 13th International Conference on Advanced Communication Technology ,no.5,pp 196 - 201 , 2011

[15]L. Capra and M. Musolesi, "Autonomic trust prediction for pervasive systems," in *Proceedings of International Conference on Advanced Information and Network Applications*, pp. 1–5, 2006.