# EFFICIENT STRATEGY ON DETECTION OF MALICIOUS NODES IN WIRELESS SENSOR NETWORK

**A DISSERTATION**

***Submitted in partial fulfillment of the requirements for the award of the degree of***

MASTER OF TECHNOLOGY

*in*

COMPUTER SCIENCE AND ENGINEERING

*By*

**MANDEEP SINGH**

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)

JUNE, 2008

# CANDIDATE'S DECLARATION

I hereby declare that the work, which is being in this dissertation report, entitled **"Efficient Strategy On Detection Of Malicious Nodes In Wireless Sensor Network"**, is being submitted in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Computer Engineering**, in the Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee is an authentic record of my own work, carried out from June 2007 to June 2008, under guidance and supervision of **Dr. Kuldip Singh**, Professor and **Dr. Manoj Mishra,** Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee.

The results embodied in this dissertation have not submitted for the award of any other Degree or Diploma.

**Date** 30\06\08

**Place: Roorkee**

Mandeep Singh

**(Mandeep Singh)**

# CERTIFICATE

This is to certify that the statement made by the candidate is correct to the best of my knowledge and belief.

Manoj Misra

**Dr. Kuldip Singh**
**Professor**

**Dr. Manoj Mishra**
**Professor**

# ACKNOWLEDGEMENTS

I express my deep sense of gratitude to my guide **Dr. Kuldip Singh** Professor, Department of Electronics & Computer Engineering and **Dr. Manoj Mishra** Professor, Department of Electronics & Computer Engineering, for there keen interest, there able guidance, regular source of encouragement and assistance throughout this work. I am extremely grateful for there timely suggestions and faith in me. I deem it my privilege to have carried out the work under there guidance.

I am also grateful to the staff of Information Security Laboratory and the departmental library of Department of Electronics & Computer Engineering for their sincere co-operation.

Mandeep Singh

# ABSTRACT

In wireless sensor networks a large number of distributed sensors collaborate to deliver information to the sinks. Such scenario assumes trust between sensor nodes. However, sensors nodes may fail or may be compromised. These compromised nodes take the participation in the routing. They have the ability to disconnect a wireless sensor network from its central base station. Henceforth, security is a critical challenge for creating robust and reliable Wireless sensor networks.

For detection of malicious nodes in wireless sensor networks, we could use REWARD (Receive, Watch, and Redirect) detection scheme. This methodology detects malicious nodes which always drops packets received. We could observe that in the implementation of the scheme, good amount of energy would be consumed and moreover the energy at the sensor node is limited. Therefore, we propose and implement a modified scheme of REWARD by minimizing the number of packets sent by the intermediate nodes in the network. Hence, by this strategy we optimize the energy at the node and also reduce the overhead of the Wireless Sensor network.

# CONTENTS

## LIST OF FIGURES

# CHAPTER 1 INTRODUCTION AND STATEMENT OF PROBLEM

## 1.1 Introduction

Wireless sensor networks are provisioned to consist of large number of tiny, limited capacity nodes and limited battery and also known for their flexibility, cost effectiveness, and ease in deployment. They are being widely used for various monitoring systems, data collection, and process control applications etc [1].

Due to the growing concern of wireless sensor networks in Military applications [2] and transportation monitoring system [3], they are deployed at various hostile environments. One of the challenges faced in these deployment are compromised or malicious nodes. Existence of malicious nodes in wireless networks may be quite harmful and may disrupt the network operations by injecting false information or by not cooperating in tasks such as packet forwarding. Such malicious node activity can severely affect the network operations. The malicious node may be chosen on the routing path from the sink(s) to many sensor nodes. We call such paths infected routes. So looking over these aspect securities is a critical challenge for creating robust and reliable wireless sensor networks.

In wireless sensor network security solutions are generally grouped into two main cate1gories: prevention based techniques and detection based techniques. Prevention techniques, such as encryption and authentication, are often the first line of defense against an attacker. Detection based techniques aim at identifying and excluding the attacker after prevention based techniques fail. The detection techniques are categorized into signature and misbehaving nodes detection. Signature detection techniques match the known attack profiles with the current events whereas

misbehaving nodes detection scheme detects based on significant deviations from the established normal system operation on nodes.
Prevention based security techniques are less effective because of the shared broadcast medium and resource-limited network elements. In multihop wireless networks, nodes forward packets for other nodes and this requires additional trust requirements which increase the complexity of prevention based security solutions. The security algorithms for wireless sensor network have to be designed with the limited computational power, limited memory and limited battery life of sensors in mind. Securing the wireless sensor network with conventional prevention and detection techniques is difficult due to the scalability problems, computation, communication and storage overhead associated with these methods.

One of the detection schemes is REWARD scheme[4] for detection of malicious nodes in wireless sensor networks that isolates malicious nodes and establishing trust routing in wireless sensor networks. REWARD scheme identifies and isolate of malicious nodes using trust routing and avoidance of insecure locations. The two main components of architecture are:
(i) Trust routing and
(ii) Malicious node discovery and isolation.

In the study of the REWARD scheme (which we discuss in detail in later chapters), we could observe that it consumes good amount of energy, moreover the energy at the sensor node is limited. Therefore, we propose and implement a modified scheme of REWARD, in which we send ACK packets instead of sending the same packet as in case of REWARD scheme, with using the identifier. We further minimize the number of ACK packets sent by the intermediate nodes in the network. Hence, we optimize the energy at the node and also reduce the overhead of the Wireless Sensor network. We also argue that our algorithm aptly suits for both cases of detecting single as well multiple malicious nodes in the network.

## 1.2 Problem Statement

This thesis aims at detecting malicious nodes in wireless sensor networks. Malicious nodes behave like normal nodes most of the time but drop packets which are forwarded to these nodes. We try to achieve an efficient strategy to detect these nodes with minimum overhead on network in terms of traffic and reduce transmission power consumption of nodes in network.

## 1.3 Overall objectives

- The main objective of our research is to develop efficient detection model in terms of energy, to secure the networks from malicious attacks (Malicious node discovery and isolation).
- Trust routing and
- To reduce the overall communication overhead and make REWARD scheme energy efficient.

## 1.4 Organization of the Report

This report comprises of six chapters including that introduces the topic and states the problem.

Chapter 2 gives an overview of the Wireless sensor networks and attacks on wireless sensor networks. Also, taxonomy of the existing detection approach is discussed in brief.

Chapter 3 gives an overview of the REWARD scheme used for detection of malicious node in wireless sensor networks. We discus limitations of REWARD scheme and propose solution to overcome some of them.

Chapter 4 describes the system design that includes the system components and the simulation model. The implementation details are also charted out in terms of topology used for simulation purposes, procedures and simulation parameters.

Chapter 5 discusses the simulation results of this approach in detecting malicious node.

Chapter 6 concludes the work and gives the directions for future work.

# CHAPTER 2
# BACKGROUND AND LITERATURE REVIEW

Wireless Sensor Networks (WSNs) are known for their flexibility, cost effectiveness, and ease in deployment; as a result they are being widely used for various monitoring systems, data collection, and process control applications, etc [1]. In spite of these advantages, wireless sensor networks are highly prone to software and hardware failures, and security threats and intrusions attacks. These attacks are very easy to implement and can significantly disrupt the functioning of sensor networks.

## 2.1 Wireless Sensor Networks

Advancement in technologies has enabled the development of multi-functional tiny devices known as sensor nodes [5]. These nodes consist of sensing, data processing and communicating components. Network of sensor nodes is known as wireless sensor network (WSN). One of the primary purposes of WSN is to gather information and send it to the base station for accumulation and/or analysis. Wireless sensor network typically consist of sensor nodes which sense data from their ambience, and pass it on to a centralized controlling and data collecting identity called base station. Base stations are powerful devices with a large storage capacity to store incoming data. They generally provide gateway functionality to another network, or an access point for human interface. A base station may have an unlimited power supply and high bandwidth links for communicating with other base stations. In contrast, wireless sensors nodes are constrained to use low power, low bandwidth, and short range links. Therefore, in most of the sensor networks, it is necessary to use a multi-hop routing protocol where the sensors try to reach the nearest base station by establishing communication links between themselves [6]. A sensor network application can be configured to enable the sensors to send data at regular intervals and/or to respond to a query generated by a base station. WSN can be used in multiple applications in different spheres of life like monitoring applications, acoustic detection, seismic detection, military surveillance, inventory tracking etc.

## 2.2 Components of Wireless Sensor Network

The major components of a typical sensor network are: sensor nodes, the sensor field, the sink and the task manager, as shown in Figure 2.1. It proceeds to define these components in further detail:

A sensor field can be considered as the area in which the nodes are placed i.e. the area in which we expect a particular phenomenon to occur.

A Sensors nodes [5] or motes are the heart of the network. They are in charge of collecting data and routing this information back to a sink.

A sink is a sensor node with the specific task of receiving, processing and storing data from the other sensor nodes. They serve to reduce the total number of messages that need to be sent, hence reducing the overall energy requirements of the network. Such points are usually assigned dynamically by the network. Regular nodes can also be considered as sinks if they delay outgoing messages until they have aggregated enough sensed information. For this reason sinks are also known as data aggregation points.

The task manager or base station is centralized point of control within the network, which extracts information from the network and disseminates control information back into the network. It also serves as a gateway to other networks, a powerful data processing/storage centre and an access point for a human interface. Hardware wise the base station is either a laptop or a workstation. Data is streamed to these workstations either via the internet, wireless channels, satellite etc [6].

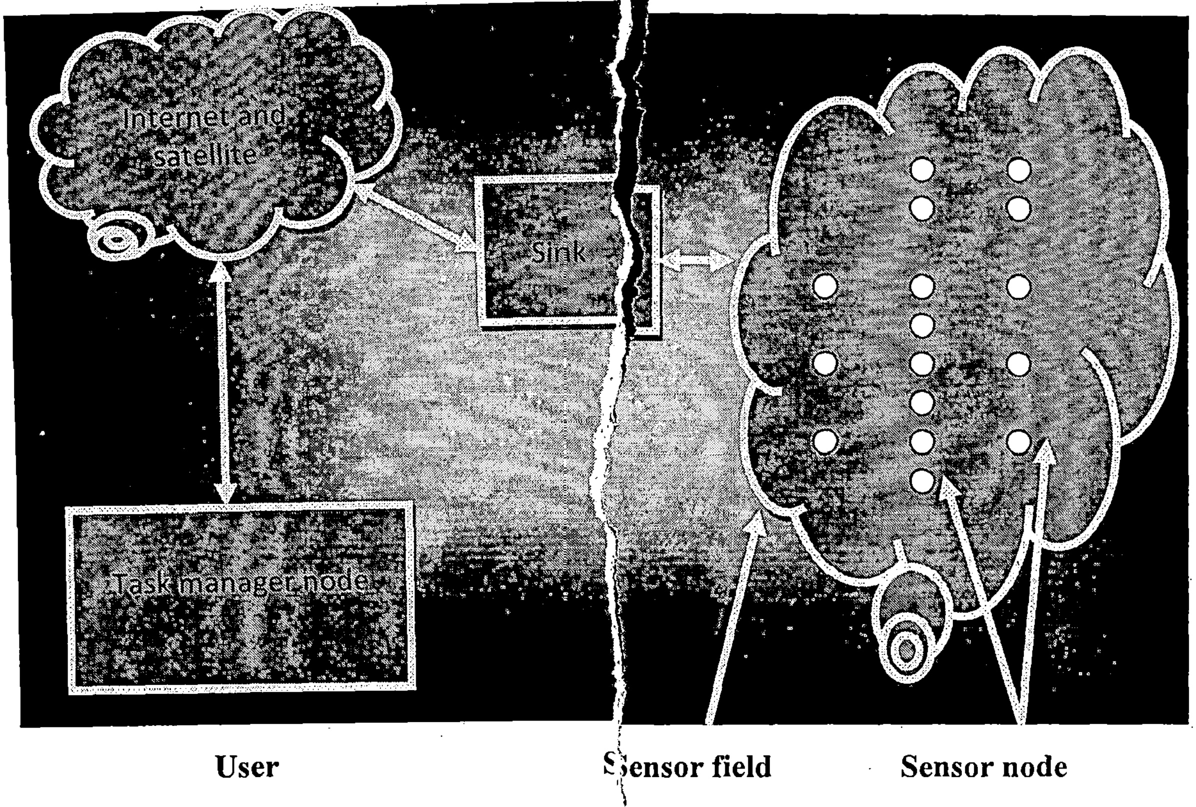


Figure 2.1: Wireless Sensor Network Communication Structure

## 2.3 Communication Architecture

Generally, the sensor nodes communicate using RF, so broadcast is the fundamental communication primitive. In the sensor applications developed so far, the communication patterns within the network fall into the following categories:

- Node to base station communication, e.g. sensor readings, specific alerts.
- Base station to node communication, e.g. specific requests, key updations.
- Base station to all nodes, e.g. routing beacons, queries or reprogramming of the entire network.
- Communication amongst a defined cluster of nodes (say, a node and all its neighbors). Clustering can reduce the total number of messages sent and thus save energy [7, 8, 9] by using in-network processing techniques such as ***data aggregation*** [10,11] (an aggregation point can collect sensor readings from surrounding nodes and forward a single message representing an aggregate of

the values) and ***passive participation*** (a node that overhears a neighboring sensor node transmitting the same reading as its own current reading can elect to not transmit the same).

## 2.4 Important characteristics of wireless sensor network

The terminology wireless sensor network is referred to a heterogeneous system combining tiny sensors [12]. The important characteristics are as follows:

1. This huge sensor sea consists of innumerable low-power and low-cost wireless nodes. Their primary job is to monitor and report the conditions in the deployed environment.

2. The sensor networks have limited space and hence these hefty cryptographic protocols cannot reside on them. On the contrary these limitations can work in our favor as the adversaries have to also match the capabilities of the sensors, for e.g. even they are limited by the bandwidth restrictions (number of packets per second), hence they can‘t eavesdrop by bombarding it with fake messages.

3. Energy is the most important element of concern, it can be truly said that energy saved is energy earned. Every iota of energy used brings the wireless nodes closer to their end.

4. The sensor network communicates in a broadcast medium; hence this wireless media is very unsafe. Also the adversaries can use this to flood the network and drain the sensors energy. It can be typified by the term pursuit evasion game [12] as there is a group of pursuers who attempt to capture the evaders.

## 2.5 Challenges routing schemes for wireless sensor network

There are a number of challenges [6] that one must address when devising routing schemes for WSNs.

First, wireless sensor nodes still have limited memory storage and reduced processing power, therefore it is resource-consuming to maintain routing information proactively. Consequently, it is essential to quickly find routes in an as-needed manner. These routes should be easily forgotten/discarded yet subsequently quickly recomputed.

Second, sensor nodes either run on batteries or harvest for energy and once deployed they are unattended and expected to operate for a long time. Their energy resources are limited or unreplenishable. Thus, it is crucial to use it efficiently to extend the network lifetime and service the application.

Third, wireless sensor nodes have short communication ranges, therefore routing decisions ought to rely on localized information from immediate neighbors.

Fourth, despite the reduced human intervention on the sensor devices, the application requires that the network performs efficiently and intelligently. Therefore, it is proper to control and program the sensors' tasks (routing, data fusion, etc.) remotely, i.e. general directions are sent to the sensors so they can adapt to the situation and decide how to route based on the conditions of the environment and the nodes' status.

Fifth, the majority of existing WSN routing protocols have built-in routing objectives and strategies. It is not possible to change the routing objective (e.g. shortest distance, minimum number of hops, minimum energy consumed) once the protocol is implemented on the nodes. In addition, the routing destination is usually identified via an address, but sometimes, destinations need not to be defined through addresses but based on their attributes, their sensor readings or their geographical locations. To face this inflexibility, we decouple the routing objective from the routing strategy and the routing destination, and change the routing objective when needed. We use the decoupling approach proposed in [6].

Finally, WSN applications raise new routing challenges and more complex communication requirements and it is imperative sometimes to satisfy and make routing decisions over multiple (possibly conflicting) objectives. To address these issues, we propose a routing framework for WSNs that is adaptive, constraint-based, and multi-objective. We use real-time search as the initial routing strategy and will explore other methods suitable to localized learning and conflicting objectives like fuzzy logic and Bayesian inference.

## 2.6 Kind of attacks in wireless sensor network

The various kinds of attacks [13] are as follows:

I. Spoofed: Spoofed, altered or replayed routing information: This is the most direct attack. By spoofing, altering or replaying routing information the attacker can complicate the network by creating routing loops, attracting or repelling traffic, generating false error messages, shortening or extending source routes or partitioning the network.

II. Selective Forwarding: In such an attack the adversary includes himself/herself in a data flow path of interest. Then the attacker may choose not to forward certain packets and drop them causing a sort of black hole. A variation of this attack is when the adversary only drops packets coming from a specific source whilst reliably forwarding other packets. Such attacks are much harder to detect than black hole attacks.

III. Sinkhole Attacks: The goal of a sinkhole attack is to lure traffic to a malicious part of the network. Such attacks are usually the launching block for other attacks such as selective forwarding. Sinkholes work by making a compromised node attractive to its neighbors. This is done by advertising high quality routes i.e low latency routes. Fooled neighbors will then forward all their data destined to the base station to the lying node. Sensor networks are susceptible to these attacks due to their multi hop nature and the specialized communication patterns they use.

IV. Sybil Attack: The Sybil attack targets fault tolerant schemes such as distributed storage, multipath routing and topology maintenance. This is done by having a malicious node present multiple identities to the network. This attack is especially confusing to geographic routing protocols as the adversary appears to be in multiple locations at once.

V. Wormholes Attack: In these attacks the adversary tunnels messages received in one part of the network over a low latency link, to another part of the network where the messages are then replayed. The simplest occurrence of this attack is to have a malicious node forwarding data between two legitimate nodes. Wormholes often convince distant nodes that they are neighbors, leading to quick exhaustion of their energy resources. An attacker close to the base station can completely disrupt routing by creating well positioned wormholes that convince nodes multiple hops from the base station that they are only a couple of hops away through the wormhole. When this attack is coupled with selective forwarding and the Sybil attack it is very difficult to detect.

VI. Hello flood attacks: In many routing protocols, nodes broadcast hello messages to announce their presence to their neighbors. A node receiving such a message can assume that the node that sent the message is within it's range. An attacker with a high powered antenna can convince every node in the network that it is their neighbor. If the attacker also advertises a high quality route it can get every node to forward data to it. Nodes at a large distance from the attacker will be sending their messages into oblivion leaving the network in a state of confusion. This attack can also be thought of as a type of broadcast wormhole. Routing protocols dependant on localized information are extremely vulnerable to such attacks.

## 2.7 Security requirements in wireless sensor network

Sensor networks are used in a number of domains that handle sensitive information. However as is it obvious due to the nature of wireless network that the data is freely available in air. The loss of confidentiality and integrity and the threat of denial of service (DoS) attacks are risks typically associated with wireless communications.

Unauthorized users may gain access to agency systems and information, corrupt the agency's data, consume network bandwidth, degrade network performance, and launch attacks that prevent authorized users from accessing the network, or use agency resources to launch attacks on other networks. There are many security challenges in WSN [14,15]. Some of them are as follows: Confidentiality, Authentication, Freshness, and Integrity etc. Because of the limitations due to battery life, nodes are built with power conservation in mind, and generally spend large amounts of time in a low-power "sleep" mode or processing the sensor data.

### 2.7.1 Confidentiality

Confidentiality requirement is needed to ensure that sensitive information is well protected. The confidentiality objective is required in sensors' environment to protect information traveling between the sensors nodes of the network, since an attacker having the appropriate equipment may eavesdrop on the communication. By eavesdropping, the attacker could overhear critical information such as sensing data and routing information. Based on the sensitivity of the data stolen, an attacker may cause severe damage. Furthermore, by stealing routing information the attacker could introduce his own malicious nodes into the network in an attempt to overhear the entire communication. Malicious nodes are a big threat to confidentiality objective since the attacker could steal critical data stored on nodes such as cryptographic keys that are used to encrypt the communication.

### 2.7.2 Authentication

As in conventional systems, authentication techniques verify the identity of the participants in a communication. In the case of wireless sensor networks, it is essential for each sensor node and base station to have the ability to verify that the data received was really send by a trusted sender and not by an attacker. If false data are supplied into the network, then the behavior of the network could not be predicted and most of times will not outcome as expected. Authentication objective is essential to be achieved when clustering of nodes is performed.

Clustering involves grouping nodes based on some attribute such as their location, sensing data etc and that each cluster usually has a cluster head that is the node that joins its cluster with the rest of the sensor network (meaning that the communication among different clusters is performed through the cluster heads). In these cases, where clustering is required, there are two authentication situations which should be investigated; first it is critical to ensure that the nodes contained in each cluster will exchange data only with the authorized nodes contained and which are trusted by the specified cluster (based on some authentication protocol). Otherwise, if nodes within a cluster receive data from nodes that are not trusted within the current community of nodes and further process it, then the expected data from that cluster will be based on false data and may cause damage. The second authentication situation involves the communication between the cluster heads of each cluster; communication must be established only with cluster heads that can prove their identity. No malicious node should be able to masquerade as a cluster head and communicate with a legitimate cluster head, sending it false data or either compromising exchanged data.

### 2.7.3 Integrity

Moving on to the integrity objective, there is the danger that information could be altered when exchanged over insecure networks. Lack of integrity could result in many problems since the consequences of using inaccurate information could be disastrous, for example for the healthcare sector where lives are endangered. Integrity controls must be implemented to ensure that information will not be altered in any unexpected way. Many sensor applications such as pollution and healthcare monitoring rely on the integrity of the information to function with accurate outcomes; it is unacceptable to measure the magnitude of the pollution caused by chemicals waste and find out later on that the information provided was improperly altered by the factory that was located near by the monitored lake. Therefore, there is urgent need to make sure that information is traveling from one end to the other without being intercepted and modified in the process.

### 2.7.4 Freshness

One of the many attacks launched against sensor networks is the message replay attack where an adversary may capture messages exchanged between nodes and replay them later to cause confusion to the network. Data freshness objective ensures that messages are fresh, meaning that they obey in a message ordering and have not been reused. To achieve freshness, network protocols must be designed in a way to identify duplicate packets and discard them preventing potential mix-up.

## 2.8 Related Work

Huang *et al.* [16] proposed a mechanism that needs separate monitoring nodes, specifically one monitor per cluster (nodes that are in one-hop range form a cluster). The approach requires monitors to be active. If there is one monitor per cluster, the monitor does most of the work. In WSNs, there is a risk that monitor nodes run out of energy before the network does or before the network gets partitioned. This contradicts one of the main goals of prolonging WSN lifetime and keeping WSN connected as much as possible (since battery replacement is a very costly or unavailable alternative).. All the above approaches monitor individual nodes all the time. Continuous monitoring of each and every node is not feasible for resource-constrained WSNs especially when extending lifetime is the main goal in the design of WSNs. Our proposed solution, DPDSN avoids continuous monitoring of every node.

Du *et al.* [17] had presented a scheme to detect malicious beacon nodes. They reasoned that it would be difficult for a compromised beacon node to get away with sending beacon signals with the wrong location information. This is because the location and beacon signal sent by the malicious beacon node will both have to be falsified. Beacon nodes in the network are given a set of node ID's and keys that allows them to communicate with the other beacon nodes of the network while appearing to be a non-beacon node. Compromised nodes are detected when a valid beacon node gets a beacon signal from a malicious beacon node whose estimated location based off the beacon signal is different from the location given by the beacon signal. Attacks using locally replayed beacon signals are discovered since it is difficult for the compromised beacon node to achieve the expected round trip time for communication between neighboring nodes.

Buchegger *et al.* [18] proposed a mechanism that detects misbehaving nodes by means of observations or reports about several types of attacks. This allows nodes to find routes around misbehaving nodes and to isolate them from the network. Nodes have a monitor for observations, reputation records for first-hand observations and trusted second-hand reports, trust records to control trust given to received warnings, and a path manager to adapt their behavior according to reputation of other nodes. This approach involves continuous monitoring similar to Marti's approach and collecting information about intrusion detections at other places in the network. The overhead is prohibitive for WSNs.

Michiardi *et al.* [19] proposed a collaborative reputation mechanism that has a watchdog component. However, it is complemented by a reputation mechanism that differentiates between subjective reputation (observations), indirect reputation (positive reports by others), and functional reputation (task specific behavior). They are weighted for a combined reputation value used to make decisions about cooperation with or gradual isolation of a node. This approach involves continuous monitoring and collecting information about intrusion detections at other places in the network for specific functions. The overhead is too high for WSNs.

# CHAPTER 3
# PROPOSED DETECTION SCHEME

## 3.1 Review of REWARD Detection scheme

REWARD (Receive, Watch, and Redirect) detection scheme is a security service algorithm for detection of malicious node which is always drop packets received. The REWARD security service provides alternative paths for routing in an attempt to avoid misbehaving nodes and regions of detected attacks. The security servers are nodes that keep records of malicious node and modify forwarding of packets to bypass insecure nodes or malicious node. Wireless routing protocols are used in REWARD detection scheme for path discovery. Zdravko Karakehayov [4] is used AODV to discover route form source to destination. In REWARD, every node listen to its neighbor nodes transmissions to detect malicious nodes in wireless sensor network. When a node receives a packet it selects a neighbor node form routing table for forwarding packet to selected neighbor node. A common approach is to choose node closest to the destination.

REWARD utilizes two types of broadcast messages when some nodes drop a packet, called alarm messages. One of the messages is broadcast by destination known as MISS (Material for Intersection of Suspicious Sets) message and other messages are by intermediate nodes known as SAMPDA (Suspicious Area, Mark a Packet Drop Attack) message [4, 20].

- MISS - When the destination receives the query it reply its location to source and waits for a packet. If the packet does not arrive within the time then destination broadcast MISS message. The destination copies the list of all involved nodes in query from source to destination into the MISS message. Since the reason for not receiving the packet is most likely a malicious node in the path. All nodes listed in MISS message are under suspicion.

- SAMPDA – After route discovery, source node forward a packet to next node in the path. Each node in the path tunes the transmit power to reach packet to its both immediate neighbors, one forward node ($N_i$) and one backward node ($N_{i-1}$). Each node in a path transmit packet and wait for reply if the packet is forwarded properly. If any node in a path acts as a malicious node then its neighbor node will broadcast SAMPDA message.

Source will forward packet to its neighbor node in the discovered path. Each node tunes the transmit power to reach both immediate neighbors [21]. The nodes transmit packets and watch if the packets are forwarded. If a malicious node does not act as a forwarder, the previous node in the path will broadcast a SAMPDA message. An example of single malicious node is shown in Figure 3.1.

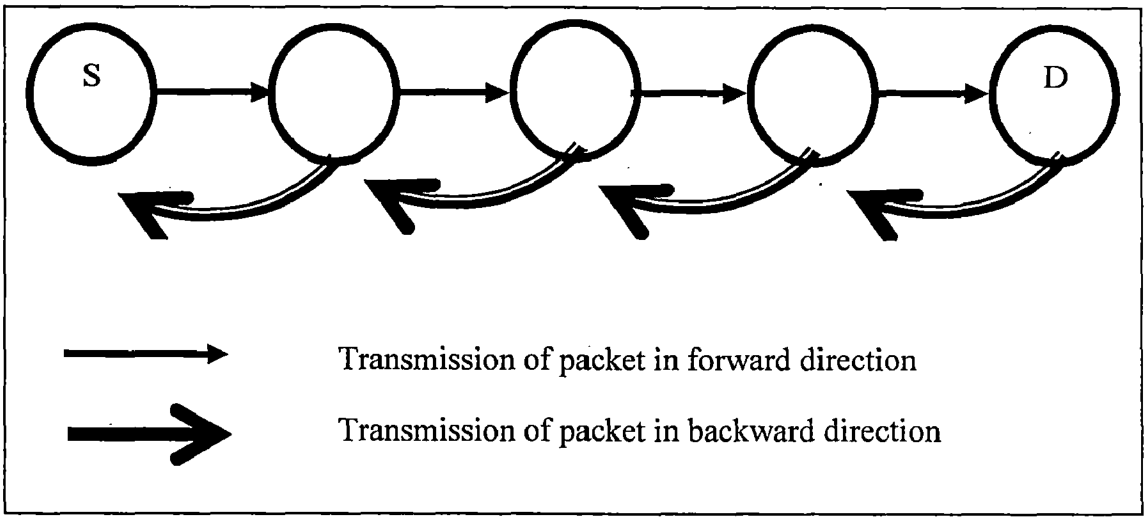

S
D
Transmission of packet in forward direction
Transmission of packet in backward direction

Figure 3.1 Transmissions must be received by both neighbors

## Algorithm: REWARD detection scheme

```
Algorithm:
    1. Discover path from source to destination using routing protocol.
    2. Destination will reply its location through the discovered path and wait for a packet for certain period of time. If time elapse destination will broadcast MISS message.
    3. Source will forward packet to its neighbor node in the discovered path.
    4. Until node is not equal to destination node
            a. Node will forward packet to its neighbor node in the path.
            b. Wait for a reply from neighbor node for certain period of time.
            c. If reply =0
                    Broadcast SAMPDA message
               End
```

If source node receives MISS or SAMPDA messages then source will find out a new path to the destination excluding malicious node in the path using routing protocol.

REWARD is a scalable method capable of waging counter attacks against a different number of malicious nodes. When a team of malicious nodes would attempted to drop packet. In this case the algorithm requires the nodes to listen for two retransmissions. Since each transmission must be received by two nodes backward in the path, the transmit power must be increased. The first malicious node forwards the packet using the required transmit power to deceive two nodes backward. The second malicious node drops the packet; however the attack is detected by the last node before the malicious node. An example of team of malicious node is shown in Figure 3.2.

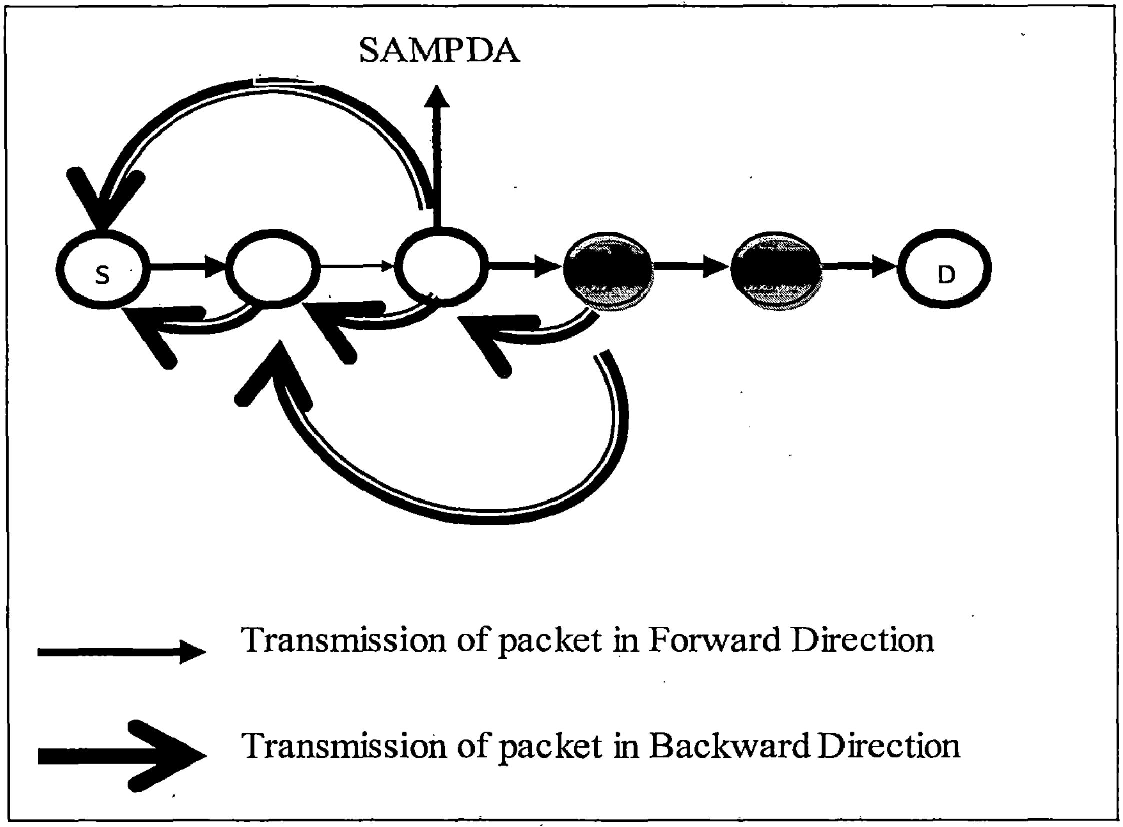

SAMPDA
S
D
Transmission of packet in Forward Direction
Transmission of packet in Backward Direction

Figure 3.2 REWARD detection against two malicious nodes

## 3.2 Limitations of REWARD

The following are the limitations of REWARD

- The destination broadcasts MISS message if packet does not arrive within a specific period of time. The destination copies the list of all involving nodes from the query to the MISS message and broadcast MISS message which indicates the nodes containing in the path are malicious node, even if the packet is dropped by a single node in the path. As all nodes listed in a MISS message are declared as malicious nodes by the destination so that next time these nodes are not allowed to participate in new path from source to destination.

- Consuming power is a most important factor in the wireless sensor network. In REWARD each node transmits packets to both immediate neighbors (Forward and backward), this consumes lot of power.

- If more than two malicious nodes are present in the network then REWARD algorithm requires the nodes to listen for two retransmissions as shown in Figure 3.2. The transmit power must be increased.

- Transmitting packets to both immediate neighbor nodes leads to traffic overhead in a network.

We propose solution to two retransmission problem and minimizing overhead and power consumption of forwarding packets by each node to its forward and backward neighbor in REWARD. Instead of each node transmit packet to both immediate neighbors (Forward and Backward direction), nodes simply forward packets to neighbor (Forward direction) and only specific number of nodes transmits acknowledgement backward direction. This reduces the overhead and consumes less energy. We propose to modify the REWARD algorithm to overcome the problems stated above in following section.

## 3.3 PROPOSED DETECTION SCHEME

In this section, we modify the REWARD detection scheme. In proposed detection scheme, each intermediate node along the forwarding path is in charge of detecting malicious nodes. If an intermediate node detects the misbehavior of nodes, it will broadcast an alarm packet and deliver it to the source node through multiple hops.

### 3.3.1 Assumptions

The following assumptions are appropriate to use of the proposed detection scheme.

First, we assume that routing protocol such as AODV [22] have been implemented in sensor nodes. Our algorithm will function over this protocol.

Second, although the routing layer of WSNs is threatened by various attacks, here we are considering only packet drop attack by malicious node. We don't consider the detection of link-layer jamming attacks [23], which are also able to cause packet loss.

Third, we assume that network has no channel error.

### 3.3.2 Detection Process

The data packet generated by the source node age routed using AODV protocol to the destination. The packet will be forwarded toward the base station hop-by-hop. The initial ACK_Cnt is set to ACK_Span, which is a predefined metric. When each intermediate node receives the data packet, it will decrease the ACK_Cnt by one, or resets ACK_Cnt to its initial value ACK_Span if ACK_ Cnt equals to 0 already, and forwards the data packet to the next node. Meanwhile, if the node finds ACK_Cnt equal to 0, it generates an ACK packet. The TTL in the ACK packet is initially set to ACK_TTL, which is also a predefined metric. The ACK for the data packet would be issued by intermediate nodes when the ACK_Cnt equals to the source node in the case when ACK_Cnt is made zero for the first time and to the intermediate nodes in the network in all other case figure 3.3 illustrates the scheme. We actually save on overhead and energy by minimizing the number of ACK sent over the network the scheme in comparison to REWARD. The ACK packet will traverse multiple hops until TTL is decreased to 0. When ACK_TTL equals to zero, the packet is stopped forwarding. The ACK packet is following the same path as traversed by the previous data packet but in the opposite direction. We call the nodes which are required to send out ACK packets ACK nodes, such as nodes N3, N6, and N9 in Figure 3.4.

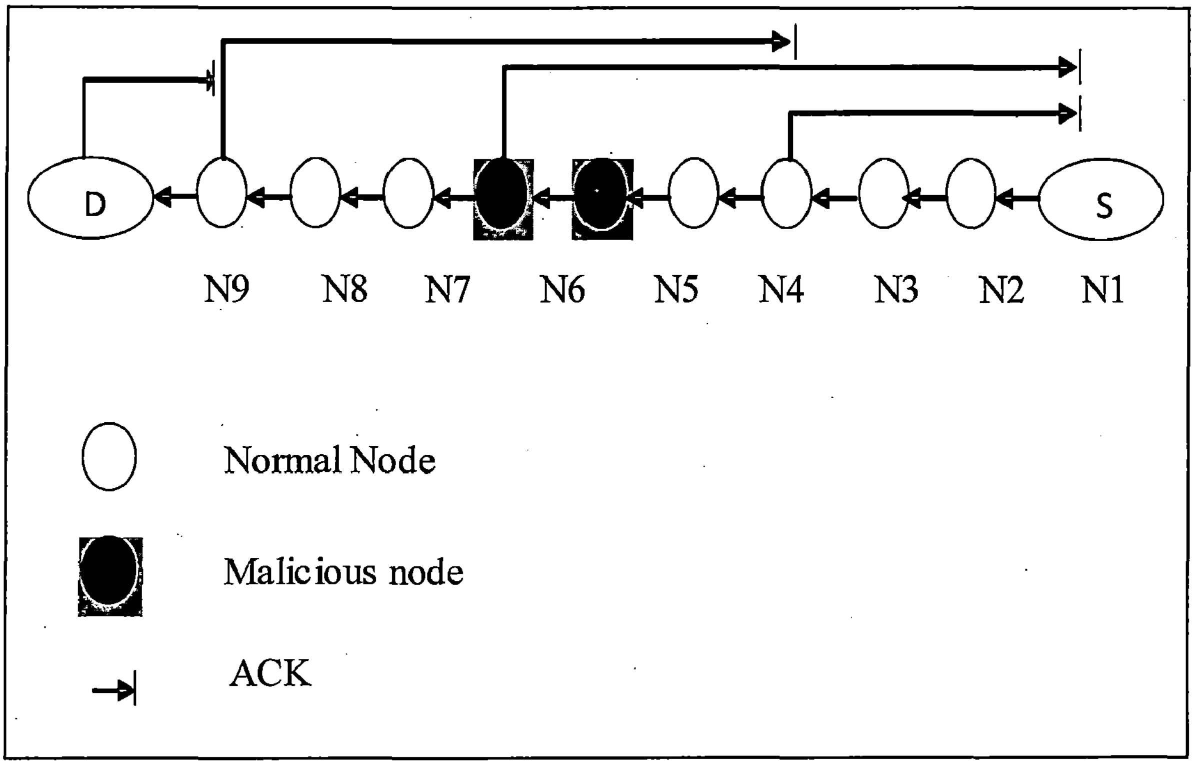


Figure: 3.3 Proposed scheme with ACK_Span=3, ACK_ TTL=6

After an intermediate node forwards a data packet to the neighbor node, it waits for ACK packets which will be returned by the neighbor. If ACK packet is not returned, the node suspects that the previous data packet might have been dropped by a malicious node. The intermediate node then generates an alarm packet. DstID in the alarm packet is the source node id. The node chooses the next node to be the malicious node and set it in the malicious Node ID field in the alarm packet. The alarm packet then is forwarded through multiple hops to the source node. Source node stores that node into blacklist and again find path using routing protocol. New path exclude those node that are in blacklist.

It is possible for a malicious node to fabricate an alarm packet but this would have only a limited effect because the malicious node can only set the next node as the suspicious node. The source node will regard both the suspicious node specified in the alarm packet and the node which generates the alarm packet as malicious nodes.

Figure 3.4 provides an example of the operation of our detection mechanism. Suppose that node *N5* and *N6* are compromised nodes, and node *N5* drops a data packet coming from the source node. First we consider node *N2*. After forwarding the data packet to node *N3*, node *N2* waits for ACK packets. Suppose that node *N2* receives an ACK packet from node *N3* but does not receive an ACK from *N6* within time. Node *N2* will set node *N3* as the suspect node in its alarm packet. The alarm packet will be forwarded through multiple hops to the source node. Next we consider node *N4*. Node *N4* receives no ACK packets. It sets the next node, node *N5*, as the suspect node. Node N4 also generates an alarm packet and sends it to the source node. Finally, we consider node *N6*, which fabricates an alarm packet which says node *N7* is the suspect node. The alarm packet is also delivered to the source node. In this way, the source node will receive 3 alarm packets for the same data packet. However, because it can be sure that the data packet did arrive at node *N4*, it's easy for the source node to remove the false alarms from nodes *N2* and *N4*. The source node finally concludes nodes *N6* and *N7* as the malicious nodes.

## Algorithm: Proposed detection scheme

```
Algorithm:
Set ACK_Span=3 and ACK_ TTL=6
    1. Discover path from source to destination using routing protocol.
    2. Destination will reply its location through the discovered path and wait for
       a packet for certain period of time. If time elapse destination will
       broadcast MISS message.
    3. Source will forward packet to its neighbor node in the discovered path and
       ACK Cnt is set to ACK Span.
    4. Until node is not equal to destination node
            d. ACK_Cnt = ACK_Cnt -1
            e. If ACK Cnt is equal to 0.
                i. Set ACK_Cnt = ACK_Span and
               ii. Node will forward packet to its neighbor node in the path
                   and generates an ACK packet.
              iii. Set TTL = ACK_ TTL
               iv. Until TTL is not equal to 0
                       The node sends the ACK packet to the previous node
                       where the previous data packet comes from
                       and TTL = TTL – 1.
            f. Else
                i. Node will forward packet to its neighbor node in the path
               ii. Wait for ACK Packet from neighbor node for certain
                   period of time.
              iii. If ACK Packet =0
                   Broadcast alarm packet.
            End
```

If source node receives MISS messages or alarm packet then source will find out a new path to the destination excluding malicious node in the path using routing protocol

# CHAPTER 4
# SYSTEM DESIGN AND IMPLEMENTATION

## 4.1 System Design

To implement the acknowledgement based reward scheme that detect malicious node. The simulation on a simple topology has been carried out using Network Simulator (ns-2) [24].

### 4.1.1 NETWORK SIMULATOR OVERVIEW

**I. Introduction**

NS-2 is an open source system that is developed using C++ and Tool Control Language TCL. Researchers can freely add new components to the system to server their own purposes. The latest version of NS-2 is version 2.32. Within this version, most of the standard protocols supported. You can find protocol from media access layer protocols such as CSMA/CD up to application protocols as FTP and HTTP. For routing protocols, there are uncast and multicast routing protocols for wire network and DSR, DSDV, AODV for wireless ad-hoc networks. Most of these protocols were developed by researchers and adopted into standard version of NS-2. NS2 is an event-driven network simulator. It has an extensible background engine implemented in C++ that uses OTcl (an object oriented version of Tcl) as the command and configuration interface. Thus, the entire software hierarchy is written in C++, with OTcl used as a front end.

**II. Tcl Scripts**

A simulation is defined by an OTcl script. Running a simulation involves creating and executing a file with a ".tcl" extension, such as "simfile.tcl."

A Tcl ns script: Defines a network topology (including the nodes, links, and scheduling and routing algorithms of a network). Defines a traffic pattern (for example, the start and stop time of an FTP session). Collects statistics and outputs the results of the simulation. Results are usually written to files, including files for Nam( the Network Animator program that comes with the full ns download.

ns is an event-driven simulator that derives its functionality through an OTcl interpreter, which runs in the background. This interpreter translates each statement in the Tcl file into an event. For example, the statement:

$ns at 0.5 "$tcp start"

is translated into event, which at 0.5 seconds into the simulation, starts up a TCP source. Marc Greis's ns tutorial [26] and the Tcl files already present in the "ns-2" directory present good examples on how to use the Tcl syntax correctly.

It is also possible to add member functions and variables to a C++ linked OTcl object. The objects in C++ that do not need to be controlled in a simulation or internally used by another object do not need to be linked to OTcl. Likewise, an object (not in the data path) can be entirely implemented in OTcl. Figure 4.1 shows an object hierarchy example in C++ and OTcl. One thing to note in the figure is that for C++ objects that have an OTcl linkage forming a hierarchy, there is a matching OTcl object hierarchy very similar to that of C++ . Figure 4.2 shows the general architecture of NS.

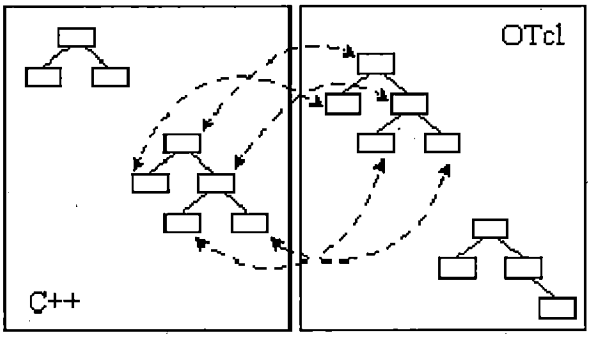


Figure 4.1. C++ and OTcl

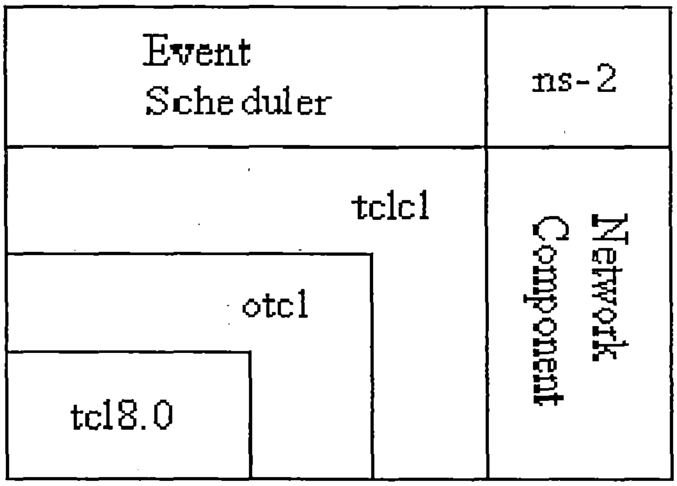


**Figure 4.2.** Architectural View of NS

## III. Software Architecture

ns is an object oriented simulator written in C++. This code serves as a backbone for the whole simulation process. The entire class hierarchy is implemented through this code; and the classes provide a wide array of network features. New classes or modules can be added by extending the current class hierarchy. Each class consists of the following of the following components:

a) **Configuration parameters:**

Configuration parameters are class attributes that can be set and queried dynamically through the Tcl scripts. These simulation parameters are usually constant during the entire simulation, but they can be changed dynamically as desired. For example, the bandwidth of a link is usually set only at the start of a simulation. On the other hand, a traffic source can be configured to transmit packets of different sizes at different times.

b) **State variables:**

Each class has a set of variables, many of which may be queried in a Tcl script to determine the state of that object. Usually, they are modified explicitly only when the object needs to be reset for another simulation run. For example, the length of a packet queue changes over time; and the instantaneous size of that queue can be queried through a Tcl command or used by a C++ method.

c) **Methods:**

The methods associated with each class specify the actions that can be performed by that object.

### IV. NS Directory Structure

Before going into a discussion of how to extend NS, let's briefly examine what information is stored in which directory or file. Figure 4.3 shows a part of the directory structure of the simulator if you installed it using the ns-allinone-2.31 package. [27]

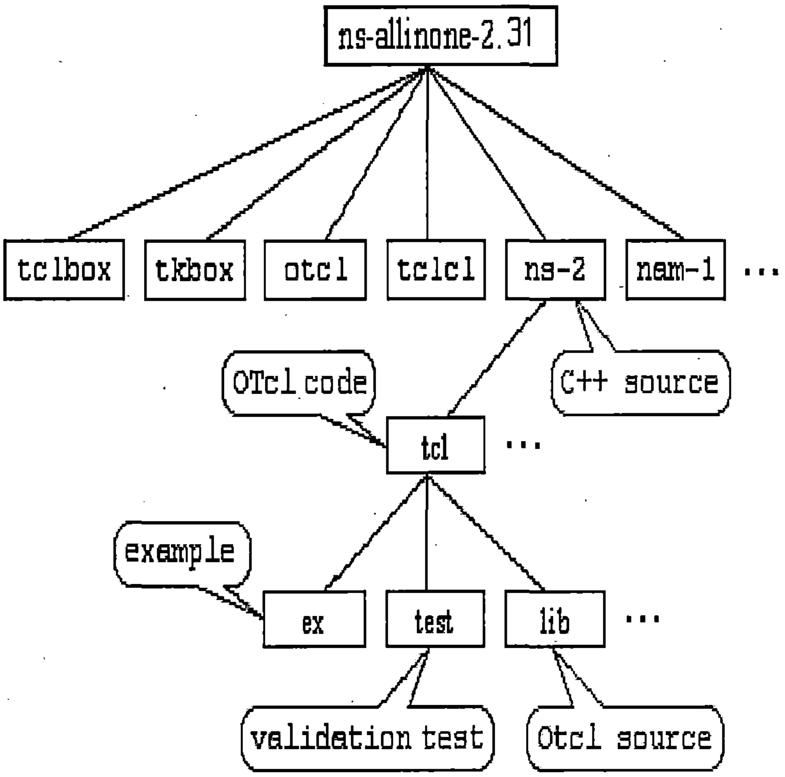


Figure 4.3 NS Directory Structure

Among the sub-directories of ns-allinone-2.31, ns-2 is the place that has all of the simulator implementations (either in C++ or in OTcl), validation test OTcl scripts and are located under a sub-directory called tcl, and most of C++ code, which implements event scheduler and basic network component object classes, except the WWW related ones, are located in the main level. For example, if you want to see the implementation of the UDP agent, you should go to "ns-allinone-2.31/ns-2" directory, and open up "udp.h", "udp.cc" and the files that contain the implementation of ancestor classes of UDP as needed.

The tcl directory has sub-directories, among which the lib directory that contains OTcl source codes for the most basic and essential parts of the NS implementation (agent, node, link, packet, address, routing, and etc.) is the place one as a user or as a developer will visit the most. Note that the OTcl source codes for LAN, Web, and Multicast implementations are located in separate subdirectories of tcl. Following is a partial list of files in "ns-2/tcl/lib" directory.

## V. Changes needed In NS2

The implementation of REWARD agent is for detection of malicious node in wireless sensor networks. To implement, REWARD Agent some changes are need to do in integrate code inside NS.

### Step 1: Packet type declaration

A constant to indicate our new packet type, *PT_ REWARD_PACKET.* That is define inside file *common/packet.h.*

First of all , find *packet _t* enumeration, where all packet types are listed and add *PT_ REWARD _PACKET* to this list as show in the next piece of code (line 6).

common/packet.h

```
1:    enum packet_t {
2:    PT_TCP,
3:    PT_UDP,
4:    PT_CBR,
5:    /* ... much more packet types ... */
6:    PT_ REWARD_PACKET ,
7:      PT_NTYPE // This MUST be the LAST one
8:      };
```

Just below in same file there is definition of **p _info** class. Inside constructor write a textual name for packet type (line 6).

common/packet.h

```
1:      p_info() {
2:      name_[PT_TCP]= "tcp";
3:      name_[PT_UDP]= "udp";
4:      name_[PT_CBR]= "cbr";
5:      /* ... much more names ... */
6:      name_[ PT_ REWARD_PACKET]= "Reward_packet";
7:      }
```

**Step 2: Tcl library**

Now some changes in Tcl files. Actually to add packet type, give default values for binded attributes and provide the needed infraestructure. In *tcl/lib/ns-packet.tcl* must locate the next code and add *Reward_packet* to the list ( in line 2).

tcl/lib/ns-packet.tcl

```
1:      foreach prot {
2:      Reward_packet
3:      AODV
4:      ARP
5:      # ...
6:      NV
7:      } {
8:      add-packet-header $prot
9:      }
```

Default values for binded attributes have to be given inside *tcl/lib/ns-default.tcl* and set packet size here. Now go to the end of the file and write something like the next code:

tcl/lib/ns-default.tcl

```
#...
# Defaults defined for Security_packet
Agent/Reward_packet set packetSize_var_ 512
```

**Step 3: Makefile**

Everything is implemented and only need to compile it. To do so edit *Makefile* file by adding new object files inside *OBJ_CC* variable as in following code (line 4).

Makefile

```
1: OBJ_CC = \
2:      tools/random.o tools/rng.o tools/ranvar.o common/misc.o common/timer-handler.o \
3:      #...
4:      reward/Reward_packete.o \
5:      #...
6:      $(OBJ_STL)
```

As modified *common/packet.h* but not *common/packet.cc* we should "touch" this last file for being recompiled. After that execute *make* and Security function is ready for work.

**Step 4 : [ns-2.31]$ touch common/packet.cc**

**Step 5: [ns-2.31]$ make**

### 4.1.2 System Components

The system consists of the following components:

Sensor node: A Sensors nodes are the heart of the network. They are in charge of collecting, processing data and routing this information back to a sink. In other world's sensor node can sense data from the environment, perform simple computations and transmit this data wirelessly to a command center either directly or in a multi-hop fashion through neighbors [27].

Source node: A source is represented by the id of the node that initiates the routing task or any other attribute value that uniquely identifies the source node

Destination node: destination is represented by a set of constraints on attributes that defines a single destination node or a destination region with multiple target nodes.

Agents: REWARD agent is created in order to provide for the functionality of the proposed framework. This is deployed in AODV.

### 4.1.3 Simulation Model

Source node: To send data from one node to other node NS2 needs agent. In NS2, data is always being sent from one agent to another. In the following Example, we create UPD agent and Null Agent to send packet from one node to other node.

```
set udp_(0) [new Agent/UDP]                 // Create a UDP Agent
$ns_ attach-agent $node_(6) $udp_(0)        // Attach to source node
set null_(0) [new Agent/Null]               // Create NULL agent
$ns_ attach-agent $node_(91) $null_(0)      //Attach to Destination node
set cbr_(0) [new Application/Traffic/CBR]   // Create CBR Traffic
$cbr_(0) set packetSize_ 1000               // Packet size
$cbr_(0) set interval_ 2.0                  //Packet will be sent every 2 sec.
$cbr_(0) attach-agent $udp_(0)              //Attach CBR with UDP
```

```
9.  $ns_ connect $udp_(0) $null_(0)            //Connect the agents
10. $ns_ at 100.0 "$cbr_(0) start"             //CBR agent send data at 100
```

We create a UDP agent and attach it to source node, then attach a CBR traffic generator to the UDP agent. CBR stands for 'constant bit rate'. The packet size is being set to 1000 and a packet will be sent every 2 seconds.

Destination node: The server is modeled by a simple UDP Sink which sends out ACK packets for packets it receives. As we create traffic sink in line 3 of above example.

## 4.2 Implementation

### 4.2.1 Simulation Topology

Figure 4.4 illustrates the simulated network topology. The topology, which is introduced here, is similar to existing REWRAD topology. Here each intermediate node along the forwarding path is in charge of detecting malicious nodes. If an intermediate node detects the misbehavior of nodes, it will broadcast an alarm packet and deliver it to the source node through multiple hops.

The simulation is carried out in Network Simulator ns-2 [24] in which the functionality of modified REWARD agent is coded. Malicious nodes are drop packets and our scheme detects the malicious node in the path. In the following Figure 4.4, Black nodes are malicious, and remaining nodes are non malicious. Source and destination are fixed and AODV protocol is used for routing.

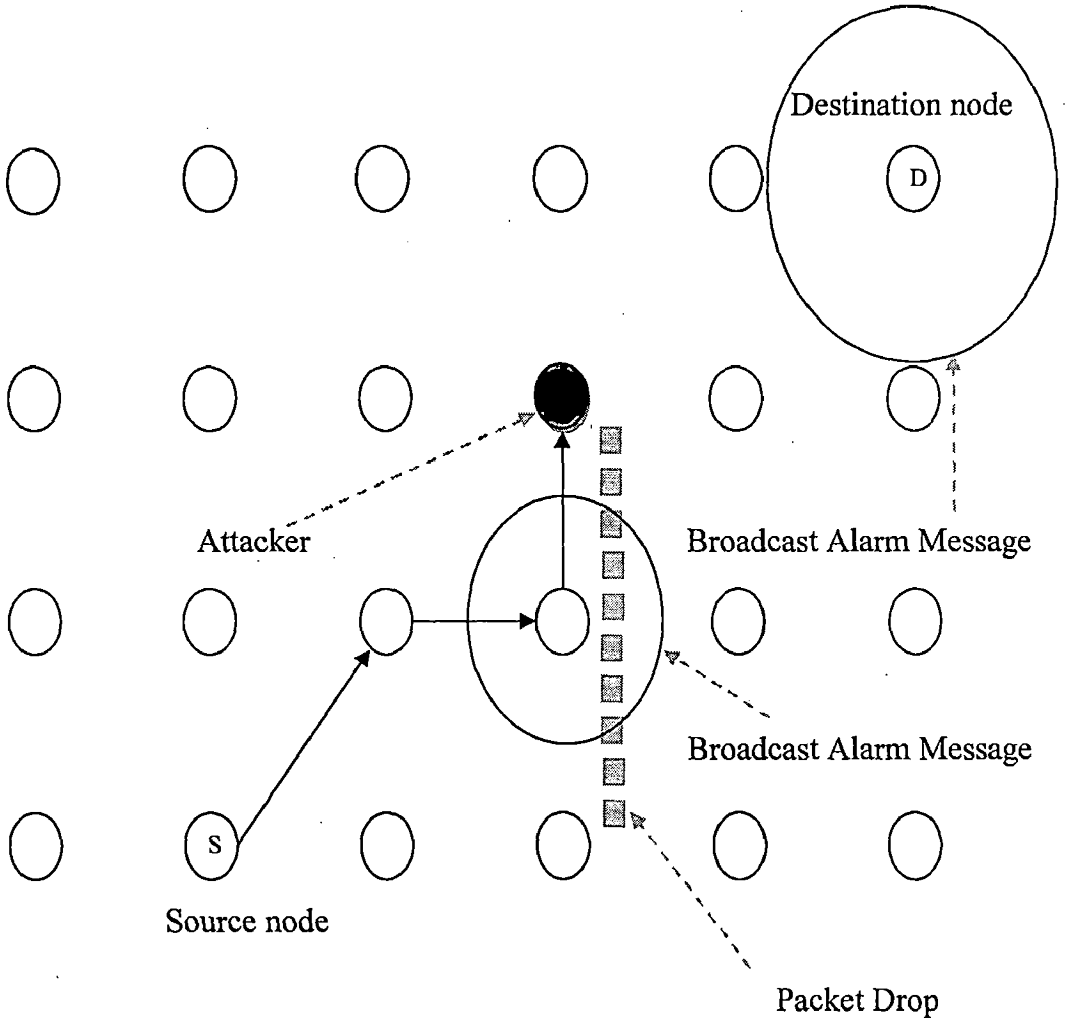


Figure 4.4 The simulated network topology.

### 4.2.2 Simulation Parameters

Table 4.1 lists the simulation parameters, their values and description of these parameters used in the simulation.

| Parameter | Value | Description |
|---|---|---|
| Chan | Channel/WirelessChannel | Channel type |
| prop | Propagation/TwoRayGround | Radio-propagation model |
| netif | Phy/WirelessPhy | Network interface type |

| | | |
|---|---|---|
| MAC | Mac/802_11 | MAC type |
| ifq | Queue/DropTail/PriQueue | Interface queue type |
| ll | LL | Link layer type |
| Ant | Antenna/OmniAntenna | Antenna model |
| ifqlen | 100 | Max packet in ifq |
| rp | AODV | Routing protocol |
| nn | 100 | Number of Nodes |
| x | 800 | X dimension of topography |
| y | 800 | Y dimension of topography |
| stop | 20.0 | Time to stop simulation |
| energyModel | EnergyModel | |
| initialEnergy | 1000.0 | Total Energy of Batter |
| rxPower | 30 | Receiving Power |
| txPower | 81 | Transmitting Power |

Table 4.1: Simulation Parameters

# CHAPTER 5
# RESULTS AND DISCUSSION

Results of the simulated model (implemented in NS2) are analyzed and discussed by varying the parameter used in the model.
In this section, we evaluate the performance, such as the communication overhead and energy consume per bit of the proposed detection algorithm through simulations. We use a field size of 800 * 800, where 100 nodes are uniformly distributed and there is one stationary sink (Destination) and one stationary source. We carry out a simulation event in which the source generates 500 data packets in total and one data packet is sent out every two seconds. Packets can be delivered hop-by-hop at 19.2 Kbps. To make our scheme more resilient in poor radio conditions, we implement a hop-by hop transport layer retransmission mechanism beneath our scheme, which is quite similar to that in PSFQ [28].

## 5.1 Analysis of Overhead

We analyze the trace file that was generated during simulation and analyzed the overhead for finding paths and sending a packet for REWAED scheme and proposed scheme. To compare the performance of proposed detection scheme with REWARD detection scheme, we chose AODV which is a wireless routing protocol, Nodes executing the AODV protocol, to find the path and the forward packets to the neighbor.

The cost of sending a packet be Tx = 81mW and the cost of receiving a packet be Rx=30mW , Number of nodes N = 100, The average number of node neighbors (*neighbor* of a node is any other node within its broadcast range) *m* = 4. *p* be the length of the path. The approximate cost of route request and route reply is *N (*Tx + *m*Rx*)* and *p* (Tx + Rx*)*, respectively. The cost of finding a path is *p(*Tx + Rx*)*.

Therefore, the ratio of the cost of finding a path to the cost of Routing protocol path discovery is *p(*Tx + Rx*)* / *(N (*Tx + *m*Rx*)* + *p(*Tx + Rx*))*.

Here numbers of acknowledgement are reduced than compare to acknowledgment generated in case of REWARD scheme, thus packet overhead in our scheme is reduced. We obtain result by analyzing the trace file, which shows the overhead of sending packet in the network. We tested by varying the path length from 5 to 15 hops. We randomly selected malicious nodes and assumed that only malicious node drop packets. The intermediate nodes were used to verify the packet is send properly or not. Based on simulation results obtained, we calculated the ratio of overhead as show in figure 5.1, the overhead is proportional to the path length in case of REWARD scheme.

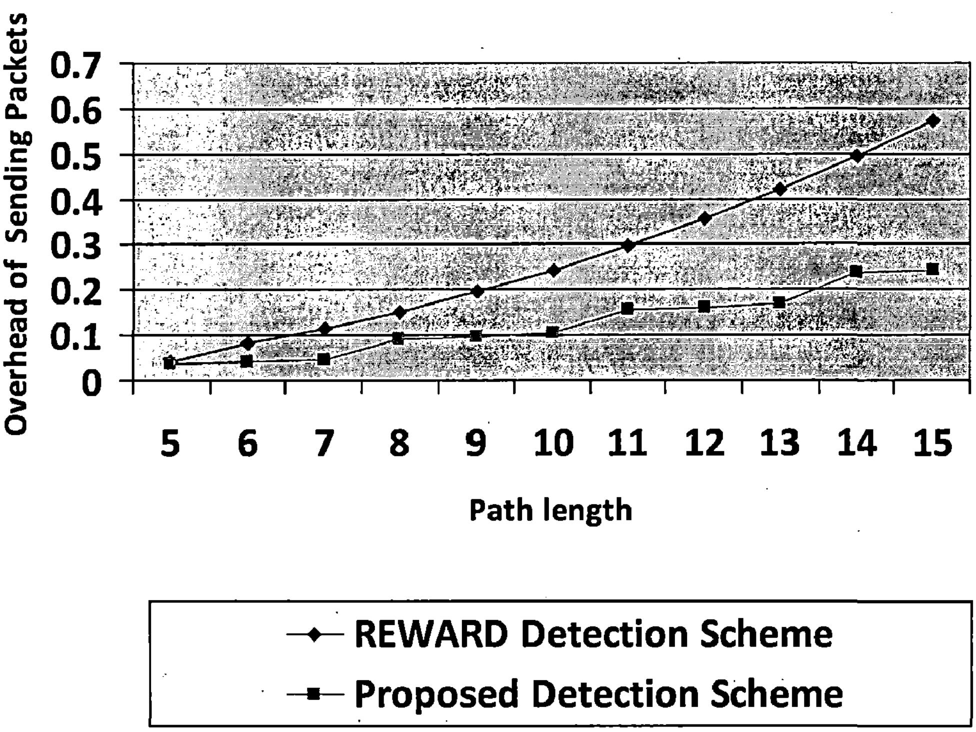
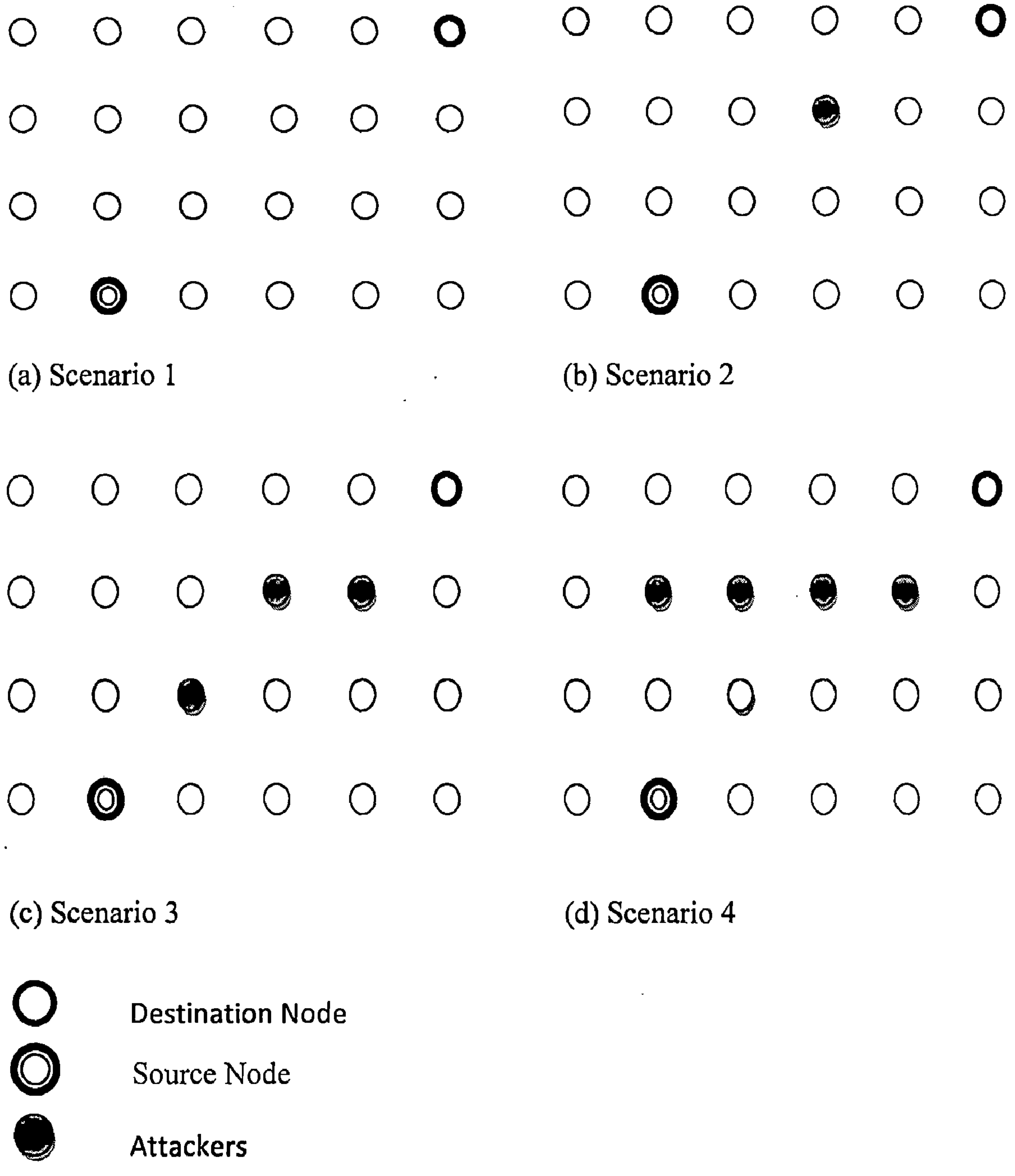


Figure 5.1 Comparing overhead of REWARD and Proposed scheme

## 5.2 Scenarios for the Simulation Study

To consider different attack models, we use the four different scenarios shown in Figure 5.2. In scenario 1, there is no malicious node present in the network. In scenario 2, only one malicious node resides on the shortest path from the source to destination constructed by AODV. In scenario 3, most of the nodes constructing the shortest path are malicious. In scenario 4, the malicious nodes form a wall across the network and try to separate the source and destination.

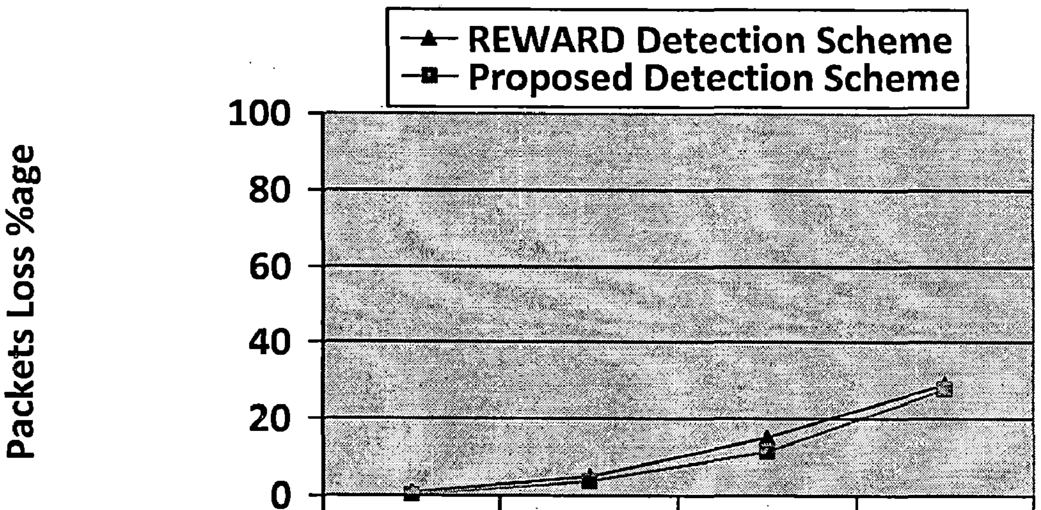

(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

(d) Scenario 4

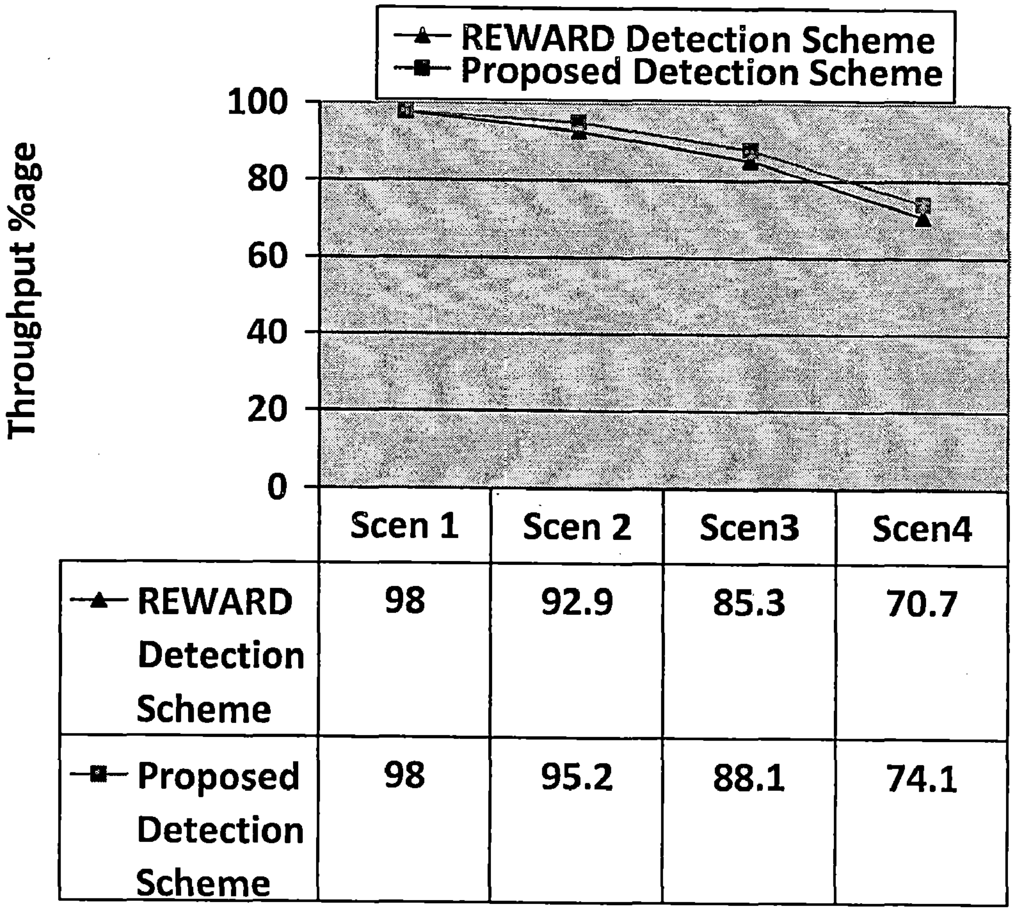
Figure 5.2 Scenarios for the Simulation Study



| | Scen 1 | Scen 2 | Scen3 | Scen4 |
|---|---|---|---|---|
| ▲ REWARD Detection Scheme | 1 | 5.1 | 15.2 | 29 |
| ■ Proposed Detection Scheme | 0.33 | 3.9 | 11.5 | 27.6 |

Figure 5.3 Simulation Result with varying number of malicious node

The results obtained of the simulation, shown in figure 5.3, indicate that on the average, the packet loss in nodes executing the proposed algorithm was almost same than executing the REWARD algorithm but when malicious nodes from a wall across the network then the packet loss in our scheme 3% lower than REWARD scheme.



| | Scen 1 | Scen 2 | Scen3 | Scen4 |
|---|---|---|---|---|
| ▲ REWARD Detection Scheme | 98 | 92.9 | 85.3 | 70.7 |
| ■ Proposed Detection Scheme | 98 | 95.2 | 88.1 | 74.1 |

Figure 5.4 Simulation Result with varying number of malicious node

By comparing throughput of four scenarios from Figure 5.4 we can state that with increasing in malicious nodes throughput is decreasing. In scenario 1 performance is similar in both cases whereas in scenario 2,3 and 4 has better throughput than REWARD scheme.

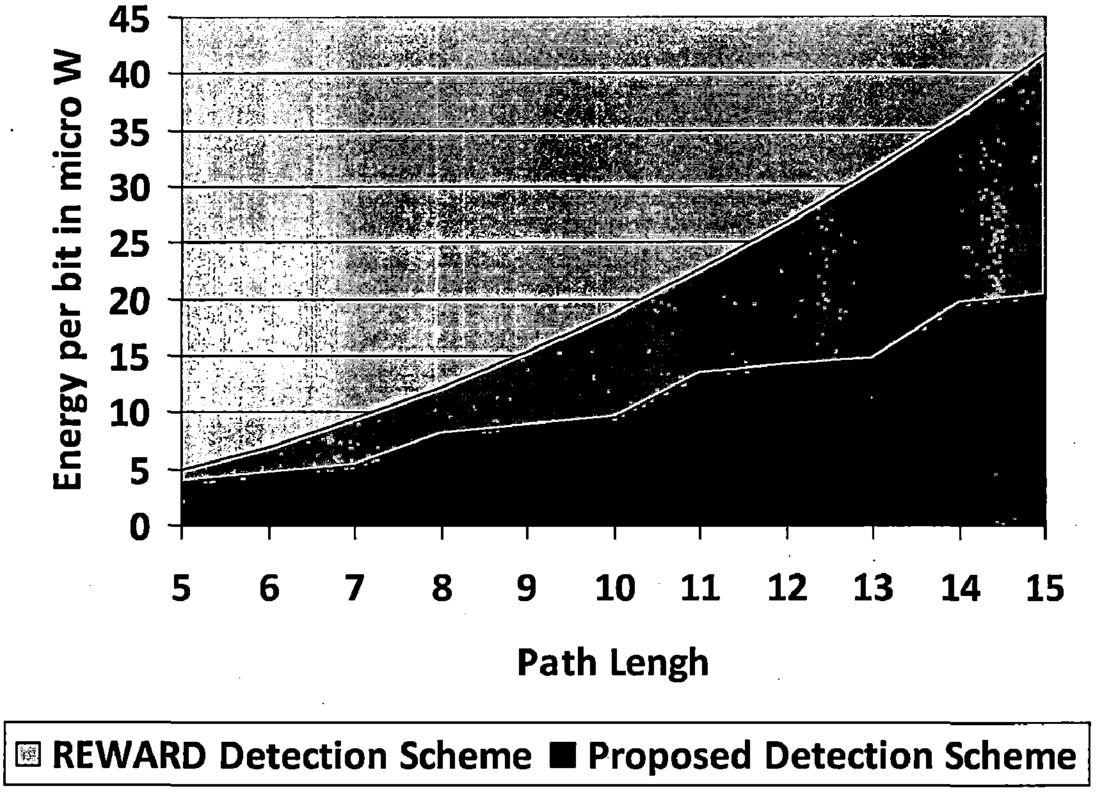


Figure 5.5 Comparing energy consumption in REWARD and Proposed scheme

| Path Length | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| REWARD (mW) | 04.90 | 07.00 | 09.45 | 12.25 | 15.40 | 18.90 | 22.75 | 26.95 | 31.50 | 36.40 | 41.65 |
| Our Model (mW) | 03.85 | 04.55 | 05.25 | 08.05 | 08.75 | 09.45 | 13.3 | 14.00 | 14.70 | 19.60 | 20.30 |

In our proposed scheme, since acknowledgements overhead is reduced compared to REWARD scheme energy consumption is decreased and it can be easily verified from Figure 5.5. Also we can see from Figure 5.5 as the number of hops increases in the path length rate of energy consumption is more in REWARD scheme.

# CHAPTER 6

# CONCLUSIONS AND FUTURE WORK

## 6.1 Conclusions

A critical issue for security in wireless sensor networks is how to detect attacks on the network in an accurate and efficient manner. We have presented a simple and efficient security scheme for detecting packet drop attack by malicious node. We have implemented variation of REWARD detection scheme which optimizes the energy of the node also comparatively reduces the traffic on the network by sending ACK packets instead of sending the same packet as in case of REWARD scheme, with using the identifier. Thus we minimized the number of ACK packets sent by the intermediate nodes in the network. We had evaluated the scheme for single and multi malicious node detection in the network. In our work, AODV was used as a routing methodology in the sensor network

In our experiment we generated a malicious node and detect the same. We had tested our strategy on a field size of 800×800 m$^2$ using 100 nodes are uniformly distributed.

## 6.2 Suggestions for Future Work

Through simulations, we observe that it's difficult to distinguish packet loss due to compromised nodes from that due to outside jammers without jamming detection technique. Both of them share similar symptoms. This form one the areas for future research where could find solutions to detect malicious nodes which cause both the problems of jamming and packet loss.

Further, assumptions made in our work like channel error which also leads to packet loss could be included for further research.

# REFERENCES

[1] I.F. Akyildiz , W. Su , Y. Sankarasubramaniam and E. Cayirci. "Wireless Sensor Networks: A Survey" *IEEE Computer Networks*, vol. 38, no. 4, pages 393-422, Mar. 2002.

[2] S.M. Diamond and M.G. Ceruti, "Application of Wireless Sensor Network to Military Information Integration" *In Proceedings of 5th IEEE International Conference on Industrial Informatics,* vol. 1, pages317-322, June 2007

[3] Yuan Lingyun, Zhu Yunlong and Qu Licheng, "Research on wireless sensor network applied in highway transportation monitoring system," *The Seventh International Conference on Electronic Measurement and Instruments*, vol 4, pages 336-340, Aug. 2005.

[4] Z. Karakehayov, "Security - Lifetime tradeoffs for wireless sensor networks", *In Proceedings of IEEE Conference on Emerging Technologies & Factory Automation, ETFA*, pages 646-650, Sept. 2007.

[5] I.F. Akyildiz, , W. Su, Y. Sankarasubramaniam, and E.Cayirci. "A survey on sensor network", *In IEEE Communication Magazine,* vol. 40, no. 8, pages 102–114, 2004.

[6] Chris Karlof, and David Wagner. "Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures", *In Proceeding of 1st IEEE International Workshop on sensor Network Protocols and Applications*, pages 293 – 312, May 2003.

[7] C.Intanagonwiwat, R.Govindan and D. Estrin., "Directed diffusion: A scalable and robust communication paradigm for sensor networks", *In Proceedings of the 6th annual international conference on Mobile computing and networking,* pages 56-67, Aug. 2000.

[8] C. Karlof, Y. Li, and J. Polastre., "ARRIVE: An Architecture for Robust Routing", *In Volatile Environments. Technical Report UCB/CSD-03-1233, University of California at Berkeley,* pages 1-11, Mar.2003.

[9] S. Madden, R. Szewczyk, M. Franklin, and D. Culler., "Supporting Aggregate Queries over Ad-Hoc Wireless Sensor Networks", *In Proceeding of 4th IEEE Workshop on Mobile Computing Systems and Applications,* pages 49-58, June 2002.

[10] R. Samuel Madden, J. Michael Franklin, M. Joseph Hellerstein, and Wei Hong., "TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks", *In Proceeding of Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002),* vol. 36, pages 131-146, 2002.

[11] B. Krishnamachari, D. Estrin, and S. B. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks", *In Proceedings of the 22nd international Conference on Distributed Computing Systems ICDCSW-02*, pages 575-578, July 2002.

[12] Chris Karlof, Naveen Sastry, David Wagner, "Tinysec: A Link Layer Security Architecture for Wireless Sensor Networks", *In Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 162 – 175, Nov. 2004.

[13] J. Newsome, E. Shi, D. Song and A. Perrig, "The Sybil Attack in Sensor Networks: Analysis & Defenses", *In Proceedings of the Third international Symposium on information Processing in Sensor Networks* IPSN *'04.* Pages 259-268, Apr. 2004

[14] Sabbah, Adnan Majeed, Kyoung-Don Kang, Ke Liu and Nael Abu-Ghazaleh, "An Application -Driven Perspective on Wireless Sensor Network Security", *In Proceedings of the 2nd ACM international workshop on Quality of service & security for wireless and mobile networks*, Pages 1-8, Oct. 2006.

[15] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security Protocols for Sensor Networks," *In Proceeding Wireless Networks*, vol. 8, no. 5, pages 521–534, Sep. 2002.

[16] Huang, Y. and Lee, W., "A Cooperative Intrusion Detection System for Ad Hoc Networks," *In Workshop on Security of Ad Hoc and Sensor Networks (SASN '03)*, pages 135-147, Oct. 2003.

[17] Donggang Liu, Peng Ning, and Wenliang Du, "Detecting Malicious Beacon Nodes for Secure Location Discovery in Wireless Sensor Networks", *In Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, pages 609 – 619, 2005.

[18] S. Buchegger, and Le Boudec, J., "Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes -Fairness in Dynamic Ad-hoc Networks", *In Proceedings of 13th IEEE/ACM Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc),* pages 226 - 236 , June 2002.

[19] P. Michiardi, and R. Molva, "CORE: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks", *In Proceedings of IFIP 6th Joint Working Conference on Communications and Multimedia Security (CMS.02)*, , pages 107-122, Sep. 2002.

[20] Z. Karakehayov and I. Radev, "REWARD: A Routing Method for Ad-Hoc Networks with Adjustable Security Capability", *In NATO Advanced Research Workshop "Security and Embedded Systems*, pages 180-187, Aug. 2005.

[21] Z. Karakehayov, "Using REWARD to Detect Team blackhole Attacks in Wireless Sensor Networks", *In Workshop on Real-World Wireless Sensor Networks, REALWSN'5*, pages 1-5, June 2005.

[22] C. Perkins and E. Royer, "Ad Hoc on Demand Distance Vector Routing", *In Proceedings of 2nd IEEE Workshop Mobile Comp. Sys. & Apps*, pages 90-100, Feb. 1999.

[23] Y. W. Law, L. V. Hoesel, J. Doumen, P. Hartel and P. Havinga. "Energy-Efficient Link-Layer Jamming Attacks against Wireless Sensor Network MAC Protocols", *In Proceeding of the 3rd ACM on the Security of Ad Hoc and Sensor Networks (SASN 05)*, pages 76-88, 2005.

[24] "NS by Example" available at http://nile.wpi.edu/NS/ ,last accessed on 8 June 2008.

[25] "Tutorial for the network simulator" available at http://www.isi.edu/nsnam/ns/tutorial/ ,last accessed on April 2008.

[26] "Downloading and installing ns-2" available at http://nsnam.isi.edu/nsnam/index.php/Downloading_and_installing_ns-2 ,last accessed on November 2007.

[27] D Mahjoub, and H El-Rewini – "Adaptive Constraint-Based Multi-Objective Routing for Wireless Sensor Networks", *In Proceedings of Conference on Pervasive Services, IEEE International,* pages 72-75, 2007.

[28] C. Y. Wan, A. T. Campbell, and L. Krishnamurthy. "PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks", *In Proceeding of the first ACM International Workshop on Wireless Sensor Networks and Applications*, pages 1-11, 2002.