A RULE BASED INTRUSION DETECTION FRAMEWORK WIRELESS SENSOR NETWORK

A PROJECT REPORT

Submitted by

ESWARI T

in partial fulfillment for the requirement of award of the degree of

MASTER OF ENGINEERING

FACULTY OF COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE 641 049

(An Autonomous Institution Affiliated to Anna University, Chennai)

APRIL 2013

A RULE BASED INTRUSION DETECTION FRAMEWORK WIRELESS SENSOR NETWORK

A PROJECT REPORT

Submitted by

ESWARI T

in partial fulfillment for the requirement of award of the degree of

MASTER OF ENGINEERING
in

COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE 641 049

(An Autonomous Institution Affiliated to Anna University, Chennai)

APRIL 2013

BONAFIDE CERTIFICATE

Certified that this project work titled **“A RULE BASED INTRUSION DETECTION FRAMEWORK WIRELESS SENSOR NETWORK”** is the bonafide work of Ms. ESWARI, T. (1120108004) who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other students.

Prof. N. Jayapathi M.Tech.,	Dr. V. Vanitha Ph.D.,
PROFESSOR & HEAD	SUPERVISOR
Computer Science and Engineering,	Senior Associate Professor,
Kumaraguru College of Technology,	Computer Science and Engineering,
Coimbatore-641 049.	Kumaraguru College of Technology,
	Coimbatore-641 049.

Submitted for the Project Viva-Voce examination held on ____________.

-------------------------------	----------------------------
Internal Examiner	External Examiner

ABSTRACT

A Wireless Sensor Networks (WSNs) is an emerging technology in wireless field that involves the deployment of hundreds of low-cost, micro-hardware and resource limited sensor nodes. The technology uses sensor nodes to sense data such as temperature, pressure, image, etc. The applications of WSN include military applications, environmental monitoring, health care, home or industrial automation etc. So WSNs are vulnerable to attack, because the sensor nodes are deployed in hostile or unattended environment, they are not temper resistant and the radio links for communication are insecure. There were many schemes proposed for preventing WSN from security threats. The Intrusion Detection System can be implemented for preventing and securing the WSNs from various security threats. An intrusion detection framework is proposed for securing wireless sensor networks from routing attacks. This Intrusion Detection System can be implemented with the rules, those rules can be applied for detecting the attacks. The rule based intrusion detection framework has been introduced for showing its potential improvements towards the detecting various routing attacks such as wormhole, black hole and sybil attack. The simulated result shows that the proposed Intrusion Detection System produces high throughput and packet delivery ratio while detecting the routing attacks.

ACKNOWLEDGEMENT

First and foremost, I would like to thank the Lord Almighty for enabling me to complete this project.

I express my profound gratitude to our **Chairman Padmabhusan Arutselvar Dr.N.Mahalingam B.Sc., F.I.E., Co-Chairman Dr. B. K. Krishnaraj Vanavarayar B.Com., B.L., Correspondent Mr. M. Balasubramaniam M.com., M.B.A., and Joint Correspondent Mr. K. Shankar Vanavarayar MBA., PGDIEM.,** for giving this great opportunity to pursue this course. I would like to thank Principal **Dr.S.Ramachandran Ph.D.,** for providing the necessary facilities to complete my thesis.

I take this opportunity to thank **Prof.N.Jayapathi M.Tech.,** Head of the Department, Department of Computer Science and Engineering, for his support and timely motivation. I also thank all project committee members for their comments and advice during the reviews.

I register my sincere thanks to my Guide and Project Coordinator **Dr.V.Vanitha Ph.D.,** Senior Associate Professor, Department of Computer Science and Engineering, for arranging brain storming project review sessions. I am grateful for her support, encouragement and ideas. I would like to convey my honest thanks to all **Teaching** and **Non Teaching** staff members of the department for their support.

I dedicate this project work to my **Parents** for no reasons but feeling from bottom of my heart that without their love, this work wouldn't be possible.

- ESWARI T.

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ADC	Analog to Digital Converter
ACK	Acknowledgement
CTP	Collaboration Tree Protocol
CWSN	Cluster Based Wireless Sensor Network
CH	Cluster Head
DoS	Denial of Service
DM/CI	Data Mining and Computational Intelligence
IDF	Intrusion Detection Framework
IDS	Intrusion Detection System
MAC	Media Access Control
SN	Sensor Node
WSN	Wireless Sensor Network

CHAPTER 1

INTRODUCTION

Wireless Sensor Network (WSN) is an emerging technology in wireless field which consists of spatially, distributed autonomous sensors to monitor physical or environmental condition such as temperature, sound, vibration, pressure, motion or pollutants and to co-operatively pass their data through the network to main location (Akyildiz, 2002).

1.1 WSN ARCHITECTURE

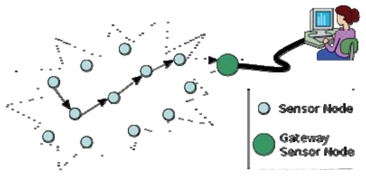

Figure 1.1 WSN Architecture

In Figure 1.1, the sensor nodes or field devices are end devices mounted on locations where physical parameters are to be monitored. **The gateway or access point**

who is responsible for configuration of the network, scheduling communication between devices, management of the routing tables, monitoring and reporting the health of the network.

1.2 SENSOR NODE AND ITS COMPONENTS

A Wireless Sensor Network consists of the sensor nodes which are small, lightweight and portable. These nodes form a network by communicating with each other directly or through other nodes. The main components of a sensor node are microcontroller, transceiver, external memory, power source and one or more sensors as shown in Figure 1.2.

The transducer (ADC - Analog to Digital Converter) is responsible to generate electrical signals based on sensed phenomena and physical effects. The microcontroller's work is to process and store the sensor output. The transceiver receives command from a central computer or base station and transmits data to the computer or station. Sensor nodes are catered power by a battery. Some sensor nodes include external memory which may be on-chip memory of a microcontroller and flash memory. The need of memory of a sensor node is application specific. Each node may also belong to two extra components like location finding system and mobilizer. The location finding system is required since the user may in need of location with high accuracy, and mobilizer may be needed to move sensor nodes to carry out the assigned tasks.

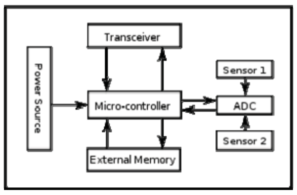

Figure 1.2 Components of a sensor node

Sensor nodes are fitted with an onboard processor. Instead of sending the raw data to the nodes responsible for the fusion, they use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data.

1.3 DESIGN CHALLENGES OF WSN

a. **Scalable and flexible architecture -** The network must preserve its stability. Introducing more nodes into the network means that additional communication messages will be exchanged, so that these nodes are integrated into the existing network.

b. **Error-prone wireless medium -** The wireless medium in WSN can be greatly affected by noisy environments.

c. **Fault tolerance and adaptability** - Fault tolerance means maintaining sensor network functionalities without any interruption due to failure of sensor node.

d. **Infrastructure** - Sensors network are infrastructure less in which nodes can communicate directly with base station.

e. **Node Deployment** - Sensor network can be deployed randomly in geographical area. After deployment, they are maintained automatically without human presence.

f. **Dynamic changes** - As in sensor network, nodes are deployed without any topology and they are adaptable to changes due to addition of new nodes or failure of nodes.

g. **Power Consumption** – If wireless sensor node is microelectronic device then it is equipped with a limited number of power source. Nodes are dependent on battery for their power. Hence power conservation and power management is an important issue in sensor network.

h. **Hardware design-** While designing any hardware of sensor network, it should be energy-efficient.

i. **Limited computational power and memory size-** It is another factor that affects WSN in the sense that each node stores the data individually and sometime more than one node stored same data and transferred to the base station which waste the power and storing capacity of nodes so they must develop effective routing schemes and protocols to minimize the redundancy in the network.

j. **Security**- Security is an important parameter in sensor network. Since sensor networks are data centric, there is no particular id associated with sensor nodes and attacker can easily insert himself into the network and steal the data by becoming the part of network without the knowledge of sensor nodes of the

network. So it is difficult to identify whether the information is authenticated or not.

1.4 APPLICATIONS

Wireless Sensor Networks are used in a variety of applications which requires the constant monitoring and detection of specific events.

a. **Area Monitoring -** In area monitoring, the WSN is deployed over a region where some phenomenon is to be monitored. When the sensors detect the event being monitored (heat, pressure), the event is reported to one of the base stations, which then takes appropriate action. For example: military uses these sensors to detect enemy intrusion; a civilian uses these sensors for geo-fencing of gas or oil pipelines.

b. **Air Pollution monitoring -** Wireless Sensor Networks have been deployed in several cities to monitor the concentration of dangerous gases for citizens.

c. **Forest fires detection -** A network of sensor nodes can be installed in a forest to control when a fire starts. The nodes will be equipped with sensors to control temperature, humidity and gases which are produced by fire in the trees or vegetation. The early detection is crucial for a successful action of the fire-fighters.

d. **Agriculture** - Using Wireless Sensor Network within the agricultural industry is increasingly common. Gravity fed water systems are monitored using pressure transmitters to monitor water tank levels, pumps can be controlled using wireless I/O devices. Irrigation automation enables more efficient water use and reduces waste.

1.5 ATTACKS ON SENSOR NETWORKS

Wireless Sensor Network is vulnerable to different kind of attacks. Attacks divided them into different categories based on physical layer, application layer, network layer and MAC layer attacks. The Table 1.1 shows layering based attacks and possible security approaches.

Table 1.1 Layering based attacks and security approaches

Layer	Attacks	Security Approach
Physical Layer	Jamming and tampering	Use spread spectrum techniques and MAC layer admission control mechanisms
Data Link Layer	Jamming and collision	Use error correcting codes and spread spectrum techniques
Network Layer	Packet drop, bogus routing information and tunnel	Authentication
Transport Layer	Injects false messages and energy drain attacks	Authentication
Application Layer	Attacks on reliability	Cryptographic approach

1.5.1 Physical layer attacks

Since Wireless Sensor Networks can be deployed in hostile environment or densely populated areas, physical access to individual nodes is possible. Even casual passerby may to damage, destroy, or tamper with sensor devices. Destruction of the node could cause gaps in sensor or communication coverage. More equipped attacks can interrogate a device's memory, stealing its data or cryptographic keys. Its code can be replaced with a malicious program which is potentially undetectable to neighboring nodes. The attacks in physical layer are jamming and tampering.

a. Jamming attack

A well-known attack on wireless communication jamming, interferes with the radio frequencies of a network's nodes that are used. An adversary can disrupt the entire network with *k* randomly distributed jamming nodes, putting *N* nodes out of service, where *k* is much less than *N*. For single frequency networks, this attack is simple and effective.

b. Tampering attack

An attacker can also tamper with nodes physically, and interrogate and compromise them. An attacker can damage or replace sensor and computation hardware or extract sensitive material such as cryptographic keys to gain unrestricted access to higher levels of communication. Node destruction may be indistinguishable from fail-silent behavior.

1.5.2 Link layer attack

The link or Media Access Control (MAC) layer provides channel arbitration for neighbor-to-neighbor communication. Cooperative schemes that rely on carrier

sense, which let nodes detect if other nodes are transmitting, are particularly vulnerable to DoS. The attacks in MAC layer are jamming and collision.

Collision

Adversaries may only need to induce a collision in one octet of a transmission to disrupt an entire packet. A change in the data portion would cause a checksum mismatch at some other receiver. A corrupted Acknowledgement (ACK) control message could induce costly exponential back-off in some MAC protocols.

1.5.3 Network layer attack

Higher layers may not require fully reliable transmission streams, but the network layer provides a critical service nonetheless. In a large-scale deployment, messages may traverse many hops before reaching their destination. Unfortunately, as the aggregate network cost of relaying a packet increases, so does the probability that the network will drop or misdirect the packet along the way. Most of the attacks on wireless sensor are routing related. It may due to abnormal updates to routing table, malicious packet on a route, false route requests, too many packets, no reply from destination, packet with incorrect encryption, failed to forward a packets, etc..

a. Spoofed, altered, or replayed routing information

This attack targets the routing information exchanged between the nodes. Adversaries may be able to create routing loops, attract or repel network traffic, extend or shorten source routes, generate false error messages, partition the network, and

increase end-to-end latency. Figures 1.3 show how an adversary can attract, repeal the network traffic respectively, by advertising a false path. And a scenario in which an adversary node creates a routing loop in the network.

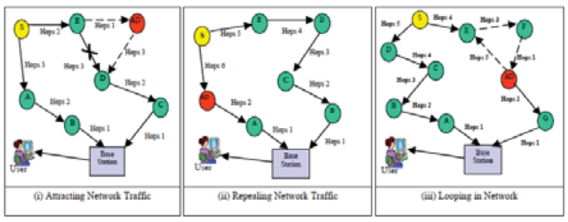
(i) Attracting Network Traffic
(ii) Repealing Network Traffic
(iii) Looping in Network
Base Station
User

Figure 1.3 Spoofed, altered, or replayed routing information

b. Selective forwarding attack

However a malicious node may refuse to forward certain messages and simply drop them, ensuring that they are not propagated any further. Figure 1.4 shows scenarios of selective forward attack. In first scenario source node 'S' forwards its data packet D1, D2, D3, D4 to node 'A' and node 'A' forward these received packets to node 'B'. In other hand an adversary node AD selectively forwards packets D1, D3 while dropping packet D2 and D4. In another scenario an adversary may selectively drop packets originated from one source and forward that of others.

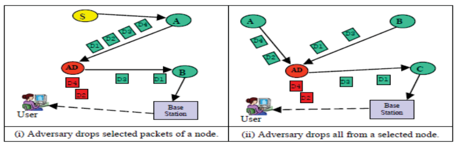
(i) Adversary drops selected packets of a node.
(ii) Adversary drops all from a selected node.
AD
Base Station
User

Figure 1.4 Selective forwarding attack

c. Sinkhole attack

By sinkhole attack, the adversary tries to attract nearly all the traffic from a particular area through a compromised node. A compromised node which is placed at the centre of some area creates a large "sphere of influence", attracting all traffic destined for a base station from the sensor nodes. The Figure 1.5 demonstrates sinkhole attack where 'SH' is a sinkhole. This sinkhole attracts traffic from nearly all the nodes to route through it.

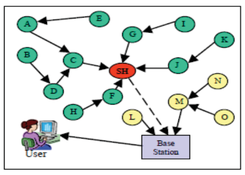

Figure 1.5 Sink hole attack

d. Sybil attack

Most protocols assume that nodes have a single unique identity in the network. In a Sybil attack, an attacker can appear to be in multiple places at the same time. This can be convincing by creating fake identities of nodes located at the edge of communication range. Multiple identities can be occupied within the sensor network either by fabricating or stealing the identities of legitimate nodes. Hence identity fraud leads to the Sybil attack, proper authentication can defend it. The Figure 1.6 demonstrates Sybil attack where an adversary node 'AD' is present with multiple identities. 'AD' appears as node 'F' for 'A', 'C' for 'B' and 'A' as to 'D' so when 'A' wants to communicate with 'F' it sends the message to 'AD'.

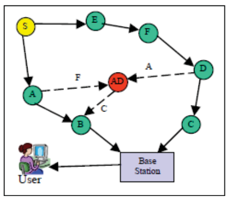

Figure 1.6 Sybil attack

e. Wormhole attack

In this attack an adversary could convince nodes who would normally be multiple hops from a base station that they are only one or two hops away via the

wormhole. The simplest case 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 adversary situated close to a base station may be able to completely disrupt routing by creating a well-placed wormhole. Figure 1.7 demonstrates Wormhole attack where 'WH' is the adversary node which creates a tunnel between nodes 'E' and 'I'. These two nodes are present at most distance from each other.

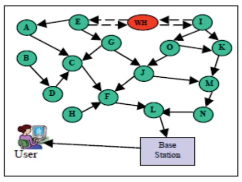

Figure 1.7 Worm hole attack

f. HELLO flood attack

A laptop-class attacker with large transmission power could convince every node in the network that the adversary is its neighbor, so that all the nodes will respond to the HELLO message and waste their energy. If the attacker subsequently advertises low-cost routes, nodes will attempt to forward their messages to the attacker. The Figure 1.8 depicts how an adversary node 'AD' broadcast hello packets to convince nodes in the network as neighbor of 'AD'. Though some node like I,H,F are far away

from 'AD' they think 'AD' as their neighbor and try to forward packets through it which results in wastage of energy and data loss.

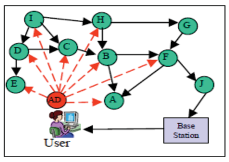

Figure 1.8 Hello Flood Attack

g. Black-hole attack

The black hole attack positions a node in range of the sink and attracts the entire traffic to be routed through it by advertising itself as the shortest route. This attack can isolate certain nodes from the base station and creates a discontinuity in network connectivity. When it receives data from neighboring nodes it drops them. In the Figure 1.9 BH is the black-hole which first convenes the network that it is the nearest node to base station and attracts the network to rout data through it. When it receives data from neighboring nodes it drops them.

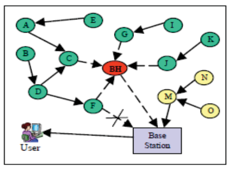

Figure 1.9 Black hole attack

1.5.4 Transport layer attacks

The transport layer attack includes the process of injecting false data in messages transmission and makes energy exhaustion. The attacks in this layer are message integrity and energy drain attack.

a. Data integrity attack

Data integrity attacks compromise the data travelling among the nodes in WSN by changing the data contained within the packets or injecting false data. The goals of this attack are to falsify sensor data and by doing so compromise the victim's research. It also falsifies routing data in order to disrupt the sensor network's normal operation, possibly making it useless. Figure 1.10 shows an example of Data Integrity attack. In the Figure node A sends a data packet to B. This packet contains destination id (B), data (10) and packet sequence number (1). An adversary node AD modifies this data as 5 and forwards it to node B.

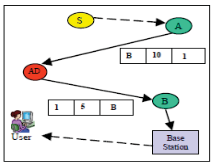

Figure 1.10 Data Integrity Attack

b. Energy drain attack

It is difficult to replace/recharge sensor node batteries in WSN. Because there is a limited amount of energy available, attackers may use compromised nodes to inject fabricated reports into the network or generate large amount of traffic in the network. and drain the finite amount of energy in a battery powered network. In Figure 1.11 adversary node 'AD' generates false data continuously. Its immediate neighbor nodes 'D','F' and 'G' responds to 'AD' and finally drains there battery.

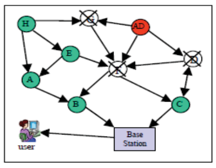

Figure 1.11 Energy Drain Attack

1.5.5 Application layer attacks

The most common kind of application or service level attack is the denial of services attack. A denial of service attack is any event that diminishes or eliminates a network's capability to perform its expected functions. It is the general result of any action that prevents any part of a WSN from functioning correctly or in a timely manner. Hardware failures, software bugs, resource exhaustion, environmental conditions, or any complicated interaction between these factors can cause Denial of Service (DoS) attack.

a. Denial of Service attack

A DoS attack is any event that diminishes or eliminates a network's capacity to perform its expected function. Hardware failures, software bugs, resource exhaustion, environmental conditions, or any complicated interaction between these factors can cause a DoS.

1.6 WSN SECURITY ANALYSIS

Securing the Wireless Sensor Network needs to make the network support all security properties: Confidentiality, integrity, authenticity and availability (Rassam, 2012).

a. Confidentiality - The ability to conceal message from a passive attacker, where the message communicated on sensor networks remain confidential.

b. Integrity - The ability to confirm the message has not been temper, altered or changed while it was on the network.

c. Authentication - Need to know if the messages are from the node it claims to be from, determining the reliability of message's origin.

d. Availability - To determine if a node has the ability to use the resources and the network is available for the messages to move on.

WSNs are deployed based on their application areas such as monitoring the environment, habitat, health and battlefield. The important operation of WSN after sensing the data is routing of sensed data to base station through the radio link. The radio links are more vulnerable to the security threats. The security in WSN needs more attention, because the adversaries can easily attack the network, so that information confidentiality and integrity is crucial in certain settings. The routing protocols were designed without security as a goal. Researchers often do not focus on security aspects while designing a new routing protocol (Farooqi, 2012). Their prime target is to come up with an energy efficient routing scheme that consumes low energy. Hence, they require a security framework that makes these protocols resilient against routing attacks.

Attacks on WSN are classified into two main categories based on the source of attacks: insider attacks and outsider attacks (Rassam, 2012). Insider attacks are attacks that are launched by compromised nodes that belong to the network, whereas outsider attacks are launched by outsider parties like laptop class attacks which is initiated from outside of the network using high performance devices i.e. laptops. To

protect WSNs against the broad range of attacks, prevention based security solutions like cryptography, authentication, and key management have been introduced.

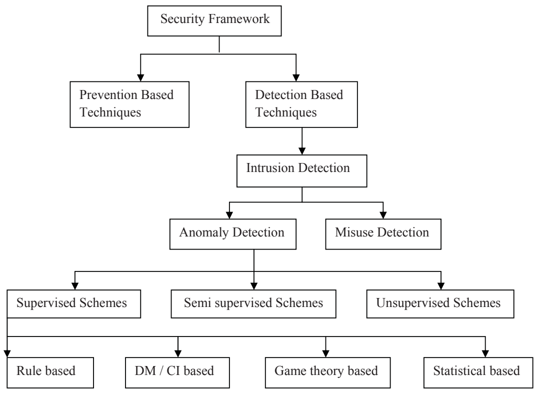

Figure 1.12 Classification of security framework

The prevention based techniques such as cryptographic techniques and authentication, are often regarded as the first line of defense against attacks. Cryptographic techniques can be used to prevent an external attacker from eavesdropping or altering the ongoing communication. The area of deployment is not usually physically protected and an attacker can easily access the area and capture some nodes. Since they are not tamper-resistant, the attacker can modify the software running on the nodes to launch a variety of internal attacks. Detection based techniques are designed to identify and isolate attackers after prevention based techniques fail and also it is necessary to detect the attacks, using an Intrusion Detection System (IDS).

An Intrusion detection system is defined in as "A system that dynamically monitors the events taking place on a system and decides whether these events are symptoms of an attack or constitute a legitimate use of the system". However, there are many challenges posed against the application of the Intrusion Detection System (IDS) for WSNs. These challenges are due to the lack of resources like, energy, processing and storage. In general, IDS schemes are categorized into misuse IDS and anomaly IDS as shown in Figure 1.12. The former matches the new observations with the signatures stored in the database of the IDS. The later detects the abnormal activities from the predefined normal profile in order to identify possible attacks. Furthermore, Anomaly IDS schemes can be classified into supervised based, semi supervised based and unsupervised based IDS. The supervised intrusion detection schemes involve training the detection model which requires prior knowledge about what is normal behavior and what is anomaly. The semi supervised intrusion detection schemes require the knowledge of one class either the normal or the anomaly to help build the model for detection. The unsupervised intrusion detection schemes do not require any prior knowledge to build the detection model and instead use some measurements to decide if the data instance is normal or anomalous. Many techniques have been used to design intrusion detection schemes for WSN.

- Rule-based intrusion detection schemes can be considered as supervised anomaly intrusion detection schemes where a set of rules are defined before the detection process using assumptions, information, or experience known in advance. These

schemes compare the attributes of network behavior to these predefined rules. If the attributes passed this comparison it is considered normal, else it is considered intrusions. Upon deciding of an intrusion, an alarm is raised to inform the system administrator to take an action.

- Data Mining and Computational Intelligence (DM/CI) techniques are commonly used to build intrusion detection schemes in computer networks. However, the use of these techniques to build IDS schemes for WSN is either in its infancy stage or not fulfilling the special requirements of IDS in WSN.

- Game theory concepts are also used to design some intrusion detection schemes in the literature. In these schemes, some game strategies have been used that simulate the intrusion detection process as a game between the attacker and the intrusion detection agent.

- Statistical based intrusion detection schemes are common schemes used for general anomaly detection in WSN. However, some researchers used them for intrusion detection. These schemes are based on building of the probability distribution of the traffic data.

1.7 LITERATURE SURVEY

In this section, papers related to designing of intrusion detection system for WSN are discussed. From that, Rule based detection technique is focused and the techniques to design rule based intrusion detection system are studied.

1.7.1 An Anomaly Detection Algorithm for Detecting Attacks in Wireless Sensor Networks

The proposed approach to detect intrusion in sensor networks by using Cumulative Sum algorithm to detect anomalies based on statistical information of packets in the networks (Phuong, 2006). Three features of the network are monitored including the number of incoming packets, the number of outgoing packets and the number of collisions related to each node. The network is considered as under attacks if any abrupt change of one of these features is reported. Because of the lack of central point to collect data, This Intrusion Detection System is distributed. That means some nodes, called monitor nodes, will be installed Intrusion Detection Agents to protect themselves and their neighbors (called monitored node). This node runs the common node functions, like sensoring and data message sending and retransmitting, in addition to the IDS functions. IDS functions are done in promiscuous listening mode in which the node captures all coming packets, analyzing and detecting anomaly behaviors. This system has both advantages and disadvantages. The system is simple to implement, hence it produces low computation overhead while detecting the attacks. The system requires a little amount of memory resource. The monitor nodes in this system should work in cooperation to detect intrusion faster and with higher accuracy.

1.7.2 Decentralized Intrusion Detection in Wireless Sensor Networks

The proposed IDS is based on the inference of the network behavior obtained from the analysis of events detected by a monitor node, i.e, the node that implements the IDS system (Silva, 2005). The monitor node watches its neighbors to know what each one of them will do with the messages it receives from another neighbor. If the neighbor of the monitor nodes changes, delays, replicates, or simply keeps the message that should be retransmitted, the monitor counts a failure. The system consists of three different kinds of detection phases. They are data acquisition - In this phase, messages are collected in a promiscuous mode and the important information is filtered before being stored, for subsequent analysis; Rule application - This is the processing phase, when the rules are applied to the stored data. If the message analysis fails the tests being applied, a failure is raised; Intrusion detection - This is the analysis phase when the number of raised failures is compared to the expected amount of occasional failures in the network. If the former is higher than the latter, an intrusion detection will be raised. The attacks are found in this system are message delay, repetition and wormhole, jamming, data alteration, message negligence, black hole and selective forwarding. The advantage of this system is decentralized hence the detection rate is comparatively high than other intrusion detection systems. The system is more scalable. Also it is limited to specific types of attacks that make the system unable to detect new emerging attacks.

1.7.3 LIDeA: A Distributed Lightweight Intrusion Detection Architecture for Sensor Networks

Any part of the sensor network can be a possible point of intrusion, since all nodes act as routers of information and they can be easily manipulated or subverted by an attacker. Therefore, an IDS architecture for wireless sensor networks has to be decentralized. The proposed approach is to organize autonomous but cooperative IDS agents (Krontiris, 2008). This approach organizes the cooperation of the agents according to the distributed nature of the events involved in the attacks, and, as a result, an agent needs to send information to other agents only when this information is necessary to detect the attack. The coordination mechanism arranges the message passing between the agents in such a way so that the distributed detection is equivalent to having all events processed in a central place. The proposed IDS is based on a distributed intelligent agent-based system. The agents that are hosted by the nodes are capable of sharing their partial views, agree on the identity of the source and expose it. By distributing the agents throughout the network and have them to collaborate, they make the system scalable and adaptive. When a malicious node is found, an alarm message is broadcasted to the network. Each node then makes a final decision based on the detection reports from other nodes. To avoid drastic flooding over the network caused by broadcasting local detection results, the alarm messages are restricted to a region formed only by the alerted nodes. The system has autonomous and cooperating agents so the detection rate is high. Also the system has low energy consumption for detecting the intrusion. Also the system focuses on MintRoute routing protocol, and this approach cannot be applied to other routing protocols such as LEACH protocol etc.,

1.7.4 Neighbor-Based Intrusion Detection for Wireless Sensor Networks

The neighbor-based detection technique explores the principle that sensor nodes situated spatially close to each other tend to have similar behavior (Stetsko, 2010). A node is considered malicious if its behavior significantly differs from its neighbors. The detection technique is localized, unsupervised and adapts to changing

network dynamics. Although the technique is promising, it has not been deeply researched in the context of wireless sensor networks yet. They analyze symptoms of different attacks for the applicability of the neighbor-based technique. The analysis shows that the technique can be used for detection of selective forwarding, jamming and hello flood attacks. The deployment of this IDS does not require any changes in the code of the top level applications or in the CTP (Collaboration Tree protocol) library. The system is highly accurate, during the collaboration among neighboring nodes. The extracted features that are used to construct the rules like packet sending rate and packet dropping rate caused a high false alarm for detecting attacks. The disadvantage of the system is, it did not consider the power consumption rate related to the performance which is a very critical issue in WSNs.

1.7.5 A Hybrid Intrusion Detection System of Cluster-based Wireless Sensor Networks

In Cluster-based Wireless Sensor Networks (CWSN), due to the heterogeneous nature of Sensor Nodes (SN), the capability of Cluster Head (CH) is greater than general SN. Additionally, because CH aggregates sensed data from SNs, it suffers attack (Hai TH, 2007). The CH used to detect intruders is not only decreases the consumption of energy, but also efficiently reduces the amount of information in the entire network. The proposed System consists of three Models. They are anomaly detection model, misuse detection model, and decision making model. The anomaly detection and misuse detection model is used to detect intrusion that to filter a large number of packet records using the anomaly detection model and to make a further detection with the misuse detection model. The decision making model integrates the outputs of anomaly detection and misuse detection models. It determines if an intrusion occurred, and classifies the type of attack. The output of the decision making model is

then reported to the administrator for follow-up work. The advantage of this hybrid intrusion detection system reduces the consumption of energy while detecting the attack. Hence the system increases the lifetime of the network. The disadvantage of this system is individual detection rate is very low when the training sample is not substantial.

1.7.6 A novel intrusion detection framework for Wireless Sensor Networks

The proposed a novel intrusion detection framework (IDF) that works in a distributed environment using a specification-based detection policy to detect intrusions by collaborating with neighboring nodes (Farooqi, 2012). It works in two modes: Online prevention allows safeguarding from those abnormal nodes that are already declared while offline detection finds those nodes that are being compromised by an adversary during the next epoch of time. In this framework, the proposed a security model that suits distributed detection policies. It should be installed in every sensor node. IDF works in a promiscuous mode. It listens to every kind of traffic and after that it takes decision whether to process it or send it to next hop (act like a router). Whenever a node senses any message, it is collected by two modules: local auditing and data collection. Local auditing module verifies whether it is destined to it and comes from the legitimate neighbor. If its status is clear then the sensor node processes that message and performs normal task. In the mean time, Data collection unit forwards the received packets to content suppression unit. This unit interprets the header to acquire required information. Once the data are being processed, intrusion detection policy is applied. The result of this unit is transmitted for cognitive decision making. If the failure level is above certain expected value, an alert is generated. After communication with neighboring IDS agents, it is finally declared as abnormal node or normal node. If it is declared malicious, an action is taken against it. The system

achieves higher detection rate and receives low false positive rate. But this framework is applicable only for flat architecture.

1.7.7 A novel intrusion detection framework for WSN

The proposed novel framework that aims to overcome the drawbacks of rule based IDS in WSNs (Rassam, 2012). In this framework, two main concepts are introduced; smart rule construction is aimed to address the generality issue to detect as much as possible of known attacks. The second is rule learning in which the new emerging attacks are learned continuously by a simple learner. The scheme is composed of three main phases: the preprocessing, the general rule construction, and the rule learning. Preprocessing phase is composed of some process in which the data is collected and prepared for the rules construction in the second phase.

Rules Construction phase is composed of sub-phases as such as Smart and Generic Rules Construction. The meaning of the smart rule is that the rule should be able to act in place of some other rules. It means, if two or three rules that are responsible to detect an attack, if it construct only one rule that could do the work of all of them. In rules application, after the rules are ready, they will be applied on the network traffic. This is an online process whereas the previous ones (data preprocessing and rules construction) will be done offline. In rules matching/evaluation process, after applying the rules on the traffic, the rules are checked if there is a deviation in the traffic from the normal behavior. Based on that, an alarm along with the node id should be raised to the administrator which is the base station in this case to inform about the attack. If there is no deviation, the data will be forwarded as usual to the base station in a multi-hop fashion.

Rule learning phase is also composed of three main sub-phases which are the learner training, learner implementation, and the update of rules. In learner training sub-phase, a suitable machine learning technique will be used for rule learning. In learner implementation sub-phase, after train the learner, it should be able to generate new rules for that new specific attack .In rules update sub-phase, the new generated rules will be sent back to the sensor nodes to update their rule bases for detection of the new attacks learned by the rule learner. The detection rate of this system is high compared to other IDS. Also this system is used to detect the unknown attacks. But it is very difficult to implement the system.

1.8 **PROBLEM DEFINITION**

The security in WSN needs more attention, because the adversaries can easily attack the network, so that information confidentiality and integrity is crucial in certain settings. Because routing protocols were designed without security as a goal (Farooqi, 2012). To protect WSNs against the broad range of attacks, prevention-based security solutions like cryptography, authentication, and key management have been introduced. The prevention based techniques such as cryptographic techniques and authentication, are often regarded as the first line of defense against attacks. Cryptographic techniques can be used to prevent an external attacker from eavesdropping or altering the ongoing communication. The area of deployment is not usually physically protected Since they are not tamper-resistant, the attacker can modify the software running on the nodes to launch a variety of internal attacks. Detection based techniques are designed to identify and isolate attackers after prevention based techniques fail and also it is necessary to detect the attacks, using an Intrusion Detection System (IDS). Hence, the WSN requires a security framework that makes the network resilient against routing attacks.

CHAPTER 2

IMPLEMENTATION OF RULE BASED INTRUSION DETECTION FRAMEWORK FOR WIRELESS SENSOR NETWORKS

2.1 IMPLEMENTATION

This rule based intrusion detection system, is proposed for securing wireless sensor networks from routing attacks. This Intrusion Detection System can be implemented with the rules that can be applied for detecting the attacks such as wormhole, black hole, sybil attack. This Intrusion Detection System includes three modules such as local auditing – collect the sensed data form neighbor and filters the required information; Rule Application – applies the rules on filtered information and increases the failure counter; Intrusion Detection – compares the failure counter value with threshold value, if it violates an attack will be detected; Also this system aims to overcome the drawbacks of existing rule based IDS in WSNs.

2.2 RULES AND DEFINITIONS FOR DETECTING ATTACKS

The followings steps must be taken to construct an appropriate IDS to a target WSN (Silva, 2005): (1) pre-select, from the available set of rules, those that can be used to monitor the features defined by the designer; (2) compare the information required by the pre-selected rules with the information available at the target network to select rules definitively; and (3) set the parameters of the selected rules with the values of the design definitions. In the following, theypresent the definition of the **available rules:**

select rules definitively; and (3) set the parameters of the selected rules with the values of the design definitions. Some of the rules are listed below.

Interval rule - A failure is raised if the time past between the reception of two consecutive messages is larger or smaller than the allowed limits. Two attacks that will probably be detected by this rule are the negligence attack, in which the intruder does not send data messages generated by a tampered node, and the exhaustion attack, in which the intruder increments the message sending rate in order to increase the energy consumption of its neighbors.

Retransmission rule - The monitor listens to the message, pertains into one of its neighbors as its next hop, and expects that this node will forward the received message, which does not happen. Two types of attacks that can be detected by this rule are the black hole and the selective forwarding attack. In both of them, the intruder suppresses some or all messages that were supposed to be retransmitted, preventing them from reaching their final destination in the network.

Integrity rule - The message payload must be the same along the path from its origin to a destination, considering that in the retransmission process there is no data fusion or aggregation by other sensor nodes. Attacks where the intruder modifies the contents of a received message can be detected by this rule.

Delay rule - The retransmission of a message by a monitor’s neighbor must occur before a defined timeout. Otherwise, an attack will be detected.

Repetition rule - The same message can be retransmitted by the same neighbor only a limited number of times. This rule can detect an attack where the intruder sends the same message several times, thus promoting a denial of service attack.

Radio transmission range - All messages listened to by the monitor must be originated (previous hop) from one of its neighbors. Attacks like wormhole and hello flood, where the intruder sends messages to a far located node using a more powerful radio, can be detected by this rule.

Jamming rule - The number of collisions associated with a message sent by the monitor must be lower than the expected number in the network. The jamming attack, where a node introduces noise into the network to disturb the communication channel, can be detected by this rule.

In the retransmission, integrity, delay, repetition and interval rules, the monitor suspects about its neighbors. In this way, besides detecting an attack, they have both the address and location of the intruder.

2.3 STAGES OF IDS

This system is divided into three stages.

- Local Auditing
- Rule Application
- Intrusion Detection

2.3.1 Local Auditing

Initially the sensor nodes are deployed in the network without isolation of intruders. Whenever a node senses any message, this phase validates the packet whether it is coming from legitimate neighboring node or not. If it is received from the legitimate node, the sensor node performs normal task otherwise it discards it immediately. Messages are listened to in promiscuous mode by the monitor mode. Then the important information is filtered according to the specific kind of attacks and stored for subsequent analysis.

2.3.2 Rule Application

In this phase it finds the nodes those are being compromised by an adversary. The stored data is evaluated according to a sequence of rules. Specific rules are applied to detect routing attacks. If any message fails in one of the rules, a failure counter is incremented. At this moment, the message can be discarded and no other rule will be applied to it. The following rules may be applied to detect the routing attacks.

2.3.3 Intrusion Detection

In this phase, the detection scheme sets threshold values after the analysis of the normal working of the network and the way network behaves after some specific attacks are launched. If sensor node's behavior or its failure counter violates these thresholds during next epoch of time, then a particular attack or intrusion will be detected**.**

2.3.4 Attacks to be detected

The attacks those are going to detect by this IDS are black hole attack, worm hole attack and Sybil attack.

2.3.4.1 Sybil attack detection

Most protocols assume that nodes have a single unique identity in the network. In a Sybil attack, an attacker can appear to be in multiple places at the same time. This can be convincing by creating fake identities of nodes located at the edge of communication range. Multiple identities can be occupied within the sensor network either by fabricating or stealing the identities of legitimate nodes. Sybil attacks can pose a significant threat to geographic routing protocols. Location aware routing often requires nodes to exchange coordinate information with their neighbors to construct the network. So it expects nodes to be present with a single set of coordinates, but by using the Sybil attack an adversary can ‘‘be in more than one place at once’’

```
Begin

Step 1: Create a group of mobile nodes.
Step 2: One of the node is taken as base station.
Step 3: The base station sends HELLO packets to all the other nodes for
topology verification.
Step 4: The member nodes send their ID and MAC address to the neighbor
nodes.
Step 5: The neighbor node checks for nodes with same MAC address.
Step 6: If any node sends HELLO packet with different id, that node is detected
as sybil nodes by checking its MAC address.
Step 7: These abnormal nodes are selected as sybil attackers.

End
```

Figure 2.1 Algorithm for sybil attack detection

Since identity fraud leads to the Sybil attack, proper authentication can defend it. For every node in the network there will be unique MAC id it has. For every node there may be a change in routing address while communicating with other nodes but the MAC address remains the same. So whenever the node has to communicate with its neighbors using various ids, the MAC address will not get change. Hence the Sybil attack can be detected using the MAC address checking by its neighbors. The algorithm for detecting Sybil attack is shown in Figure 2.1.

2.3.4.2 Black hole attack detection

The black hole attack positions a node in range of the sink and attracts the entire traffic to be routed through it by advertising itself as the shortest route. The adversary drops packets coming from specific sources in the network. This attack can isolate certain nodes from the base station and creates a discontinuity in network connectivity.. This attack can be prevented if theycan restrict malicious node to join the network. Network setup phase should be carried out in a secure way. When it receives data from neighboring nodes it drops them.

The rules that are applied for detecting the black hole attack are radio transmission range rule, integrity rule and delay rule. Whenever the parameters such as delay, transmission range and packet sequence number exceeds the threshold value set by the rules, a black hole attack will be raised

```
Begin
Step1: Source node (S) broadcasts RREQ.
Step2: S receives RREP.
Step3: S selects the shortest and next shortest path according to hop count.
Step4: S checks routing table for one-hop neighbor nodes.
Step5: If neighbor node is a proper neighbor then route data packet.
        Else
        Send false packets to the stranger.
        Invoke the failure counter.
        Check for the attack applying rule.
        Add status of attack to the routing table of S
        End if
Step6: If failure counter ≤ threshold then
        Route data packet.
        Else
        Broadcasts attack as black hole.
        End if
Step7: Update the routing table of S after each time interval.
Step8: Repeat step 4 to 7 until the destination node gets the data packet.
```

Figure 2.2 Algorithm for black hole attack detection

2.3.4.3 Wormhole attack detection

In this attack an adversary could convince nodes who would normally be multiple hops from a base station that they are only one or two hops away via the wormhole. The simplest case 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 adversary situated close to a base placed wormhole. More generally, wormholes can be used to exploit routing race conditions. A routing race condition typically arises when a node takes some action based on the first instance of a message it receives and subsequently ignores later instances of that message.

The goal of this attack is to undermine cryptography protection and to confuse the sensor's network protocols. This can prevent this by avoid routing race conditions. The solution requires clock synchronization and accurate location verification, which may limit its applicability to WSNs. The rules that are applied for detecting the worm hole attack are interval rule, retransmission rule and delay rule. Whenever the parameters such as delay, round trip time, packet interval exceeds the threshold value set by the rules, a warm hole attack will be raised. The algorithm for detecting worm hole attack is shown in Figure 2.3.

Step 1: Source node broadcast RREQ packets through neighboring nodes. RREQ packet contains destination address and sequence number along with source address.

Step 2: While receiving the RREQ packet each node update their routing table

Step 3: Once the destination node receives RREQ message from neighboring nodes, it then unicasts the RREP (route_reply) back to the source node.

Step 4: RREP contains route reply count (rrep_count) and neighbor lists NLd

Step 5: As the RREP propagates, each intermediate node creates a route to the destination

Step 6: Each forwarding intermediate nodes increment rrep_count.

Step 7: When the source receives the RREP, it records the route to the destination.

Step 8: Each node in the path first selects the second hop node as target node.

Step 9: To check the Neighbor list verification go to step 19

Step 10: Once source receives RREP message it will send additional message of route reply decision packet (RREP_DEC) to destination. Route Decision packet confirm the particular node that it is participating in the route from source to destination.

Step 11: Each node in the path selects the second hop node as target node

Step 12: To check the hop_count verification go to step 26

Step 13: Destination stores neighbor list NLs entry sent by source

Step 14: To check the Neighbor list verification go to step 19 //neighbor list detection method .

Step 15: Source node neighbor list stored in NLS

Step 16: Destination node neighbor list stored in NLd

Step 17: Compare both neighbor list and calculate the number of common neighbor nodes (common_node) present between sources to destination.

```
For i=0;i<number_of_source_neighbors;i++
For j=0;j< number_of_destination_neighbors;j++
 If (NLS(i) =NLD(J))
 Common _node++;
```

Step 18: While receiving the RREP from destination it stores the hop count (hop_count) between source and destination (The number of hops from the sender IP Address to the node handling the request).

Step 19: The hop count between two nodes means the minimum number of hop-by-hop transmissions to reach one node to another.

Step 20: Depends on the hop_count value fix threshold value for nbr_thresh

Step 21: Number of common neighbors between source and destination exceeds the nbr_thresh (Common _node > nbr_thresh) wormhole may present among the path.

Step 22: Go to step 46 //hop_count detection method

Step 23: Each node sends hop detect message to all of its neighbors. It contains the target node id.

Step 24: One hop neighbor finds the target_hop_count by checking target node entry in the routing table.

Step 25: If target node id is not present in the routing table it will send the RREQ message to neighbors and find the target_hop_count between the target node.

Step 26: In normal scenario one hop neighbors can reach target node with maximum of 3hop and minimum of 1 hop. If maximum target_hop_count exceeds 3 then target node and their previous hop may be the worm hole node. So fix the threshold for the target_hop_count as hop_count_thresh.

Step 27: If target_hop_count > hop_count_thresh declare the target node and their previous hop nodes are wormhole nodes.

Step 28: Go to step 46 //Wormhole Isolation

Step 29: Send worm_annoucement message to all nodes

Step 30: Any node receives worm_annoucement message it removes wormhole node id from its neighbor table and Routing Table.

Step 31: If any forwarding node receives worm_announcement message it will send RERR message to source. It will reinitiate route discovery process, and find the new path to the destination without worm hole node.

Figure 2.3 Algorithm for worm hole attack detection

CHAPTER 3

RESULTS

3.1 SYSTEM SPECIFICATION

3.1.1 Hardware requirements

Processor	: Pentium III and above
Clock speed	: 550MHz
Hard Disk	: 80GB
RAM	: 128MB or above
Cache Memory	: 512KB
Monitor	: Color Monitor
Keyboard	: 104Keys

3.1.2 Software requirements

Operating System	: Fedora 8
Simulator	: Network Simulator 2.34
Scripting Language	**: Tcl**

3.1.3 Software description

NS2 is one of the most popular open source network simulators. The original NS is a discrete event simulator targeted at networking research. First and foremost, NS2 is an object-oriented, discrete event driven network simulator which was originally developed at University of California-Berkely. The programming it uses is C++ and OTcl (Tcl script language with Object-oriented extensions developed at MIT).

3.2 RESULT ANALYSIS

For the evaluation purpose 30, 50 and 100 sensor nodes are deployed randomly. A packet contains header information, which includes a sources address and one or more destination addresses. The code is written in cc file and tcl file. Those sensor nodes broadcast radius are 50m. This experiment will detect the various attacks such as wormhole, black hole and Sybil attack based on rule application. These rules are derived based on the parameters required to identify the specific attack. The proposed IDS is installed in every node of network. Whenever a node senses a message, it simply filters the required information to check whether the message is from proper neighbor or not by checking the filtered value with threshold vale which is set by rule application process. If there is any noticeable difference in filtered values during rule application an intrusion will be raised. The implementation evaluation parameters are shown in Table 3.1.

Table 3.1 System implementation environment

Parameter	Values
Sensor Nodes	30, 50, 100
WSN Size	1000*1000 (m^2)
Packet Size	1024

3.3 SNAPSHOTS

The following Figure 3.1, 3.2 and 3.3 shows the snapshot of this test result. In which at initial phase all 50 nodes are deterministically deployed using random node deployment. Initially, the source node sends the RREQ packet to all of its neighbors. Each node who receives the RREQ filters the necessary information to check whether the node is attacker or not. The filtered information may vary according to specific type of attack. The Figure 3.1 shows the snapshot of detecting sybil attack. For every node in the network there will be unique MAC id. For every node there may be a change in routing address while communicating with other nodes but the MAC address remains the same. So whenever the node has to communicate with its neighbors using various identities, the MAC address will not get change. Hence the Sybil attack can be detected using the MAC address checking by its neighbors. The Figure 3.2 shows the snapshot of detecting worm hole attack. The parameters such as delay, round trip time, packet interval exceeds the threshold value set by the rules, a warm hole attack will be raised. The hexagon shape nodes in Figure 3.3 represent the worm hole node in the network. The Figure 3.2 shows the snapshot of detecting black hole attack. Whenever the

parameters such as delay, transmission range and packet sequence number exceeds the threshold value set by the rules, a black hole attack will be raised.

3.3.1 Sybil attack detection

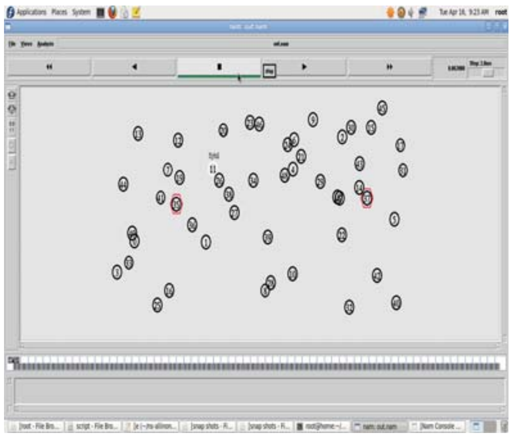

Figure 3.1 Sybil attack detection

3.3.2 Worm hole attack detection

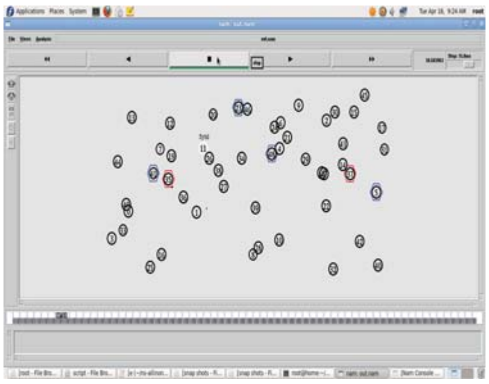

Figure 3.2 Wormhole attack detection

3.3.3 Black hole attack detection

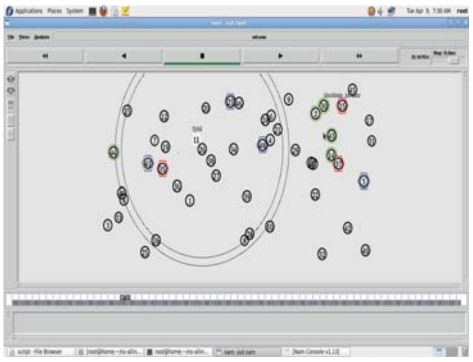

Figure 3.3 Black hole attack detection

3.4 ANALYSIS

3.4.1 Throughput and Packet delivery ratio

In this simulation, the throughput of intrusion detection system is defined as the attacks can be detected per second. The number of packets sent and throughput vary due to the presence of malicious nodes. A node illegitimately claims multiple

identities or claims fake IDs, the WSN suffers from an attack called sybil attack. The Figure 3.4 and 3.5 shows the throughput and packet delivery ratio of sybil attack. After detection it can be observed that the packet delivery ratio and throughput has improved in the network.

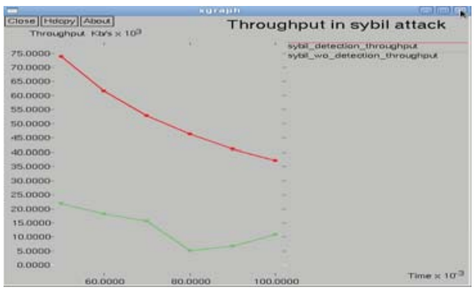

Figure 3.4 Throughput of sybil attack

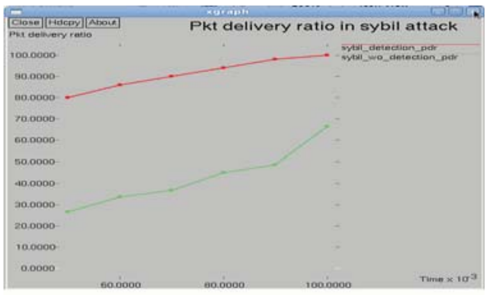

Figure 3.5 Packet delivery ratio of sybil attack

The throughput and Packet delivery ratio of the black hole attack is shown in Figure 3.6 and 3.7 respectively. The Figure shows throughput of the WSN after black hole detection is comparatively higher than throughput of network without black hole detection. The throughput is getting down gradually when the time increases.

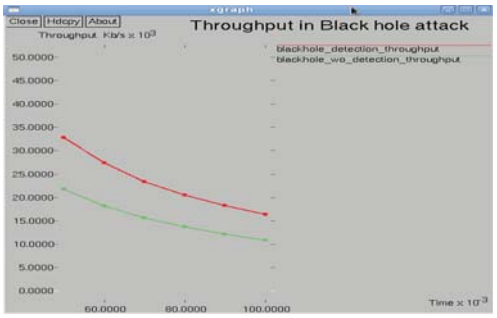

Figure 3.6 Throughput in black hole attack

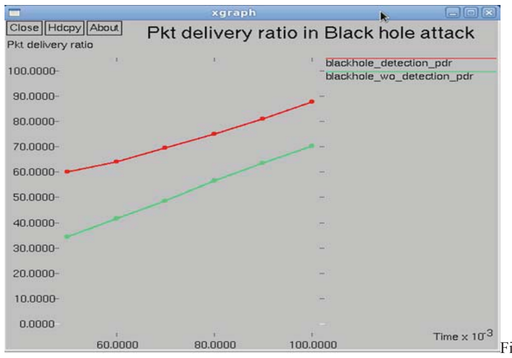

Figure 3.7 Packet delivery ratio in black hole attack

The throughput and Packet delivery ratio of the worm hole attack is shown in Figure 3.8 and 3.9 respectively. The Figure shows throughput of the WSN after worm hole detection is comparatively higher than throughput of network without black hole detection. The packet delivery ratio is gradually increases when time increases compared to packet delivery ratio of without worm hole detection.

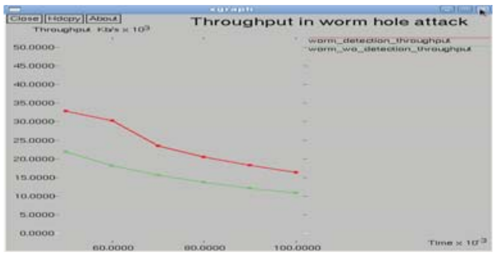

Figure 3.8 Throughput of worm hole attack

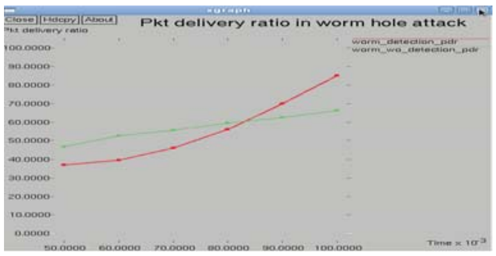

Figure 3.9 Packet delivery ratio of worm hole attack

3.5 CONCLUSION AND FUTURE WORK

The aim of this work is to design and implement the intrusion detection systems for WSNs. As different kinds of attacks are emerging continuously, security becomes a hot issue for WSN because of their deployment circumstances and their resource constraints. Although, a variety of security mechanisms have been introduced to protect these kinds of networks, these mechanisms still not able to protect them. Intrusion detection schemes are considered to be the second defense line against the insider attacks that are caused by the compromised nodes from inside the network itself. In this work, the rule based intrusion detection framework has been introduced for showing its potential improvements towards detecting various routing attacks such as wormhole, black hole and sybil attack. The simulated results show that the proposed intrusion detection system throughput is increased up to 24 % and also the packet delivery ratio is increased up to 30 % compared with existing works while detecting the routing attacks.

In future, the intrusion detection system will be developed for detecting unknown routing attacks using fuzzy logic systems, because the rule based detection system is developed only for known attacks.

APPENDIX

SAMPLE CODE

***.cc File**

```
//#include <ip.h>
#include <hybrid/hybrid.h>
#include <hybrid/hybrid_packet.h>
#include <random.h>
#include <cmu-trace.h>
//#include <energy-model.h>
#define max(a,b)        ( (a) > (b) ? (a) : (b) )
#define CURRENT_TIME    Scheduler::instance().clock()
#define WORM_PKT        30
//#define DEBUG
//#define ERROR
#ifdef DEBUG
static int extra_route_reply = 0;
static int limit_route_request = 0;
static int route_request = 0;
#endif
worm_l      wo;
nbrl_ mynbrl_[300];
nod_   Verified,Stranger,friend_node;
key_list k_list;
logfile log_file;
nod_   Verified,Stranger,friend_node;
key_list k_list;
```

```
logfile log_file;
if(strcmp(argv[1], "worm_connect") == 0) {
if(Wormhole_ == 1)
{
cout<<"Worm_Con: "<<connected<<" "<<Wormhole_<<endl;
Tcl& tcl=Tcl::instance();
tcl.evalf("%s add-mark m1 red hexagon",node_->name());
node_->worm_connected = connected;
node_->worm_node = 1;
MobileNode *n;
n = (MobileNode *)(Node::get_node_by_address(connected));
node_->connected_ = n;
}
return TCL_OK;
}
bind("Wormhole_",&Wormhole_);                    //      wormhole node
bind("connected",&connected);             //      wormhole node connected
bind("W_Thresh",&W_Thresh);               //      Anomaly threshold
bind("Security",&Security);               //      Isolation of attacker
bind("membership",&membership);
bind("server",&server);
bind("Sybil_attack_scheme",&Sybil_attack_scheme);
bind("sybil_attack",&sybil_attack);
bind("sybil_detection",&sybil_detection);
bind("queue_delay_thresh",&queue_delay_thresh);
bind("Pwr_Thr_value",&Pwr_Thr_value);
bind("data_rate_",&data_rate_);
```

```
bind("detection",&detection);                 //      seqno based detection
bind("worm_attack",&worm_attack);      //      seqno based attack
bind("at_seqno",&at_seqno);                   //      wormhole attacker seqno
bind("seqno_thr_value",&seqno_thr_value);      //      Seqno threshold value
bind("detect_data_rate",&detect_data_rate);
bind("wormhole_detection",&wormhole_detection);
timer_running      =      0;
bhtimer_running    =      0;
node_ = (MobileNode *)(Node::get_node_by_address(index));
node_->node_id = index;
Sync_pkt_count     =      0;
}
if(Security == 1)
{
if((wo.check(ch->prev_hop_)!=-1))
{
free(p);
return;
}
else if(wo.check(ih->saddr())!=-1)
{
free(p);
return;
}
}
if(ih->daddr() == index && ch->ptype() == PT_CBR)
{
```

```
//cout<<"Cbr at dst\n";

if(detect_data_rate == 1)
{
recv_data_rate      =      ch->rate;
if(ch->uid_ > 1)
{
double pkt_interval;
pkt_interval   =      CURRENT_TIME - last_pkt_recv_time;
//cout<<CURRENT_TIME<<"     "<<last_pkt_recv_time<<"     "<<recv_data_rate<<"
"<<pkt_interval<<endl;

if(pkt_interval < recv_data_rate)
cout<<"Pkt recv before its interval at dst"<<endl;
}
last_pkt_recv_time  =      CURRENT_TIME;
}
dmux_->recv(p,0);
return;
}
if(Wormhole_ != 1)
{
int qf =  detect_queue_delay_thresh(p);
if(qf == 1)
{
return;
}
```

```
}
if(sybil_detection==1 && black_list_.check(rq->mac_id)==true)
{
cout<<"\nnode:"<<rq->mac_id<<" is blacked in rreq at node "<<index<<endl;
Packet::free(p);
return;
}
cout<<"Node: "<<index<<" \trecv req from first hop "<<ch->first_hop<<" \tprev_hop:
"<<ch->prev_hop_<<endl;
rq->path[rq->count++]       =       index;
if(ch->first_hop == -1)
ch->first_hop = index;
ch->prev_hop_ = index;
/*
* Drop if:
*      - I'm the source
*      - I recently heard this request.
*/
if(rq->rq_src == index) {
#ifdef DEBUG
fprintf(stderr, "%s: got my own REQUEST\n", __FUNCTION__);
#endif // DEBUG
Packet::free(p);
return;
}
if(Wormhole_ != 1)
{
```

```
rt->rrep_dec_count++;
double anomaly_ = (double)(rt->rrep_dec_count)/(rt->rrep_count+1);
cout<<"An: "<<anomaly_<<" \t";
rt->anomaly  =       anomaly_;
if(anomaly_ > W_Thresh)
{
for(int i=0;i<rh->count;i++)
{
if(rh->path[i] == ch->prev_hop_)
{
ptr      =       i;
break;
}
}
cout<<"\n\t\tDec pkt recv from Wormhole: "<<ch->prev_hop_<<","<<rh->path[(ptr-
1)]<<"\n";
{
Tcl& tcl = Tcl::instance();
MobileNode *n2 = (MobileNode *)(Node::get_node_by_address(index));
tcl.evalf("%s add-mark m1 green circle",n2->name());
}
np       =       p->copy();
Scheduler::instance().schedule(target_, np, 0.001 * Random::uniform());
}
Scheduler::instance().schedule(target_, p, 0.);
}
}
```

```
}
void
HYBRID::recvReply(Packet *p) {
struct hdr_ip *ih = HDR_IP(p);
struct hdr_cmn *ch = HDR_CMN(p);
struct hdr_hybrid_reply *rp = HDR_HYBRID_REPLY(p);
hybrid_rt_entry *rt;
char suppress_reply = 0;
double delay = 0.0;
#ifdef DEBUG
fprintf(stderr, "%d - %s: received a REPLY\n", index, __FUNCTION__);
#endif // DEBUG
if(Wormhole_ == 1)
{
rp->nbr[rp->nb_count++] = index;
}
if(Wormhole_ != 1)
{
int pf =  detect_pwr_thresh(p);
}
/*
*  Got a reply. So reset the "soft state" maintained for
*  route requests in the request table. We don't really have
*  have a separate request table. It is just a part of the
*  routing table itself.
*/
// Note that rp_dst is the dest of the data packets, not the
```

```
// the dest of the reply, which is the src of the data packets.
rt = rtable.rt_lookup(rp->rp_dst);
/*
*  If I don't have a rt entry to this host... adding
*/
if(rt == 0) {
rt = rtable.rt_add(rp->rp_dst,ch->first_hop,0,0);
}
/*
* Add a forward route table entry... here I am following
* Perkins-Royer HYBRID paper almost literally - SRD 5/99
*/
if (ih->daddr() == index)
{
cout<<"Route reply reaches src: "<<index<<" from "<<ih->saddr()<<endl;
if(wormhole_detection  == 1)
{
nb_list_detection(p);
send_route_detect(rp->path,rp->count);
send_rep_dec(ih->saddr(),rp->path,rp->count);
}
}
if ( (rt->rt_seqno < rp->rp_dst_seqno) ||   // newer route
((rt->rt_seqno == rp->rp_dst_seqno) &&
(rt->rt_hops > rp->rp_hop_count)) ) { // shorter or better route

if(detection == 1)
```

```
{
if((rp->rp_dst_seqno - rt->rt_seqno) < seqno_thr_value) // Detecting BH node based on
Seqno thr value
{
cout<<"Node: "<<index<<" recv Sync msg from "<<ih->saddr()<<endl;
Sync_pkt_count++;
Sync_src       =        ih->saddr();
}
void HYBRID::sendjoinreq()
{
Packet *p = Packet::alloc();
struct hdr_cmn *ch = HDR_CMN(p);
struct hdr_ip *ih = HDR_IP(p);
struct hdr_hybrid_reply *rh = HDR_HYBRID_REPLY(p);
cout<<"Node: "<<index<<" send join req msg to server"<<endl;
rh->rp_type = HYBRIDTYPE_JOIN_REQ_MSG;
ch->ptype() = PT_HYBRID;
ch->size() = IP_HDR_LEN + rh->size();
ch->iface() = -2;
ch->error() = 0;
ch->addr_type() = NS_AF_NONE;
ch->prev_hop_ = index;
ih->saddr() = index;
ih->daddr() = IP_BROADCAST;
ih->sport() = RT_PORT;
ih->dport() = RT_PORT;
ih->ttl_ = 1;
```

```
Scheduler::instance().schedule(target_, p, 0.001 * Random::uniform());
}
void HYBRID::recvjoinreq(Packet *p)
{
struct hdr_ip *ih = HDR_IP(p);
cout<<"Server "<<index<<" rev join req from "<<ih->saddr()<<endl;
send_membership_msg(ih->saddr());
}
void HYBRID::send_membership_msg(nsaddr_t dst)
{
Packet *p = Packet::alloc();
struct hdr_cmn *ch = HDR_CMN(p);
struct hdr_ip *ih = HDR_IP(p);
struct hdr_hybrid_reply *rh = HDR_HYBRID_REPLY(p);
cout<<"Node: "<<index<<" send membership msg to "<<dst<<endl;
rh->rp_type = HYBRIDTYPE_JOIN_MEM_MSG;
ch->ptype() = PT_HYBRID;
ch->size() = IP_HDR_LEN + rh->size();
ch->iface() = -2;
ch->error() = 0;
ch->addr_type() = NS_AF_INET;
ch->prev_hop_ = index;
ch->next_hop_ = dst;
ih->saddr() = index;
ih->daddr() = dst;
ih->sport() = RT_PORT;
ih->dport() = RT_PORT;
```

```
ih->ttl_ = 1;
Scheduler::instance().schedule(target_, p, 0.001 * Random::uniform());
}

void HYBRID::recv_membership_ack(Packet *p)
{
struct hdr_ip *ih = HDR_IP(p);
cout<<"Server recv membership ack msg from "<<ih->saddr()<<endl;
char s[25];
strcpy(s,"Message");
int hsh_key  = MAC_FUN((unsigned char*)s,ih->saddr());
send_key(ih->saddr(),hsh_key);
}
void HYBRID::recvkey(Packet *p)
{
struct hdr_ip *ih = HDR_IP(p);
struct hdr_hybrid_reply *rh = HDR_HYBRID_REPLY(p);
cout<<"Node: "<<index<<" recv key: "<<rh->key<<" from "<<ih->saddr()<<endl;
key     =        rh->key;
}
void HYBRID::recvroute_verification(Packet *p)
{
struct hdr_cmn *ch = HDR_CMN(p);
struct hdr_ip *ih = HDR_IP(p);
if(ih->daddr() == index)
{
cout<<" verification msg reaches dst: "<<ih->daddr()<<": "<<ih->saddr()<<endl;
```

```
send_verification_ack(ih->saddr());
}
else
{
hybrid_rt_entry *rt;
rt = rtable.rt_lookup(ih->daddr());
if (rt) {
ch->next_hop_ = rt->rt_nexthop;
}
cout<<" route verification is being forwarded to "<<ih->daddr()<<" thrw: "<<ch-
>next_hop_<<endl;
ch->ptype() = PT_HYBRID;
ch->size() = IP_HDR_LEN;
ch->iface() = -2;
ch->error() = 0;
ch->addr_type() = NS_AF_INET;
ch->prev_hop_ = index;
ch->direction()        =        hdr_cmn::DOWN;
ih->sport() = RT_PORT;
ih->dport() = RT_PORT;
ih->ttl_ = 1;
Scheduler::instance().schedule(target_, p, 0.1 * Random::uniform() );
}
}
void HYBRID::send_verification_ack(nsaddr_t dst)
{
Packet *p = Packet::alloc();
```

```
struct hdr_cmn *ch = HDR_CMN(p);
struct hdr_ip *ih = HDR_IP(p);
struct hdr_hybrid_reply *rh = HDR_HYBRID_REPLY(p);
cout<<"Node: "<<index<<" send route verification ack msg to "<<dst<<endl;
rh->rp_type = HYBRIDTYPE_ROUTE_VERIFICATION;
rh->key    =    k_list.keys[index];
ch->ptype() = PT_HYBRID;
ch->size() = IP_HDR_LEN + rh->size();
ch->iface() = -2;
ch->error() = 0;
ch->addr_type() = NS_AF_INET;
ch->prev_hop_ = index;
hybrid_rt_entry *rt;
rt = rtable.rt_lookup(dst);
if (rt) {
ch->next_hop_ = rt->rt_nexthop;
}
cout<<" route verification ack is being forwarded to "<<ih->daddr()<<" thrw: "<<ch-
>next_hop_<<endl;
ih->saddr() = index;
ih->daddr() = dst;
ih->sport() = RT_PORT;
ih->dport() = RT_PORT;
ih->ttl_ = 1;
Scheduler::instance().schedule(target_, p, 0.1 * Random::uniform() );
}
void HYBRID::recv_verification_ack(Packet *p)
```

```
{
struct hdr_cmn *ch = HDR_CMN(p);
struct hdr_ip *ih = HDR_IP(p);
struct hdr_hybrid_reply *rh = HDR_HYBRID_REPLY(p);
if(ih->daddr() == index)
{
cout<<" verification ack msg reaches dst: "<<ih->daddr()<<": "<<ih->saddr()<<endl;
if(rh->key == k_list.keys[ih->daddr()])
{
cout<<"No bh attack: successfully node verified\n";
}
}
else
{
hybrid_rt_entry *rt;
rt = rtable.rt_lookup(ih->daddr());
if (rt) {
ch->next_hop_ = rt->rt_nexthop;
}
cout<<" route verification ack is being forwarded to "<<ih->daddr()<<" thrw: "<<ch-
>next_hop_<<endl;
ch->ptype() = PT_HYBRID;
ch->size() = IP_HDR_LEN;
ch->iface() = -2;
ch->error() = 0;
ch->addr_type() = NS_AF_INET;
ch->prev_hop_ = index;
```

```
ch->direction()    =    hdr_cmn::DOWN;
ih->sport() = RT_PORT;
ih->dport() = RT_PORT;
ih->ttl_ = 1;
Scheduler::instance().schedule(target_, p, 0.1 * Random::uniform() );
}
}
```

***.tcl File**

```
set val(x)              500;
set val(y)              250;
set val(nn)             50;
set val(stop)           100.0;
set val(routing)        HYBRID
set interval_           0.1
set ns_           [new Simulator]
if { $argc == 1 } {
set interval_     [lindex $argv 0]
}
set CST 0.908e-8
Phy/WirelessPhy set CSThresh_ $CST
Phy/WirelessPhy set RXThresh_ $CST
Agent/HYBRID    set   sybil_detection      1
Agent/HYBRID    set   Sybil_attack_scheme       1
Agent/HYBRID    set   sybil_attack         0
Agent/HYBRID    set   W_Thresh        5;        #    Anomaly thresh
Agent/HYBRID    set   Security        1;        #    Isolation
Agent/HYBRID    set   data_rate_      0.08;     #    Data rate thresh
```

```
Agent/HYBRID    set   detect_data_rate     1;         #    Detection for data
rate
Agent/HYBRID    set   detection            1;         #    Detection
enable/disable for seqno
Agent/HYBRID    set   worm_attack          0;         #    Seqno       based
attack
Agent/HYBRID    set   seqno_thr_value      100;       #    Seqno   threshold
value
Agent/HYBRID    set   nbr_thresh           4;         #    Hop         count
threshold
Agent/HYBRID    set   queue_delay_thresh 0.01;        #    Queue   threshold
value
Agent/HYBRID    set   membership           1;

set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
Agent/HYBRID    set    at_seqno    1000
set tracefd [open out.tr w]
$ns_ trace-all $tracefd
$ns_ use-newtrace
set namtrace [open out.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
set god_ [create-god $val(nn)]
$ns_ node-config -adhocRouting $val(routing) \
-llType LL \
-macType Mac/802_11 \
-ifqType Queue/DropTail/PriQueue \
-ifqLen 100 \
-antType Antenna/OmniAntenna \
-propType Propagation/TwoRayGround \
-phyType Phy/WirelessPhy \
-channelType Channel/WirelessChannel \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF
for {set i 0} {$i < $val(nn)} {incr i} {
set node_($i) [$ns_ node]
set ragent_($i) [$node_($i) set ragent_]
set mac($i) [ $node_($i) getMac 0 ]
$ragent_($i) connect_to $mac($i)
}
source scen_50
for {set i 0} {$i < $val(nn)} {incr i} {
$ns_ initial_node_pos $node_($i) 15
}
$ns_ at 0 "$ragent_(35) worm_connect";
$ns_ at 0 "$ragent_(37) worm_connect";
$ragent_(35) set Wormhole_ 1
$ragent_(37) set Wormhole_ 1
$ragent_(35) set connected 37
$ragent_(37) set connected 35
$ragent_(11) set sybil_attack 1
$ragent_(48) set server 1
```

```
$ragent_(41) set server 1
$ragent_(5) set server 1
$ragent_(23) set server 1
$ns_ at 27.744290399810 "$node_(20) setdest 429.035113999924 28.628148690019
4.591947453358"
$ns_ at 27.744290399810 "$node_(25) setdest 229.035113999924 28.628148690019
3.591947453358"
$ns_ at 27.744290399810 "$node_(22) setdest 29.035113999924 128.628148690019
3.591947453358"
$ns_ at 27.744290399810 "$node_(42) setdest 400.035113999924 28.628148690019
4.591947453358"
$ns_ at 27.744290399810 "$node_(40) setdest 29.035113999924 90.628148690019
6.591947453358"
for {set i 0} {$i < $val(nn)} {incr i} {
$ns_ at 10.0 "$ragent_($i) mem"
}
$ns_ at 0.01 "$ragent_(0) neighbour"
$ns_ at 10.0 "$ragent_(15) set worm_attack      1"
proc cbr_traffic { fid src dst start stop } {
global ns_ node_ interval_
set udp_($fid) [new Agent/UDP]
$ns_ attach-agent $node_($src) $udp_($fid)
set cbr_($fid) [new Application/Traffic/CBR]
$cbr_($fid) set packetSize 512
$cbr_($fid) set interval_ $interval_
$cbr_($fid) attach-agent $udp_($fid)
set null_($fid) [new Agent/Null]
```

```
$ns_ attach-agent $node_($dst) $null_($fid)
$ns_ connect $udp_($fid) $null_($fid)
$ns_ at $start "$cbr_($fid) start"
$ns_ at $stop "$cbr_($fid) stop"
}
cbr_traffic 0 41 14 5 20
cbr_traffic 1 44 37 25 50
$ns_ at $val(stop).0002 "puts \"ns exiting..\"; $ns_ halt"
puts "starting simulation..."
$ns_ run
```

REFERENCES

1. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey", Computer Networks, vol. 38, pp. 393-422, 2002.

2. Ashfaq Hussain Farooqi, Farrukh Aslam Khan ,Jin Wang Sungyoung Lee (2012), “A novel intrusion detection framework for wireless sensor networks”, Pervasive Ubiquitous Computing DOI 10.1007/s00779-012-0529-y, 2012.

3. Hai TH, Khan F, Huh EN, “ Hybrid intrusion detection system for wireless sensor networks”, In Computational science and its applications, 4706th edition. Springer, Berlin, 2007.

4. Krontiris I, Dimitriou T, and Freiling F. C., "Towards intrusion detection in wireless sensor networks", In Proc. of the 13th European Wireless Conference, 2007.

5. Krontiris I, Dimitriou T, Giannetsos T, “ LIDeA: a distributed lightweight intrusion detection architecture for sensor networks”, In ACM secure communication, Istanbul, Turkey, 2008.

6. Li G, He J, Fu Y, “A group based intrusion detection scheme in wireless sensor networks”. In: The 3rd international conference on grid and pervasive computing—workshop, pp 286–291, 2008.

7. Murad A. Rassam, M.A. Maarof and Anazida Zainal,” A Survey of Intrusion Detection Schemes in Wireless Sensor Networks”, American Journal of Applied Sciences 9 (10): 1636-1652, 2012.

8. I. Onat and A. Miri, "An intrusion detection system for wireless sensor networks", in IEEE International Conference on, Wireless And Mobile Computing, Networking And Communications, (WiMob'2005), pp. 253-259 Vol. 3-253-259 Vol. 3, 2005.

9. Phuong TV, Hung LX, Cho SJ, Lee YK, Lee S, "An anomaly detection algorithm for detecting attacks in wireless sensor networks". In: Intelligent and security informatics, San Diego, pp 735–736, 2006.

10. A. P. R. d. Silva, M. H. T. Martins, B. P. S. Rocha, A. A. F. Loureiro, L. B. Ruiz, and H. C. Wong, "Decentralized intrusion detection in wireless sensor networks", Montreal, Quebec, Canada, pp. 16-23, 2005.

11. A. Stetsko, L. Folkman, and V. Matyas, "Neighbor-Based Intrusion Detection for Wireless Sensor Networks," in International Conference on Wireless and Mobile Communications Los Alamitos, CA, USA, pp. 420-425, 2010.

12. Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks",IEEE Communications Surveys & Tutorials, vol. 8, pp. 2-23, 2006.

13. Xie, M., S. Han, B. Tian and S. Parvin,, "Anomaly detection in wireless sensor networks: A survey". J. Network Comput. Appli., 34: 1302-1325. DOI: 10.1016/j.jnca.2011.03.004, 2011.