AN INTRUSION DETECTION SYSTEM FOR WIRELESS SENSOR NETWORKS

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

SANDIP LOKHANDE

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

JUNE, 2009

CANDIDATE'S DECLARATION

I hereby declare that the work being presented in the dissertation report titled **"AN INTRUSION DETECTION SYSTEM FOR WIRELESS SENSOR NETWORKS"** in partial fulfillment of the requirement for the award of the degree of **Master of Technology** in **Computer Science & Engineering**, submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee (India), is an authentic record of my own work carried out during the period from July 2008 to June 2009, under the guidance of **Dr. Manoj Misra, Professor, Department of Electronics and Computer Engineering, IIT Roorkee.**

I have not submitted the matter embodied in this dissertation for the award of any other degree or diploma.

Date: 27-06-2009

Place: Roorkee.

(SANDIP LOKHANDE)

CERTIFICATE

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

Date: 27-6-9

Place: Roorkee.

(Dr. Manoj Misra)

Professor,

Department of Electronics and Computer Engineering,

IIT, Roorkee, Roorkee - 247 667.

ACKNOWLEDGEMENTS

I would like to take this opportunity to extend my heartfelt gratitude to my guide and mentor **Dr. Manoj Misra**, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, for his trust in my work, his able guidance, regular source of encouragement and assistance throughout this dissertation work.

I also wish to thank Indian Institute of Technology Roorkee for giving me this opportunity.

Most importantly, I would like to extend my deepest appreciation to my family for their love, encouragement and moral support. Finally I thank God for being kind to me and driving me through this journey.

(SANDIP LOKHANDE)

ABSTRACT

In recent years, wireless sensor networks have many potential applications for both civil and military tasks. However, sensor networks are susceptible to many types of attacks because they are deployed in open and unprotected environment. So it is necessary to use effective mechanisms to protect sensor networks against many types of attacks. As a second line of defence, Intrusion detection is one of the major and efficient defence methods against attacks in a computer network and system. However, low memory and battery power of sensors requires that security solutions for sensor network should be designed with limited usage of computation and resources.

In this dissertation, we propose an Intrusion Detection System model which consists of two intrusion detection modules: local IDS agent and global IDS agent. Local IDS agent monitors the information sent and received by the sensor. Global IDS agent monitors the neighbouring nodes' behaviour. Local IDS agent remains active all time where as global IDS agent gets activated according to the K_m-monitor eligibility algorithm. In K_m–monitor algorithm, node decides that its global IDS agent should be activated on the basis of its one hop and two hop neighbouring nodes' global IDS agent state and monitoring degree provided by base station. Our main idea behind activating global IDS agent on some particular nodes is to reduce the energy consumption of the sensor node and hence enhance the network lifetime.

The proposed intrusion detection system has been simulated using widely known and available Network Simulator NS-2 on a Linux based platform. Its performance have been compared with the watchdog for various test cases. Results show that the proposed IDS consumes lesser energy in comparison to watchdog and thus increases the network lifetime.

CONTENTS

CANDIDATE'S DECLARATION i

ACKNOWLEDGEMENTS ii

ABSTRACT iii

CONTENTS iv

LIST OF FIGURES vi

CHAPTER 1: INTRODUCTION 1

1.1 Introduction 1

1.2 Motivation 2

1.3 Statement of the Problem 3

1.4 Organization of the Report 4

CHAPTER 2: BACKGROUND 5

2.1 Wireless Sensor Networks 5

2.1.1 Characteristics 5

2.1.2 Requirements 7

2.1.3 WSN Vulnerabilities 9

2.2 Security in WSN 9

2.2.1 Threats in WSN 9

2.2.2 Security Requirements 12

2.3 Coverage and Connectivity Configuration 14

CHAPTER 3: LITERATURE REVIEW 16

3.1 Intrusion Detection System (IDS) 16

3.2 IDS Classification 16

3.2.1 Based On Audit Data 16

3.2.2 Based On Detection Technique 17

3.2 Related Work 19

CHAPTER 4: PROPOSED IDS MODEL **23**

4.1 IDS Model 23

4.2 Detection Algorithms 27

4.3 Selection Algorithm 30

CHAPTER 5: SIMULATION **35**

5.1 NS-2 Sensor Network Extensions 35

5.2 Sensor Network Configuration 39

5.3 NS-2 Intrusion Detection System Implementation 43

CHAPTER 6: RESULTS AND DISCUSSION **45**

6.1 Testing Methodology 45

6.2 Analysis 45

CHAPTER 7: CONCLUSIONS AND FUTURE WORK **50**

7.1 Conclusions 50

7.2 Suggestions for Future Work 50

REFERENCES **51**

LIST OF FIGURES

Figure 2.1:	Wireless Sensor Networks	7
Figure 4.1:	IDS Framework	23
Figure 4.2:	Algorithm of for intrusion detection in communication nodes	28
Figure 4.3:	Local detection algorithm	29
Figure 4.4:	Global detection algorithm	29
Figure 4.5:	An example of K_m-eligibility	32
Figure 4.6:	The K_m-Monitor Eligibility Algorithm	33
Figure 5.1:	NS-2 modified (left) and added (right) classes for WSN simulation	38
Figure 6.1:	Activation of global IDS agent on sensor nodes in wireless sensor network	46
Figure 6.2:	Percentage of monitor nodes with different network density	47
Figure 6.3:	Energy consumption of monitor nodes	47
Figure 6.4:	Number of alert messages with diffrent network density	48
Figure 6.5:	Monitor Degree vs. Detection Probability	49

Chapter 1
INTRODUCTION

1.1 Introduction

Wireless Sensor Networks (WSNs) consist of small devices called sensor nodes with transceiver, processor, memory, battery and sensor hardware. One can precisely and deeply monitor the environment with widespread deployment of these devices. Sensor nodes are resource-constrained in terms of the radio range, processor speed, memory size and power. Apart from this, sensor nodes are generally stationary. The traffic rate is very low and generally the traffic is periodic as they sense and send sensed data periodically. There may be long idle periods during which sensor nodes turn off their radio to save energy consumed by idle listening. Recharging or replacing batteries is expensive and may not even be feasible in some situations. Therefore, WSN applications need to be extremely energy-aware.

WSN is mostly unguarded. Hence, capturing a node physically, altering its code and getting private information like cryptographic keys is easily possible for an attacker. Wireless medium is inherently broadcast in nature. This makes them more vulnerable to attacks. Attacks can disrupt the operation of WSN and can even defeat the purpose of their deployment. An adversary can launch Denial of Services (DoS) attacks without much effort (e.g. even without cracking keys used for cryptography-based solutions). To be practical for real-life WSN deployments, techniques for detecting attacks should be lightweight. It is important to find nodes that are posing attacks and isolate them because physical capture and subsequent loss of secret information is easily possible [1].

Security measures must be applied to protect the network from a variety of attacks. Since no intrusion prevention measures is perfect, intrusion detection becomes an important second wall of defence to protect the network. WSN has unique nature which is different from other kind of networks. It contains a large amount of tiny sensing devices which are limited in energy, computation, and communication capabilities. They are designed for specific applications (environment monitoring,

infrastructure management, public safety, medical and health care, home and office security, transportation, and military applications) and they interact closely with their physical environments. Providing adaptive new intrusion detection measures remain a challenging research problem. To provide a secure wireless sensor networks, it needs to deploy intrusion detection system (IDS) and response techniques.

Since bandwidth and power battery are limited resources in WSNs, thus an efficient way of utilizing these resources is needed in construction of IDS. In this dissertation, we propose an IDS model which detects intrusion using predefined IDS rules and uses the K_m-monitor eligible algorithm to activate the global intrusion detection modules, which are responsible for monitoring neighbouring nodes. It minimizes the monitor nodes which have to activate the intrusion detection modules, thus enhance network lifetime.

1.2 Motivation

Wireless sensor networks (WSNs) have become increasingly one of the most interesting areas over the past few years. Applications of WSNs are numerous and growing, some of them are even security critical, like military or safety applications. Wireless sensor networks are formed of sensor nodes with stringent resources in terms of battery power, processor speed, and memory and radio range. They have specific communication and traffic patterns. It is envisioned that when deployed in large scale, they can deeply monitor the surroundings.

Making sensor networks secure is especially challenging because of wireless medium, resource constraints of nodes and the fact that WSN is physically unguarded. Sensor nodes can be physically compromised which leads to the loss of a secret information. Cryptography based techniques alone are incapable of securing WSN. The code of sensor nodes can be altered to pose attacks. Even new sensors can be added just to pose attacks. Hence, intrusion detection techniques must be designed to detect attacks. Further, these techniques should be lightweight because of resource constrained nature of WSN.

Since common sensor nodes are designed to be cheap and small, they do not have enough hardware resources. Thus, the available memory may not be sufficient to create a detection log file. Moreover, a sensor node is designed to be disposed after being used by the application and it makes difficult to recover a log file due to the possible dangerous environment in which the network was deployed. The software stored in the node must be designed to save as much energy as possible in order to extend the network lifetime. Finally, another challenge to the design of IDS is the frequent failures of sensor nodes when compared to processing entities found in wired networks.

Most of the research focuses on implementing intrusion detection mechanisms in every node regardless of the node's energy. A simpler detection technique is preferable for resource limited devices. Tradeoffs should be considered between the detection effectiveness and energy efficiency of the intrusion detection system. A question that needs to be answered is whether it is possible to have fewer nodes that detect intrusion, and what impact it has on the system resources and detection effectiveness. This is the focus of this dissertation.

1.3 Statement of the Problem

The main objective of the dissertation is to design efficient IDS model for wireless sensor network which efficiently detects the intrusion in the network and satisfies energy constrain of the wireless sensor network.

In order to achieve the above objective, we divided it into smaller objectives as given below:

- Study various intrusion detection techniques and analyzing some of the proposed techniques.
- Propose a new intrusion detection system model which is energy efficient and detects the intrusions in the network.
- Evaluate the performance of the proposed approach by simulation.

1.4 Organization of the Report

The report is divided into seven chapters including this introductory chapter. Rest of the report is organized as follows:

Chapter 2 gives an overview of Wireless Sensor Networks, WSN security threats and security requirements.

Chapter 3 presents brief description about the work done in the field of intrusion detection system in wireless sensor network.

Chapter 4 describes the details of the proposed Intrusion Detection System model.

Chapter 5 discusses about the wireless sensor network simulation extension and the implementation of proposed Intrusion Detection System model.

Chapter 6 discusses simulation results obtained from simulation.

Chapter 7 concludes the dissertation and provides directions for the future work.

Chapter 2
BACKGROUND

2.1 Wireless Sensor Network

A sensor network is a collection of small distributed devices called sensor nodes, which use sensors for measurement (temperature, motion, pressure, sound) and for prediction (weather forecast, fire ignition, earthquakes, military attack, building safety). Some of the most important characteristics of wireless sensor networks are the cooperative effort of sensor nodes and their capability of self organization during the entire life cycle.

2.1.1 Characteristics

Self-organizing capabilities

A wireless sensor network (WSN) consists of a large number of sensor nodes. They are deployed over an area and form a wireless network. The position of sensor nodes need not be engineered or pre-determined. This allows random deployment in inaccessible terrains or disaster relief operations. On the other hand, this also means that sensor network protocols and algorithms must possess self-organizing capabilities.

Cooperative effort of sensor nodes

A unique feature of sensor networks is the cooperative effort of sensor nodes. Sensor nodes are fitted with an on-board processor. Instead of sending the raw data to the nodes responsible for the fusion, sensor nodes use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data.

Short-range communication and multihop routing

Since large number of sensor nodes are densely deployed and they are having short communication range. Hence, multihop communication in sensor networks is expected to consume less power than the traditional single hop communication. Furthermore, the transmission power levels can be kept low, which is highly desired

in covert operations. Multihop communication can also effectively overcome some of the signal propagation effects experienced in long-distance wireless communication.

Limitations on energy and computation power

The sensor nodes are autonomous devices with limited battery, computational power, and memory. A typical sensor node such as MICA has an 8 MHz microprocessor, 128 Kb program flash memories and 512 Kb serial flash memories. Due to size limitations AA batteries or quartz cells are used as the primary sources of power. The average sensor node will expend approximately 4.8mA receiving a message, 12mA transmitting a packet and 5μA sleeping [2]. In addition the CPU uses on average 5.5mA when in active mode. Energy dissipation for an activity is given by the formula [3]:

Energy consumption (Joule) = Power * Electric current *Time.

Dynamic Topology

Dynamic environmental conditions require the system to adapt over time to changing connectivity and system stimuli.

Operation

Figure 2.1 shows the complexity of wireless sensor networks, which generally consist of a data acquisition network and a data distribution network, monitored and controlled by a management centre. The plethora of available technologies make even the selection of components difficult, let alone the design of a consistent, reliable, robust overall system.

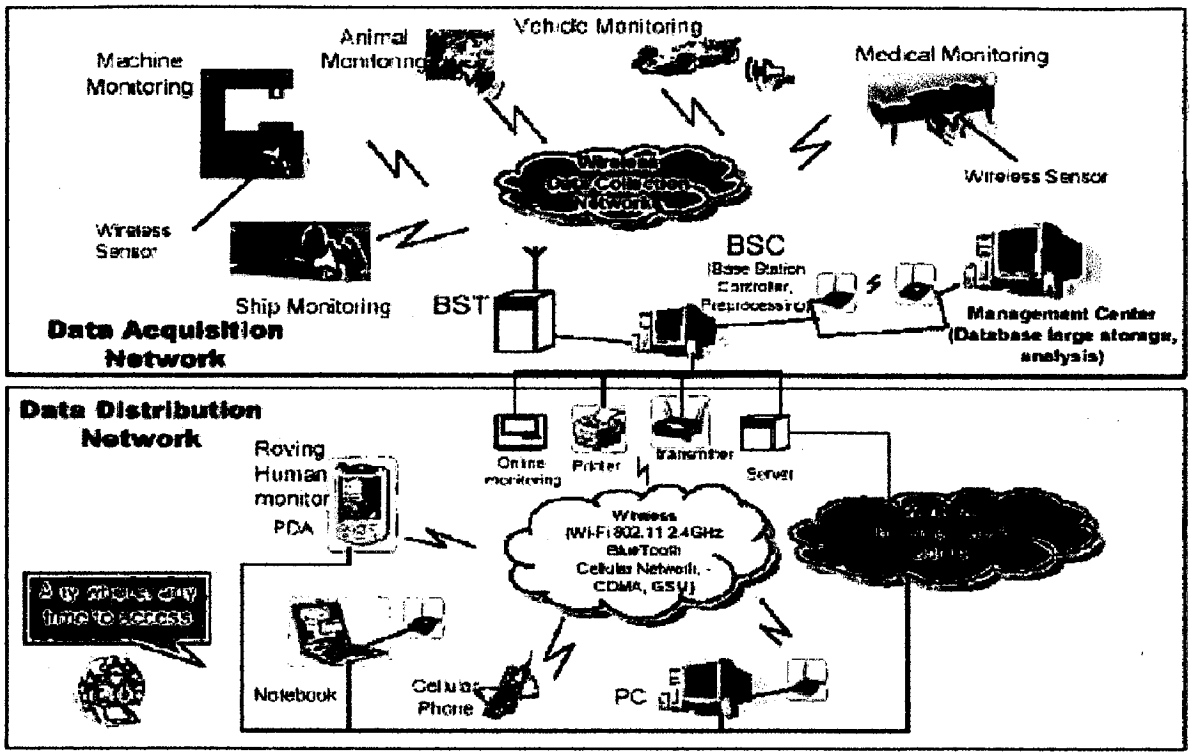

Figure 2.1 Wireless Sensor Networks

2.1.2 Requirements

Due to the characteristics and limitations of WSN, the following are the requirements in building applications on this network [4]:

Scalability

To make use of the cheap small-sized sensors, sensor networks may contain thousands of sensor nodes. Scalability and managing these huge numbers of sensors is a major issue. Clustering is one solution to this problem. In clustering, neighbouring sensors join to build one cluster (group) and elect a cluster head to manage this group.

Low energy use

In many applications, the sensor nodes will be deployed in a remote area in which case servicing a node may not be possible. Thus, the lifetime of a node may be determined by the battery life, thereby requiring minimal energy expenditure.

Efficient use of the small memory

When building sensor networks, issues such as routing-tables, data replication, security and such should be considered to fit the small size of memory in the sensor nodes.

Data aggregation

The huge number of sensing nodes may congest the network with information. To solve this problem, some sensors such as the cluster heads can aggregate the data, do some computation (e.g., average, summation, highest, etc.), and then broadcast the summarized new information.

Network self-organization

Given the large number of nodes and their potential placement in hostile locations, it is essential that the network be able to self-organize itself. Moreover, nodes may fail (either from lack of energy or from physical destruction), and new nodes may need to join the network. Therefore, the network must be able to periodically reconfigure itself so that it can continue to function. Individual nodes may become disconnected from the rest of the network, but a high degree of connectivity overall must be maintained.

Collaborative signal processing

The end goal of WSN is the detection/estimation of some event(s) of interest, and not just communication. To improve the detection performance, it is often quite useful to fuse data from multiple sensors. This data fusion requires the transmission of data and control messages. This need may put constraints on the network architecture.

Querying ability

There are two types of addressing in sensor network; data-centric, and address-centric. In data-centric, a query will be sent to specific region in the network. Whereas, in addressing-centric, the query will be sent to an individual node.

2.1.3 WSN Vulnerabilities

WSNs are more vulnerable to attacks due to the following reasons:

1. Sensor nodes are mostly physically unguarded. A capture of a single node by an attacker can result in a compromise of shared secrets or cryptographic keys.
2. Sensor nodes are more resource-constrained in terms of their radio range, processor speed, memory capacity and battery power.
3. DoS attacks can succeed more easily, since sensor nodes are resource-constrained. Thus, DoS attacks are more dangerous, more easily defying the purpose of WSN deployment, even without cracking cryptographic keys.
4. Due to specific traffic patterns, use of asymmetric cryptographic primitives incurs a heavy communication overhead. As a consequence, asymmetric cryptography—which is orders of magnitude slower than the symmetric one—is infeasible for data aggregation, considering limited resources of sensor nodes.

2.2 Security in WSN

As mentioned in the previous section, sensor networks pose unique challenges. Security techniques used in traditional networks cannot be applied directly. First, we have to make sensor networks economically viable as sensor devices are limited in their energy, computation, and communication capabilities. Second, unlike traditional, sensors are often unattended, presenting the added risk of physical attack. Third, sensor networks interact closely with their physical environments and with people, posing new security problems. Consequently, existing security mechanisms are inadequate, and new ideas are needed. The new problems inspire new research and provide an opportunity to properly address sensor network security [1].

2.2.1 Threats in WSN

Attacks could be insider attacks or outsider attacks. In an outsider attack, the attacking node is not an authorized participant of the sensor network. As the sensor network communicates over a wireless channel, a passive attacker can easily eavesdrop on the network's radio frequency range, in an attempt to steal private or sensitive

information. The adversary can also alter or spoof packets, to infringe on the authenticity of communication or inject interfering wireless signals to jam the network. Another form of outsider attack is to disable sensor nodes. An attacker can inject useless packets to drain the receiver's battery, or he can capture and physically destroy nodes.

Different from outsider attacks, insider attacks are performed by compromised nodes in the WSN. With compromised node, an adversary can perform an insider attack. In contrast to disabled node, compromised nodes activity seeks to disrupt or paralyze the network. A compromised node may be a subverted sensor node or a more powerful device, like laptop, with more computational power, memory, and powerful radio. It may be running some malicious code and seek to steal secrets from the sensor network or disrupt its normal functions. It may have a radio compatible with sensor nodes such that it can communicate with the sensor network. Next we describe some of the possible attacks on WSNs:

Routing Attacks

The simplicity of many routing protocols for WSN makes them an easy target for attacks. Karlof and Wagner in [5] classify the routing attacks into the following categories:

1. *Spoofed, altered, or replayed routing information*

While sending the data, the information in transit may be altered, spoofed, replayed, or destroyed. Since sensor nodes usually have only short range transmission, an attacker with high processing power and larger communication range could attack several sensors simultaneously and modify the transmitted information.

2. *Selective forwarding*

In this kind of attack a malicious node may refuse to forward every message it gets, acting as black hole or it can forward some messages to the wrong receiver and simply drop others.

3. *Sinkhole attacks*

In the Sinkhole attack, the goal of the attacker is to attract all the traffic. Especially, in the case of a flooding based protocol the malicious node may listen to requests for routes, and then reply to the requesting node with messages containing a bogus route with the shortest path to the requested destination.

4. *Sybil attacks*

In Sybil attack the compromised node presents itself as it as multiple nodes. This type of attack tries to degrade the usage and the efficiency of the distributed algorithms that are used. Sybil attack can be performed against distributed storage, routing, data aggregation, voting, fair resource allocation, and misbehaviour detection [6].

5. *Wormholes*

Wormhole attack is an attack in which the malicious node tunnels messages from one part of the network over a link, that doesn't exist normally, to another part of the network. The simplest form of the wormhole attack is to convince two nodes that they are neighbours. This attack would likely be used in combination with selective forwarding or eavesdropping.

6. *HELLO flood attacks*

This attack is based on the use of broadcast Hello messages by many protocols to announce themselves in the network. So an attacker with greater range of transmission may send many Hello messages to a large number of nodes in a big area of the network. These nodes are then convinced that the attacker is their neighbour. Consequently the network is left in a state of confusion.

7. *Acknowledgment Spoofing*

Some wireless sensor network routing algorithms require link layer acknowledgments. A compromised node may exploit this by spoofing these acknowledgments, thus convincing the sender that a weak link is strong or a dead sensor is alive.

Denial of Service (DoS)

This class of attacks is not concerned with the information that is transmitted. Rather, the goal of the attacker is to exhaust the resources of the networks and cause it not to function properly. Wood and Stankovic [7] classify several forms of DoS attacks based on the layer that the attack uses. At the physical layer the attacks take the form of jamming and tampering. Jamming has to do with interfering with the radio frequencies nodes are using. Tampering refers to the physical altering or even damaging of the nodes. An attacker can damage and replace a node, for example, by stealing or replacing information or cryptographic keys. At the link layer the attacker can generate collisions and exhaustion may be caused from protocols that attempt retransmission repeatedly, even when triggered by an unusual and suspicious collision.

2.2.2 Security Requirements

Authentication: Authentication is necessary to enable sensor nodes to detect maliciously injected or spoofed packets. It enables a node to verify the origin of a packet and ensure data integrity. Almost all applications require data authentication. On one hand, for military and safety-critical applications, the adversary has clear incentives to inject false data reports or malicious routing information; on the other hand, even for civilian applications such as office/home applications where we expect a relatively non-adversarial environment. Although authentication prevents outsiders from injecting or spoofing packets, it does not solve the problem of compromised nodes. Since a compromised node has the secret keys of a legitimate node, it can authenticate itself to the network. However, we may be able to use intrusion detection techniques to find the compromised nodes and revoke their cryptographic keys network-wide.

Confidentiality: Ensuring the secrecy of sensed data is important for protecting data from eavesdroppers. We can use standard encryption functions and a shared secret key between the communicating parties to achieve secrecy. However, encryption itself is not sufficient for protecting the privacy of data, as an eavesdropper can perform traffic analysis on the overheard cipher-text, and this can release sensitive information about the data. In addition to encryption, privacy of sensed data also

needs to be enforced through access control policies at the base station to prevent misuse of information. Node compromise complicates the problem of secrecy, for sensitive data may be released when a compromised node is one endpoint of the communication; or if a globally or group shared key is used, the compromised node can successfully eavesdrop and decrypt the communication between other sensor nodes within its radio frequency range.

Availability: Providing availability requires that the sensor network be functional throughout its lifetime. Denial-of-service (DoS) attacks often result in a loss of availability. In practice, loss of availability may have serious impacts. In a manufacturing monitoring application, loss of availability may cause failure to detect a potential accident and result in financial loss; in a battlefield surveillance application, loss of availability may open a back door for enemy invasion. Various attacks can compromise the availability of the sensor network. When considering availability in sensor networks, it is important to achieve graceful degradation in the presence of node compromise or node failures.

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. The data freshness objective ensures that messages are fresh, meaning that they obey a message ordering and have not been reused.

Integrity: With the implementation of confidentiality, an adversary may be unable to steal information. However, this doesn't mean the data is safe. The adversary can change the data, so as to send the sensor network into confusion. For example, a malicious node may add some fragments or manipulate the data within a packet. This new packet can then be sent to the original receiver. Data loss or damage can even occur without the presence of a malicious node due to the harsh communication environment. Thus, data integrity ensures that any received data has not been altered in transit [8].

2.3 Coverage and Connectivity Configuration

Energy is a paramount concern in wireless sensor network applications that need to operate for a long time on battery power. For example, habitat monitoring may require continuous operation for months, and monitoring civil structures (e.g., bridges) requires an operational lifetime of several years. Recent research has found that significant energy savings can be achieved by dynamic management of node duty cycles in sensor networks with high node density. In this approach, some nodes are scheduled to sleep (or enter a power saving mode) while the remaining active nodes provide continuous service. A fundamental problem is to minimize the number of nodes that remain active, while still achieving acceptable quality of service for applications. In particular, maintaining sufficient sensing coverage and network connectivity with the active nodes is a critical requirement in sensor networks [9].

Sensing coverage characterizes the monitoring quality provided by a sensor network in a designated region. Different applications require different degrees of sensing coverage. While some applications may only require that every location in a region be monitored by one node, other applications require significantly higher degrees of coverage. For example, distributed detection based on data fusion requires that every location be monitored by multiple nodes, and distributed tracking and classification requires even higher degrees of coverage. The coverage requirement for a sensor network also depends on the number of faults that must be tolerated. A network with a higher degree of coverage can maintain acceptable coverage in face of higher rates of node failures. The coverage requirement may also change after a network has been deployed, e.g., due to changes in application modes or environmental conditions. For example, a surveillance sensor network may initially maintain a low degree of coverage required for distributed detection. After an intruder is detected, however, the region in the vicinity of the intruder must reconfigure itself to achieve a higher degree of coverage required for distributed tracking.

Sensing is only one responsibility of a sensor network. To operate successfully a sensor network must also provide satisfactory connectivity so that nodes can communicate for data fusion and reporting to base stations. The connectivity of a graph is the minimum number of nodes that must be removed in order to partition the

graph into more than one connected component. The active nodes of a sensor network define a graph with links between nodes that can communicate. If this graph is K-connected, then for any possible K-1 active nodes which fail the sensor network will remain connected. Connectivity affects the robustness and achievable throughput of communication in a sensor network.

Most sensor networks must remain connected, i.e., the active nodes should not be partitioned in any configured schedule of node duty cycles. However, single connectivity is not sufficient for many sensor networks because a single failure could disconnect the network. At a minimum, redundant potential connectivity through inactive nodes can allow a sensor network to heal after a fault that reduces its connectivity, by activating more nodes. Alternatively, even transient communication disruption can be avoided by maintaining higher connectivity among active nodes. Higher connectivity may also be necessary to maintain good throughput by avoiding communication bottlenecks.

Coverage Configuration Protocol (CCP) can dynamically configure the network to provide different degrees of coverage as requested by applications. This flexibility allows the network to self-configure for a wide range of applications and environments with diverse or changing coverage requirements. CCP can provide both coverage and connectivity guarantees when the ratio of communication range and sensing range is no lower than prescribed value. Given a coverage region A and a node coverage degree K_s, the goal of an integrated coverage and connectivity configuration is maximizing the number of nodes that are scheduled to sleep under the constraints that the remaining nodes must guarantee that A is at least K_s-covered, and all active nodes are connected.

graph into more than one connected component. The active nodes of a sensor network define a graph with links between nodes that can communicate. If this graph is K-connected, then for any possible K-1 active nodes which fail the sensor network will remain connected. Connectivity affects the robustness and achievable throughput of communication in a sensor network.

Most sensor networks must remain connected, i.e., the active nodes should not be partitioned in any configured schedule of node duty cycles. However, single connectivity is not sufficient for many sensor networks because a single failure could disconnect the network. At a minimum, redundant potential connectivity through inactive nodes can allow a sensor network to heal after a fault that reduces its connectivity, by activating more nodes. Alternatively, even transient communication disruption can be avoided by maintaining higher connectivity among active nodes. Higher connectivity may also be necessary to maintain good throughput by avoiding communication bottlenecks.

Coverage Configuration Protocol (CCP) can dynamically configure the network to provide different degrees of coverage as requested by applications. This flexibility allows the network to self-configure for a wide range of applications and environments with diverse or changing coverage requirements. CCP can provide both coverage and connectivity guarantees when the ratio of communication range and sensing range is no lower than prescribed value. Given a coverage region A and a node coverage degree K_s, the goal of an integrated coverage and connectivity configuration is maximizing the number of nodes that are scheduled to sleep under the constraints that the remaining nodes must guarantee that A is at least K_s-covered, and all active nodes are connected.

Chapter 3
LITERATURE REVIEW

3.1 Intrusion Detection System (IDS)

An intrusion is defined as a set of actions that compromises confidentiality, availability, and integrity of a system. Intrusion detection is a security technology that attempts to identify those who are trying to break into and misuse a system without authorization and those who have legitimate access to the system but are abusing their privileges. The system can be a host computer, network equipment, a firewall, a router, a corporate network, or any information system being monitored by an intrusion detection system.

An IDS dynamically monitors a system and users' actions in the system to detect intrusions. Because an information system can suffer from various kinds of security vulnerabilities, it is both technically difficult and economically costly to build and maintain a system that is not susceptible to attacks. An IDS, by analyzing the system and users' operations, in search of undesirable and suspicious activities, may effectively monitor and protect against threats.

3.2 IDS Classification

There are different types of IDS available. One classification is based on audit data and it is divided into two categories: Network Based IDS and Host Based IDS. Other is based on detection techniques and it is also divided into three categories: Signature Based IDS, Anomaly Based IDS and Specification Based IDS.

3.2.1 Based On Audit Data

Network Based Intrusion Detection System:

Network Based Intrusion Detection Systems (NIDS) monitors and examines traffic or packets to and from sensor nodes in the network to detect intrusion patterns or misbehaviour of nodes. Network based IDS has some advantages, it usually provides reliable, real-time information without consuming network or host resources. As this

type of IDS monitors an attack in real time, it can respond to an attack in progress and thus limit damage. It can also detect DoS attacks. Packets are identified based on some signatures. Three primary types of signature identified from [10] are,

- String signatures – based on text string which indicates a possible attack.
- Port signatures – always monitors well-known and frequently attacked ports.
- Header condition signatures – check for dangerous or illogical combinations in packet headers.

Host Based Intrusion Detection System:

Host Based Intrusion Detection Systems (HIDS) are placed on (inside) individual hosts or nodes in the network and detect malicious activity on that host only. Basically they use small program known as IDS agent which resides within the system. They monitor the operating system to detect illegal activity, writing to log files, and generate alarms to base station or administrator of the system. They cannot monitor entire network segment. Host-based IDS can analyze the system, event logs and sometimes user activity to detect an attack on the host and to decide whether the attack was performed [10].

Moreover, host-based IDSs have some characteristics:

- They monitor file integrity of system files, access rights and privileges of user.
- They are efficient to detect trusted insider attacks.
- They are efficient to detect attacks from the outside.
- They can be organized to analyze network packets, connection attempts.

3.2.2 Based On Detection Technique

Signature Based Intrusion Detection System:

A Signature Based Intrusion Detection System gathers information or data by monitoring audit logs or network packets and compares them against a database of known patterns (also called signatures) or attributes from known malicious threats and if there is a match then, a response is initiated. The main advantage of this type of IDSs is low false alarm rates. The disadvantage with signature-based IDSs is that only

attack or threat signatures that are stored in their database are detected. As the number of patterns could range up to thousands, pattern matching consumes not only the storage but also the most of the CPU cycles to execute the pattern matching algorithms. Again there is a lag between a new threat being discovered and the signature for detecting that threat being applied to the IDS and during that lag time the IDS would be unable to detect the new threat. Additionally, this type of IDS is resource-demanding. The signature database continually needs maintenance and updating with new vulnerabilities and threats.

Anomaly Based Intrusion Detection System:

An Anomaly Based Intrusion Detection System creates normal profiles of system states or user behaviours and compares them with current activities. If a significant deviation is observed, the IDS raises an alarm. Some advantages of an Anomaly based IDS are as follows:

- The system can dynamically adapt with new, unique, or original vulnerabilities and threats. It can detect unknown attacks.
- The information it produces that can be used to define signatures for signature based IDS.

Some disadvantages of an Anomaly based IDS are that it cannot detect an attack that does not significantly change the system-operating characteristics. Moreover, Anomaly based IDSs experience following:

- High false alarm rates, which can create data noise that, can make the system unusable or difficult to use.
- Anomaly detection often requires extensive training of system event records for characterizing normal behaviour patterns.
- The host or network may experience an attack at the same time the intrusion detection system is learning the behaviour.

Specification Based Intrusion Detection System:

Specification-based detection techniques combine the advantages of misuse detection and anomaly detection by using manually developed specifications to characterize legitimate system behaviours. Specification-based detection approaches are similar to anomaly detection techniques in that both of them detect attacks as deviations from a normal profile. However, specification-based detection approaches are based on manually developed specifications, thus avoiding the high rate of false alarms. However, the downside is that the development of detailed specifications can be time-consuming [11].

Wireless sensor networks are susceptible to many forms of intrusion. In wired networks, traffic and computation are typically monitored and analyzed for anomalies at various concentration points. This is often expensive in terms of network's memory and energy consumption, as well as bandwidth. Wireless sensor networks require a solution that is fully distributed and inexpensive in terms of communication, energy, and memory requirements. In order to look for anomalies, applications and typical threat models must be understood. It is particularly important for researchers to understand how cooperating adversaries might attack the system. The use of secure groups may be a promising approach for decentralized intrusion detection.

3.3 Related Work

Wireless ad hoc networks and wireless sensor network share some common characteristics and there have been some development of IDS in wireless ad hoc network. But R. Rodrigo et al. [12] has shown in his paper that they can't be directly applied in WSNs. They proposed a novel technique for optimal monitoring of neighbours called spontaneous watchdogs from which many approaches extend. I. Onat et al. [13] proposed an anomaly detection based security scheme for large scale sensor networks. In their method, each sensor node can build a simple statistical model of its neighbour's behaviour and these statistics can be used to detect changes. The system features they choose for analyzed anomalies are average of receive power and average of packet arrival rate. Their system cannot detect selective forwarding and wormhole attacks because of their simple statistical features. S. Banerjee et al. [14] proposed an intrusion detection mechanism based on ant colonies system. Their

basic idea is to identify the affected path of intrusion in the sensor network by investigating the pheromone concentration. However, they do not specify the detail solution on the routing attacks. P. Techateerawat et al. [15] proposed an intrusion framework based on the layout and selection of monitor nodes. They proposed a voting algorithm for selection of nodes which have to activate their IDS agent. Their approach reduced the number of monitor nodes thus saving energy consume in the networks but it also reduces the possibility of detection.

A. P. Silva et al. [3] proposed decentralized IDS that are based on the inference of the network behaviour obtained from the analysis of events detected by a monitor node, i.e., the node that implements the IDS system. However, the authors do not take into account how the monitor nodes are operated and the placement of monitor nodes in their scheme. Du et al. [16] proposed anomaly detection by using a range free localization scheme. In the literature, detection mechanism that can identify compromised nodes in WSNs has been developed and analyzed. Zhang et al. [17] provided a sample to identify compromised nodes in an application where the specific beacon nodes that have their location are responsible for providing location reference to other sensors; there are two phases in this algorithm. In first phase, it computes the compromised core including some contingent compromised nodes. The second phase uses maximum matching to further eliminate compromised nodes and identifies the approximate compromised nodes. Cheng et al. [18] propose an application-independent detection model, distributed cross-layer detection model (DCD), making use of a distributed mechanism and the information of each layer in the communication protocol to detect which sensors were already compromised.

Huang et al. [19] proposed a mechanism that needs separate monitoring nodes, specifically one monitor per 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).

In [20], the authors proposed a lightweight countermeasure for the wormhole by using local monitoring mechanism to detect wormhole attacks. Each node builds a list of two hops neighbours. A node monitors the packets going in and out of its range called guard node. The guard node watches its neighbour to know if they try to make a tunnel in the network. The disadvantage of a method it consumes much computation just only for detecting wormhole attacks. The study does not consider the situation of high network density that makes the sensor nodes overload in computation and memory usage. In [21], authors proposed a hybrid, lightweight intrusion detection system integrated for sensor networks. Their proposed intrusion detection scheme take advantage of cluster-based protocol to build a hierarchical network and provide an intrusion framework based both on anomaly and misuse techniques.

Qinghua Wang et al. [22] proposed traffic profile based intrusion detection for sensor networks. Each sensor node builds its normal traffic profile in learning phase. If there is significant deviation in incoming traffic from its normal traffic profile, it is considered as an intrusion. Guangcheng Huo et al. [23] proposed a Dynamic model of IDS (DIDS) for WSN in which IDS detection tasks will be shared by various nodes equally. After deploying sensor nodes, clusters are formed according clustering formation algorithm and activate IDS in certain nodes in clusters to detect intruders. Then if one of the IDS nodes has consumed 30 percent of the overall energy which it has before activating its IDS, clusters reconfigure and IDS will be activated in new nodes and in new clusters. This procedure goes to iteration.

Rung-Ching Chen et al. [24] proposed intrusion detection in wireless sensor networks based on isolation table. The IDS compares sensor node behaviours with attack behaviours to determine anomaly information. If the node is anomaly, it will be isolated and recorded in the isolation table. The secondary cluster head (SCH) sends isolation table to primary cluster head (PCH) to integrate isolation table. If there has no anomaly, the SCH periodically sends information to avoid nodes being intruded. Finally PCH updates isolation table to base station (BS) periodically. When the PCH is changed, the new PCH can receive the isolation table from BS for keeping isolation anomaly nodes.

Though many detection techniques have been proposed but they rarely mentioned about the placement or activation of intrusion detection modules in sensor nodes. In *watchdog* [12], every node should participate in intrusion detection. Since every node is monitoring intrusion, lifetime of the network is possible reduced quickly if the workload is concentrated on the intrusion detection modules. So, an energy efficient scheme to select the intrusion detection modules is needed. The contribution of this dissertation is to combine and improve the previous works for the creation of energy efficient intrusion detection system for sensor networks.

Chapter 4
PROPOSED IDS MODEL

4.1 IDS Model

An Intrusion Detection System (IDS) detects a security violation on a system by monitoring and analyzing network activities, and sounds an alarm when an intrusion occurs [15]. There are two kinds of approaches: misuse detection and anomaly detection. Misuse detection identifies an unauthorized use from signatures while anomaly detection identifies intrusion by analysis of an event. When either of the two techniques detects violation, it will raise an alarm to warn the system which will responds to it. IDS framework normally consists of misuse detection and anomaly detection as shown in Figure 4.1.

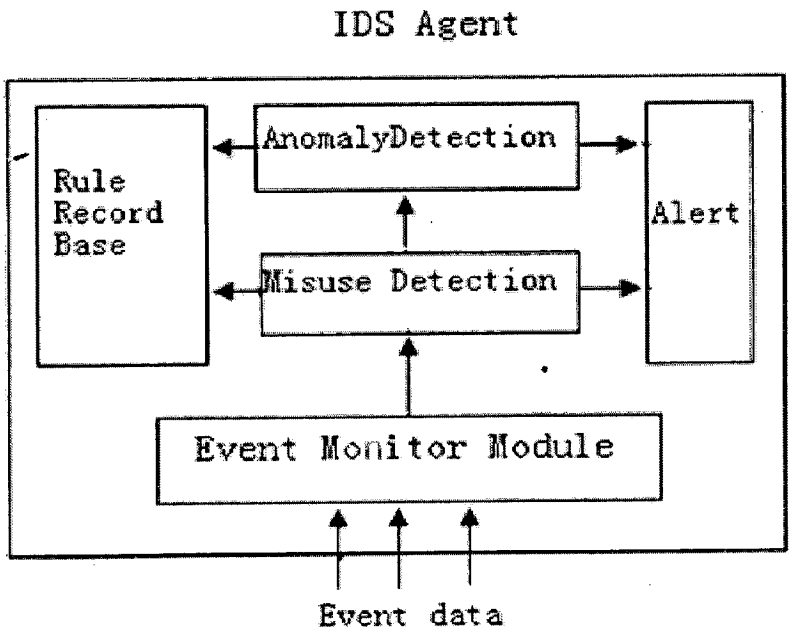

Figure 4.1 IDS Framework

Event Monitor Module is to monitor communication activities or event data in its radio range.

Rules Record Base is a data base which stores rules for judging unauthorized and high risk activities.

Misuse Detection is to analyze event data from Rules Record Base. In case of event data is matched with any rules alert signal will be raised. Otherwise, event data is forwarded to *anomaly detection* for further analysis.

Anomaly Detection is to compare event data with Rules Record Base to find harmful attacks from intruder. If probability reaches the risk threshold, alert signal will be raised.

Alert will produce an alert signal if IDS has detected an intruder according to rules. Then other system components will take some corresponding protection measures.

We assume that sensor nodes in wireless sensor network are static and initially have same energy. Preliminary work of applying IDS for sensor networks was undertaken by R. Roman et al. [12]. The authors have presented some general guidelines for applying IDS to WSNs which our work is influenced on. In our scheme, IDS is located in every sensor nodes. Each sensor node has two intrusion detection modules called local IDS agent and global IDS agent. Because of limited battery life and resources, each agent is only active when needed [21].

Local agent: Local agent module is responsible to monitor the information sent and received by the sensor. The node stores malicious node database on specific malicious nodes attacks in network. When the network is first organized, the sensor nodes don't have any knowledge about malicious nodes. After the deployment of WSNs, the malicious node database is constructed gradually. The entry in the malicious node database is created and propagated to every node by base station.

Global agent: Global agent is responsible to monitor the communication of its neighbour nodes. A node which monitors the communication of its neighbour nodes is known as monitor node. Because the broadcast nature of wireless network, every node can receive all the packets going through its radio range. Global agent must have the information of its neighbour nodes for monitoring the packets. We use local monitoring mechanism and pre-defined rules to monitor the packets [3, 20]. If the monitor nodes discover a possible breach of security in their neighbour nodes, they create and send an alert to the sink node or base station. Both agents are built on application layer.

Data structure: As mentioned above, each sensor node stores two databases: malicious nodes and neighbour nodes.

Neighbour node database: The sensor nodes build the neighbour list for monitoring and routing packets through the simple discovery mechanism. In discovery mechanism, nodes broadcast HELLO message to announce themselves to their neighbours, and a node receiving such a message may assume that it is within radio range of the sender. Each node sends its direct (one hop) neighbours list to its neighbours. After discovery process, each sensor node in network has a list of its direct neighbours and two hops neighbours [20]. Sensor node stores the neighbour list for two purposes, monitoring the packet circulating in its range and for routing packets.

Malicious node database: This malicious node database is computed and generated in the base station through the use of anomaly detection in global agent. Every node in network can be monitored by its neighbour node. The monitor node uses its neighbour list and some predefined rule to detect anomaly in data transmission. Once global agent discovers an anomaly event inside its neighbourhood, it creates and sends an alert to its base station node. The base station node will create and propagate rule for identification of malicious node to every sensor node in network. The sensor nodes update the new rule and add the entry to its malicious node database. The malicious node will be isolated from the network and not involve in the communication in the network.

Pre-defined rules: Our system follows the network-based approach; intrusions are detected by monitoring the messages exchanged by the sensor nodes. When the sensor node is first deployed, there is no entry in its internal malicious node database except for some predefined, simple rules in global agent. Global agent uses pre-defined rules and neighbour's list to watch over the communication in their neighbourhood. These rules help monitor nodes detect some common misbehaviours and specific attacks on sensor nodes, based on the previous work by A. P. Silva et al. [3].

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 neighbours.

Retransmission rule: the monitor listens to a message, pertaining to one of its neighbours 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 neighbour must occur before a defined timeout. Otherwise, an attack will be detected.

Repetition rule: the same message can be retransmitted by the same neighbour 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 neighbours. 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.

These rules are stored in rule record database and parameters of the rules are computed by the base station from information received from sensor node during learning phase. Based on the above rules, our IDS model can detect an intrusion. This

set meets the demands and restrictions of wireless sensor networks and ensures that adversarial messages are properly identified.

4.2 Detection Algorithms

We assume that when the sensor node is first deployed in the environment field, it is required the attacker an amount of time to deploy an attack. It means that no malicious node is appeared in the beginning of sensor nodes deployment. The monitor nodes use local monitoring mechanism and the predefined rules to detect anomalies in global transmission. Due to broadcast nature of wireless networks, monitor nodes will receive the packets in its radio range. These packets are caught and stored in an intrusion buffer, the information including the packet identification and type, packet's source and destination, the packet's immediate sender and receiver. The example of a packet caught by two monitor nodes in a link from node X to node Y includes {packet id, type, source node (S), destination node (D), intermediate sender (X), intermediate receiver(Y), data}. Each entry in the buffer is time stamped, which will be expired within a timeout or after the entry in the buffer have been examined by monitor nodes. When a sensor node receives a packet from a sensor node in the networks, if the intermediate sender node is in its malicious node database, which means the packet comes from a malicious node, and then the sensor node drops the packet. Otherwise, it applies a set of rule and analyses it. If a message violates one of these rules, threshold for that node is incremented. If the number of threshold for a specific node is above a given threshold (i.e., a network parameter set by the system implementer), the node is treated as an intruder and is restrained from the network [21].

The procedure of intrusion detection in communication nodes can be illustrated by the algorithm in Figure 4.2. When a sensor node receives a packet from the sender node in malicious node database, then the sensor node drops that packet. Local agent of each sensor node remains active all times. The local detection algorithm at sensor node can be illustrated as the algorithm in Figure 4.3. Local agent checks that intermediate sender of packet is in malicious node database. If packet comes from malicious node, sensor node drops that packet. If the intermediate sender of packet is not in one hop neighbour node database, it means that packet is sent by malicious node and it increment the malicious count value for that sender. CheckThreshold

method compares malicious count value with specified threshold value for generation of alert message.

Global agent will be activated according to selection algorithm described in next section. Global agent in each sensor node is illustrated as the algorithm in Figure 4.4. In global agent, each entry in the buffer is evaluated according to a sequence of rules specific to each message type. If a 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. We have adopted this strategy due to the fact that WSNs have severe resource restrictions. This strategy makes sense since the first failure already gives us an indication of an abnormal behaviour in the network. This strategy also reduces the detection latency.

```
Communication Node_i
1. Repeat <listen to the packet in its radio range>
2. Check <packet_header>
3. If (ID_i = =destination_node's ID) {
4. If Local_detection(packet)  then drop(packet)
5. Else receive(packet);
6.}
7. And If (ID_i = = intermediate_sender_node's ID) {
8. If Local_detection(packet) then drop(packet)
9. Else forward(packet);
10. }
11.If(IDS_STATE = =ACTIVE)
12. then Global_detection (packet)
13. Else Drop(packet)
14. Until No transmission
```

Figure 4.2 Algorithm of for intrusion detection in communication nodes

```
Procedure Local_detection(packet_i)
1. {
2. If Looking(intermediate_sender_node's ID, malicious node's database)
3. then return TRUE;
4. If (NOT Looking(intermediate_sender_node's ID, 1 hop neighbour) {
5. Increment Malicious_count_i ;
6. CheckThreshold(Malicious_count_i);
7.  return TRUE;
8. }
9. Else receive(packet_i)
10. }
```

Figure 4.3 Local detection algorithm

```
Procedure Global_detection(packet_i)
1. {
2. If Looking(packet_i_identification, intrusion buffer)
3. then {
4. If Check(intermediate_receive_node's ID, 2 hop
5. neighbor's list) Or Check(packet_i, predefined rules)
6. then {
7.  Increment Malicious_count_i;
8.  CheckThreshold(Malicious_count_i);
9. }
10. Else Store(packet_i, intrusion_buffer)
11. }
```

Figure 4.4 Global detection algorithm

Detection of Selective forwarding: In selective forwarding attacks, the transmission link from node A to node B is monitored by their monitor nodes. These nodes catch and store the packets going out of node A with node B as their next intermediate

node. If node B tries to stop or drop these packets, the monitor nodes will and send an alert to base station. The monitor nodes can also use the predefined rules to check if node B forwards the packet in the right path. If node B tries to send the packets to wrong path by forwarding to an unknown node, the monitor nodes will check their two hops neighbour node's list. If the destination node's identification of the forwarded packet is not in node B's neighbour list, the monitor nodes will send an alert to base station. After the packets are forwarded to right path, the entry in the monitor node's intrusion buffer is removed.

Detection of Sinkhole and Hello flood: The common feature between the two attacks is that the malicious node will convince it as the nearest path to base station by using high power transmission. All packets came to node A must be originated from A's neighbour list, the monitor nodes use neighbour's list and predefined signal rule to check if a packet is originated from a far located node.

Detection of Wormhole: Our system can wormhole attacks by inherit the advantage of local IDS agent monitoring mechanism. We use two hops neighbour's list and predefined rules to improve the detection of wormhole in WSNs.

Detection of Sybil: By using one hop and two hops neighbours list and predefined rules, Sybil attack is detected by intrusion detection system. IDS checks for advertised node in one hop neighbours list. If there is no match, it is detected as in intrusion.

Detection of DoS: Our system can detect DoS attack by interval, repetition rules. Interval rule checks for inter arrival packets time, if there is significant deviation in it, it detects as an attack. If same packets are transmitted for a large number of times by a single node, it will be detected by repetition rule.

4.3 Selection Algorithm

As mentioned in previous section, the monitor nodes observe the behaviour of packets that pass through them to destination. To minimize the number of monitor nodes activating the intrusion detection global agent, our proposed scheme make use of K_m-monitor eligibility algorithm which is based on coverage and connectivity protocol [9]

to select the nodes which monitor its neighbours. Our main idea is to choose the set of nodes which cooperatively monitor all the nodes in the networks. Our proposed scheme is based only on the neighbour node information built on each node to find these nodes. We assume that any two nodes u and v can directly communicate with each other if their Euclidian distance is less than a communication range R_c, i.e., $|uv| < R_c$. We also make the assumption that the adversary cannot successfully compromise a node during the short deployment phase.

K_m-Monitor Eligibility Algorithm

Each node executes an eligibility algorithm to determine whether it is necessary to become its IDS global agent active. Monitor degree of sensor node is the minimum number of monitor nodes that monitor the node. Given a requested monitor degree K_m, a node v is ineligible if every node within its coverage range is already K_m-monitor (monitor by at least K neighbours) by other active nodes in its neighbourhood. For example, assume the nodes covering the shaded circles in Figure 4.5 are active, the node 1 with the bold sensing circle is ineligible for $K_m=1$ because sensor node 7 is already monitored by sensor node 6, but eligible for $K_m > 1$. Before presenting the eligibility algorithm, we define the following notation.

- The monitor region of node v is the region inside its communication circle C(v), i.e., a point p is in v's monitor region if and only if $|pv| \leq R_c$ (communication range).
- A node p is called an intersection node between nodes u and v, i.e., $p \in C(u) \Lambda C(v)$, if p is in intersection area of the communication circles of node u and v.
- A node p is monitored by the node v if $|pv| \leq R_c$.

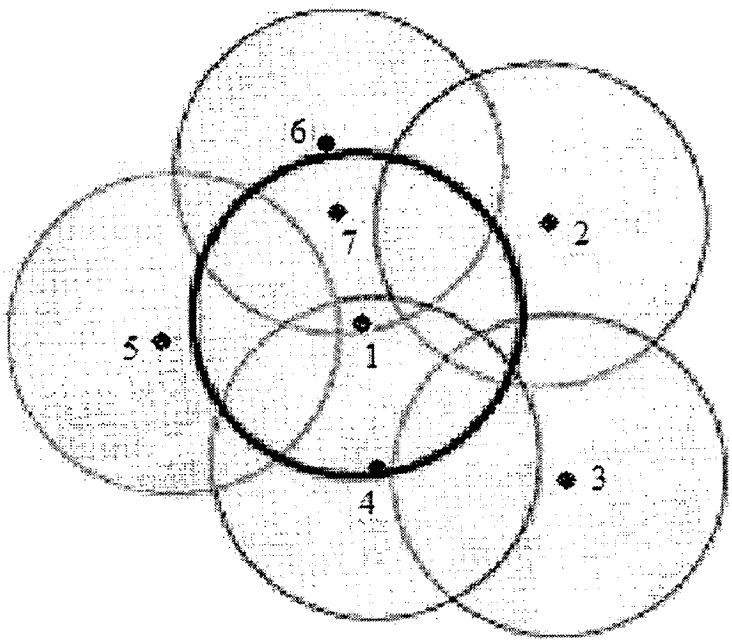

Figure 4.5 An example of K_m-eligibility

A sensor node is ineligible for turning global IDS agent active if all the nodes inside its communication circle are at least K_m-monitored. To find all the intersection nodes inside its coverage circle, a sensor node v needs to consider all the sensor nodes in its neighbour set (one hop and two hops). Common direct neighbours between a node and its neighbour node are the intersection nodes between them. AN(v) includes all the active nodes that are one hop neighbours and/or two hop neighbours to v, i.e., AN(v) = {active node u | $|uv| \leq 2R_c$ and $u \neq v$}. We define the circle C(v) of node v as the boundary of v's monitor region and NN(v) contains all one hop and two hop neighbours. If there is no intersection nodes inside the coverage circle of sensor node v, v is ineligible when there are K_m or more sensor nodes that are located at sensor v's position. The resulting monitor eligibility algorithm is shown in Figure 4.6. The eligibility algorithm requires the information about locations of all neighbours. It maintains a table of known neighbours based on the beacons (HELLO messages) that it receives from its neighbours.

```
int is_eligible (integer Km)
1. begin
2. find all intersection nodes IN inside C(v) with 1 hop and 2 hop neighbours:
3. IN = {p | (p ϵ (C(u) Λ C(v)) ) AND  u ϵ NN(v) AND |pv| ≤ Rc};
4. Find all coinciding sensors SC:
5. SC = {u | |uv|=0};
6. if (|IN|=0) {
7. if(|SC| ≥Km) return INELIGIBLE;
8. else return ELIGIBLE;
9. }
10. for (each node p ϵ IN)
11. begin
12. md(p)=|{u | u ϵ AN(v) AND |pu| ≤ Rc}|;
13. if (md(p) < Km) return ELIGIBLE;
14. end
15. return INELIGIBLE;
16. end
```

Figure 4.6 The K_m-Monitor Eligibility Algorithm

Each node determines its eligibility using the K_m-monitor eligibility algorithm based on the information about its neighbours, and may switch state dynamically when its eligibility changes. An IDS of node can be in one of three states: LEARNING, ACTIVE, and DEACTIVE. In the ACTIVE state, the node actively monitors its neighbour node and detects an intrusion, if any. When a network is deployed, all nodes are initially in the LEARNING state.

In LEARNING: Each node builds the one hop neighbour and two hop neighbour table based on HELLO beacon packets. Base station initiates first K_m-monitor eligibility algorithm execution and provides the monitor degree.

In DEACTIVE: When a beacon (HELLO, WITHDRAW, or JOIN message) is received, a node evaluates its eligibility. If it is eligible, it starts a join timer T_j, otherwise it returns to the DEACTIVE state. If it becomes ineligible after the join

timer is started (e.g., due to the JOIN beacon from a neighbour), it cancels the join timer. If the join timer expires, the node broadcasts a JOIN beacon and enters the ACTIVE state. When other nodes receive JOIN message, they updates their active node list AN.

In ACTIVE: When a node receives a HELLO message, it updates its neighbour table and executes the monitor eligibility algorithm to determine its eligibility to remain active. If it is ineligible, it starts a withdraw timer T_w. If it becomes eligible (due to the reception of a WITHDRAW or HELLO message from a communication neighbour) before the withdraw timer expires, it cancels the withdraw timer. If T_w expires, it broadcasts a WITHDRAW message and enters the DEACTIVE node. If one of the IDS nodes has consumed 30 percent of the overall energy which it has before activating its global IDS agent, and it broadcasts a WITHDRAW message. If node receives a WITHDRAW or HELLO message, it updates its active node list AN.

Both the join and withdraw timers are randomized to avoid collisions among multiple nodes that decide to join or withdraw. The values of T_j and T_w affect the responsiveness of selection algorithm. Shorter timers lead to quicker response to variations in states. Both timers are also related to the density of nodes in the network.

Chapter 5
SIMULATION

5.1 NS-2 Sensor Network Extensions

The only fundamental aspect of sensor networks missing in NS-2 was the notion of a phenomenon such as chemical clouds or moving vehicles that could trigger nearby sensors through a channel such as air quality or ground vibrations. Once a sensor detects the "ping" of a phenomenon in that channel, the sensor acts according to the sensor application defined by the NS-2 user. This application defines how a sensor will react once it detects its target phenomenon. For example, a sensor may periodically send a report to some data collection point as long as it continues to detect the phenomenon, or it may do something more sophisticated, such as collaborate with neighbouring sensor nodes to more accurately characterize the phenomenon before alerting any outside observer of a supposed occurrence. For each sensor network there is a unique sensor application to accomplish phenomena detection, such as surveillance, environmental monitoring, etc. With NS-2 [27], authors have provided the facility to invoke sensor applications by phenomena. With these sensor applications, we can study how the underlying network infrastructure performs under various constraints.

The presence of phenomena in NS-2 is modelled with broadcast packets transmitted through a designated channel. The range of phenomena is the set of nodes that can receive the PHENOM broadcast packets in that channel1. This pattern will follow whichever radio propagation model (free space, two ray ground, or shadowing) included with the phenomenon node's configuration. These propagation models roughly cover a circle, but other shapes could be achieved by varying the range of PHENOM broadcast packets and creatively moving a set of phenomenon nodes emanating the same type of phenomenon.

Our sensor network simulations have phenomenon nodes that trigger sensor nodes, but the traffic sensor nodes generate once they detect phenomena depends on the function of the sensor network. For example, sensor networks designed for energy

efficient target tracking would generate more sensor-to-sensor traffic than a sensor network designed to provide an outside observer with raw sensor data. This function is defined by the sensor application which is intended to be customized according to the traffic properties associated with the sensor network being simulated. The objects and functions we have just described are implemented in the following files:

phenom/phenom.cc, phenom.h: This file implements the PHENOM routing protocol used for emanating phenomena. It includes parameters for the pulse rate and the phenomenon type (Carbon Monoxide, heavy seismic activity, light seismic activity, sound, or generic). These types are just names that can be used to identify multiple sources of phenomena in trace files. The pulse rate is the only parameter that actually controls how a phenomenon emanates.

sensornets-NRL/sensoragent.cc, sensoragent.h: The ns manual [25] describes *agents* as "endpoints where network-layer packets are constructed or consumed". Sensor nodes use a *sensor agent* attached to the phenomenon channel for consuming PHENOM packets, and a UDP or TCP agent attached to the wireless network channel for constructing packets sent down from the sensor application. Sensor agents act as a conduit through which PHENOM packets are received and processed by sensor applications. The sensor agent does not actually look at the contents of the PHENOM packet, it simply marks the packet as received and passes it to the sensor application. This agent is implemented in sensoragent.cc.

sensornets-NRL/sensorapp.cc, sensorapp.h: The sensor application defined in this file utilizes node colour and generates sensor reports to show when the corresponding sensor node detects phenomenon3. Specifically, when the node is receiving PHENOM packets, this application changes the node colour to red, activates an "alarm" public variable, and sends a sensor report of MESG SIZE bytes to the sink node of a UDP (or TCP) connection once per TRANSMIT FREQ seconds. When the node has not received a PHENOM packet in the timeout period specified by SILENT PHENOMENON, then the node colour changes back to green. If node colour is desired to illustrate energy levels instead of sensor alarm status, then that aspect of the application can be disabled with DISABLE COLOURS.

sensornets-NRL/phenom_packet.h: This file defines the structure of PHENOM packets. The five phenomenon types defined here (CO, HEAVY GEO, LIGHT GEO, SOUND, and TEST PHENOMENON) correspond to Carbon Monoxide, heavy seismic activity, light seismic activity, audible sound, and some generic phenomenon. These types are most useful for simulations involving multiple phenomenon nodes, in order to easily distinguish who a given sensor node is detecting by looking at the NS-2 trace file.

Figure 5.1 shows where extensions are arranged within the NS-2 framework. The major additions and modifications are explained below.

trace/cmu-trace.cc, cmu-trace.h: The CMUTrace class is used to print important parts of a packet to the simulation's trace file. Since we introduced a new packet type for phenomena, we had to describe the corresponding packet format in this class. tcl/lib/ns-lib.tcl: This component of the infrastructure interprets node configurations specified in the NS-2 simulation script. Our extensions introduced two new node types, the sensor node and the phenomenon node. Therefore, we added some arguments in the node-config function to accommodate them.

tcl/lib/ns-mobilenode.tcl: In NS-2's virtual world, we are using its existing capacity for multichannel wireless networking as a means to emanate phenomena of various kinds. By using a dedicated channel for phenomena, we can simulate the unique physical medium that they occupy in the real world. Sensor nodes will need to have two interfaces, one to the 802.11 channel and one to the PHENOM channel. We implemented this kind of "multihomed" capability in ns-mobilnode.tcl.

common/packet.h: Each packet in NS-2 is associated with a unique type that associates it with the protocol that it belongs to, such as TCP, ARP, AODV, FTP, etc. Since we created a new protocol for emanating phenomena, we defined its corresponding packet type in the packet.h header file.

mac/wireless-phy.cc: NS-2 contains an energy model for wireless nodes that can be used to investigate the benefits of various energy conservation techniques, such as node sleeping or utilizing optimal network densities. The model includes attributes for specifying the power requirements of transmitting packets, receiving packets, or idly

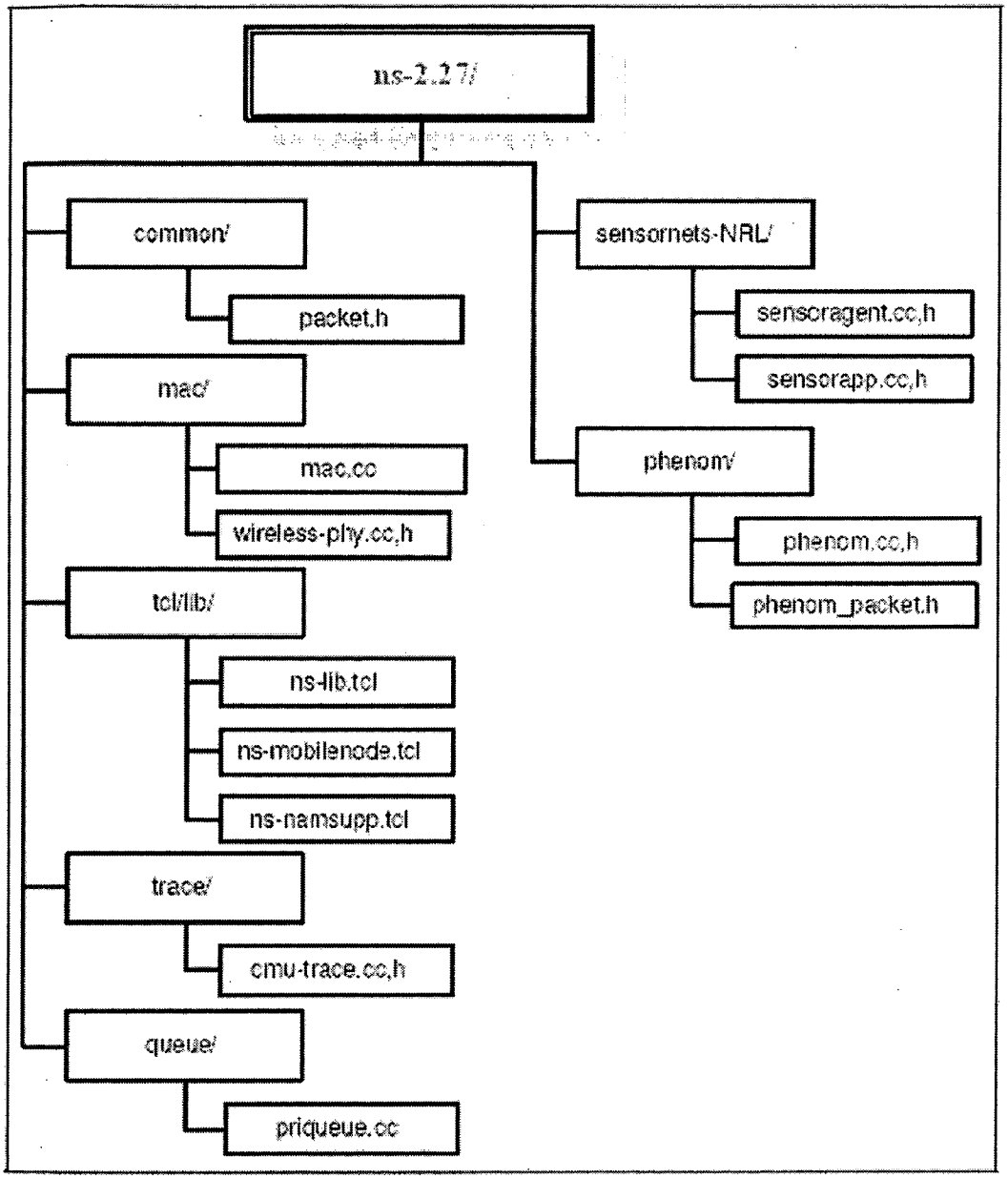

Figure 5.1 NS-2 modified (left) and added (right) classes for WSN simulation.

standing by during times of network inactivity. Sensing phenomena is a process that may consume power at another rate, so it is important to consider this where sensor network simulations are concerned. In mac/wireless-phy.cc, we have included the capability of specifying the amount of power consumed by nodes while sensing phenomena. Other small modifications were made to mac/mac.cc, tcl/lib/ns-namsupp.tcl, and queue/priqueue.cc in order to facilitate the second interface to the phenomenon channel on sensor nodes, to fix a bug in NS-2's node colouring

procedure, and to include the new PHENOM packet type into the NS-2 framework, respectively.

5.2 Sensor Network Configuration

This section describes how to code a sensor network simulation into the NS simulation script. Setting up a sensor network in NS-2 follows the same format as mobile node simulations. Places where a sensor network simulation differs from a traditional mobile node simulation are listed below. Setting up ns, god, tracing, topography objects and starting and stopping the simulation are all the same as in traditional mobile node simulations.

1) *Configure a phenomenon channel and data channel.*

Phenomenon nodes should emanate in a different channel than sensor nodes in order to avoid contention at the physical layer. All phenomenon nodes should be configured on the same channel, even if they are emanating different types of phenomena.

```
set chan_1 [new $val(chan)]
set chan_2 [new $val(chan)]
```

2) *Configure a MAC protocol for the phenomenon channel.*

Choose a MAC layer to use for emanating phenomena over the phenomenon channel. Using 802.11 is not appropriate, since phenomena should be emanating without regard to collisions or congestion control. We suggest using the basic "Mac" class instead, shown as follows:

```
set val(mac) Mac/802_11
set val(PHENOMmac) Mac
```

3) *Configure phenomenon nodes with the PHENOM "routing" protocol.*

Use node-config, just like with mobile nodes, but specify PHENOM as the routing protocol so the phenomenon is emanated according to the methods defined in phenom/phenom.cc. Also, be sure to configure in the channel and MAC layer previously specified for phenomena broadcasts. A sample node configuration statement is shown below.

```
$ns node-config \
-adhocRouting PHENOM \
 -channel $chan_1 \
-llType LL \
-macType $val(PHENOMmac) \
-ifqType Queue/DropTail/PriQueue \
-ifqLen 50 \
-antType Antenna/OmniAntenna \
-phyType Phy/WirelessPhy \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace ON \
 -movementTrace ON \
-propType Propagation/TwoRayGround
```

4) *Configure the Phenomenon node's pulse rate and type.*

The two parameters that can be used to customize phenomena are listed below. They are both optional.

a) pulse rate FLOAT

- FLOAT must be a real number.
- Describes how frequently a phenomenon node broadcasts its presence.
- Defaults to 1 broadcast per second.

b) phenomenon PATTERN

- PATTERN must be any one of the following keywords: CO, HEAVY GEO, LIGHT GEO, SOUND, TEST PHENOMENON corresponding to Carbon Monoxide, heavy seismic activity, light seismic activity, audible sound, and some other generic phenomenon.
- This option is mostly useful for simulations involving multiple phenomenon nodes, so that it is easier to distinguish who a sensor node is detecting by looking at the ns trace file.

- Defaults to TEST PHENOMENON.

The following source code illustrates how these phenomena parameters can be set to emanate

Carbon Monoxide 10 times per second:

```
[$node_(0) set ragent_] pulserate .1 ;
[$node_(0) set ragent_] phenomenon CO ;
```

5) *Configure sensor nodes.*

Sensor nodes must be configured with the -PHENOMchannel attribute and the -channel attribute. PHENOMchannel must be the same as the channel you configured the phenomenon node with. The other channel is the channel that will be used for communicating sensor reports. Sensor node configurations must also specify a MAC protocol for the phenomena channel and a MAC protocol (such as Mac/802 11) for the channel shared with other wireless nodes. This is done with the -PHENOMmacType and -macType attributes. PHENOMmacType should be the same as the macType used in PHENOM nodes, and macType should be the same as the macType used in other nodes participating in the IP network. For example:

```
$ns node-config \
-adhocRouting $val(rp) \
-channel $chan_2 \
-macType $val(mac) \
-PHENOMchannel $chan_1 \
-PHENOMmacType $val(PHENOMmac)
```

If desired, a sensor node can be configured so that a specified amount of energy will be deducted from its energy reserve each time it receives a phenomenon broadcast. To set this up, include the following parameters in the sensor node's node-config routine:

```
-energyModel EnergyModel \
-rxPower 0.175 \
-txPower 0.175 \
-sensePower 0.00000175 \
 -idlePower 0.0 \
 -initialEnergy 0.5
```

Where

- rxPower .175 indicates 175mW consumed for receiving a packet of arbitrary size,
- txPower .175 indicates 175mW consumed for transmitting a packet of arbitrary size,
- sensePower .00000175 indicates 1.75μW consumed for receiving a PHENOM broadcast packet,
- initialEnergy 5 indicates a total energy reserve of 5J.

6) *Configure non-Sensor nodes, such as data collection points, or gateways for the sensor network.*

Nodes that are not sensor nodes or phenomenon nodes should not be configured with a PHENOMchannel, since their only interface is to the MANET network. This is done with the -PHENOMchannel "off" attribute, as follows:

```
$ns node-config \
-adhocRouting $val(rp) \
-channel $chan_2 \
-PHENOMchannel "off"
```

7) *Attach sensor agents*

Create a sensor agent for each sensor node, and attach that agent to its respective node. Also, specify that all packets coming in from the PHENOM channel should be received by the sensor agent. In the following example, $i would represent the node number for the sensor node currently being configured.

```
set sensor_($i) [new Agent/SensorAgent]
$ns attach-agent $node ($i) $sensor_($i)
 [$node ($i) set ll_(1)] up-target $sensor_($i)
```

8) *Attach a UDP agent and sensor application to each node (optional).*

How the sensor nodes react once they detect their target phenomenon is a behaviour that should be defined in the sensor application. One such application might involve sensor nodes alerting a data collection point via UDP with information about the phenomenon. The following example illustrates how an application like that could be

setup. Again, $i represents the node number for the sensor node currently being configured.

```
set src_($i) [new Agent/UDP]
$ns attach-agent $node ($i) $src_($i)
$ns connect $src_($i) $sink
set app_($i) [new Application/SensorApp]
$app_($i) attach-agent $src_($i)
```

9) *Start the sensor application.*

The sensor node can receive PHENOM packets as soon as the sensor agent is attached to the node. Since the sensor agent does nothing but notify the sensor application of received phenomenon broadcasts, the sensor node does not visibly react to PHENOM packets until the sensor application has been attached and started. The following example shows how to start a sensor application:

```
$ns at 5.0 "$app_($i) start $sensor_($i)"
```

5.3 NS-2 Intrusion Detection System Implementation

Our proposed IDS system is simulated using NS-2 simulator. It is a module captures all packets and imposes intrusion detection analysis. Currently, this module is implemented as a trace module, by modifying *cmu-trace.cc* and some other related source files.

trace/idsclass.cc, idsclass.h: Each node has the intrusion detection module. Intrusion detection system remains in one of the three states: LEARNING, ACTIVE and DEACTIVE. It provides functionality to activate or deactivate IDS of the sensor node which is based on proposed monitor selection algorithm. It provides function for packet capture and extracts the information from buffered packets. Then it applies IDS rules for detection of any malicious behaviour.

trace/idstable.cc, idstable.h: Each node maintains the malicious nodes list, direct neighbour and two hop neighbour list. Each node also have IDS rule database which is used for detection of malicious behaviour of sensor nodes in the network. Each node uses pre-defined rules and neighbour's list to watch over the communication in

their neighbourhood. When a sensor node receive a packet from a sensor node in the networks, if the intermediate sender node is in its malicious node list, which means the packet comes from a malicious node, then the sensor node drop the packet .

trace/idsapp.cc, idsapph: When a malicious node is detected by its monitor node, an alert message is sent to the base station. After arrival of alert message at base station, base station broadcasts message to isolate the malicious node from the network. Sensor node inserts the entry in their malicious node list when they receives isolation message from base station. This class provides functionalities foe communication between the base station and sensor node and also schedules the IDS activation of sensor nodes in the network.

trace/monitor.cc, monitor.h: Each sensor node has IDS which consists of two agents: local IDS agent and global IDS agent., Local IDS agent remains active all time and is responsible to monitor the information sent and receive by the sensor node. Global IDS agent of sensor node gets activated through K_m-monitor eligibility algorithm. In eligibility algorithm, sensor node maintains the list of one hop and two hop global IDS agent activated neighbours and checks its eligibility to activate its global agent. This class implements the K_m-monitor eligibility algorithm.

In next chapter we discuss the results of the simulation of our proposed IDS frame work.

Chapter 6
RESULTS AND DISCUSSION

6.1 Testing Methodology

Our simulation was based on the sensor network package from the Naval Research Laboratories, running on the NS-2 simulator platform (version 2.27). Our simulation scenarios used 100-200 sensor nodes, one base station and one phenomenon node. Nodes are randomly distributed in a 500 *500 m^2 coverage region and remain stationary once deployed. The phenomenon node represents a moving object, which is being tracked by the mobile sensor nodes. The sensor nodes use the Constant Bit Rate transport protocol, and use AODV as the routing protocol. The movement of phenomenon node was randomly generated over the field. Nodes in our simulations use radios with a 2 Mbps bandwidth and a sensing range of 50m. We used TwoRayGround radio propagation model in all NS-2 simulations. Each simulation was over a time period of 600 simulation seconds.

We implemented four attacks: Sybil attack, Wormhole attack, DoS attack and Sinkhole attack. The idea of wormhole implementation is making a "tunnel" between two malicious nodes. It simulates the "tunnel" by copying the local packets and directly sending them to the WORMHOLE object installed in a remote node. In Sybil attack, we attached a number of interfaces to a single node and it advertises itself as a new node for each attached interface. The idea of DoS attack implementation is to increase the frequency of sending packets to a particular node (target node). In Sinkhole attack, malicious node broadcasts packets containing false information of closest route to the base station. SINKHOLE object broadcasts close_route packets in to network. We tested proposed model for all above attacks and obtained the results described below.

6.2 Analysis

Each sensor node in the network has the intrusion detection system. To prolong the network lifetime, we used the K_m-monitor eligibility algorithm to selectively activate global IDS modules of sensor nodes. It minimizes the active global IDS modules in the network and maintains whole network monitoring condition. Figure 6.1 shows the

global IDS module activated nodes in the sensor network by K_m-monitor eligibility algorithm.

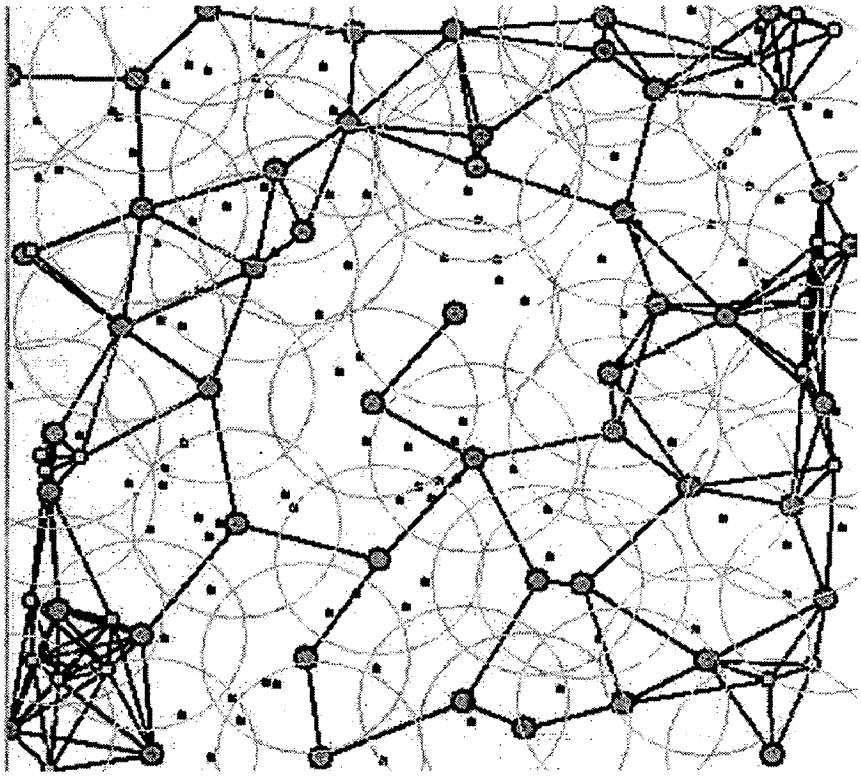

Figure 6.1 Activation of global IDS agent on sensor nodes in wireless sensor network.

Figure 6.2 shows the results of our scheme. For all four attacks, simulation gives the same result. On average, few nodes have to activate the global intrusion detection modules and the percentage of monitor nodes is quite stable. K_m-monitor eligibility algorithm activates only those sensor nodes' global agent which are required to monitor the every sensor node for a given monitor degree. Hence the percentage of monitor nodes remains quite stable for given monitor degree. If monitoring degree increases, number of monitor nodes increases. In some situations, sensing area of sensor nodes is reduced gradually due to battery depletion. Monitor nodes detect the behaviour of neighbour nodes so the percentage of monitor nodes in network is depend on radio range and network density. Our proposed model gives the good result on percentage of monitor nodes as compare to watchdog, in which every node participate in monitoring the neighbouring nodes.

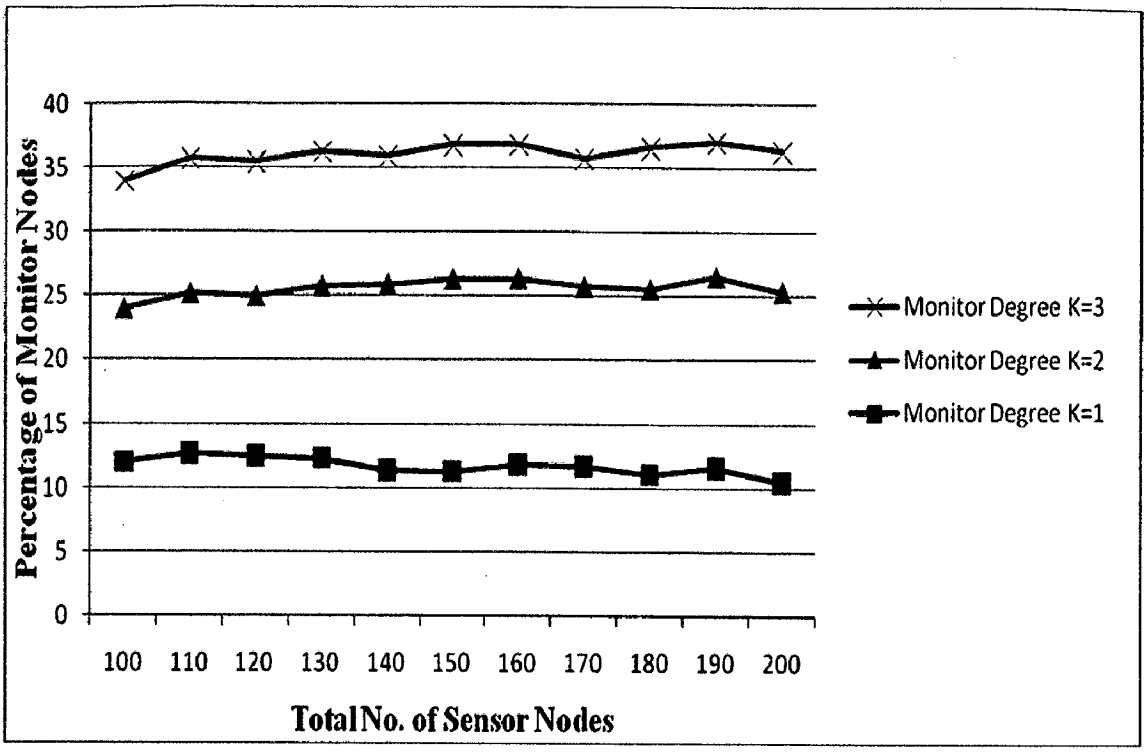

Figure 6.2 Percentage of monitor nodes with different network density

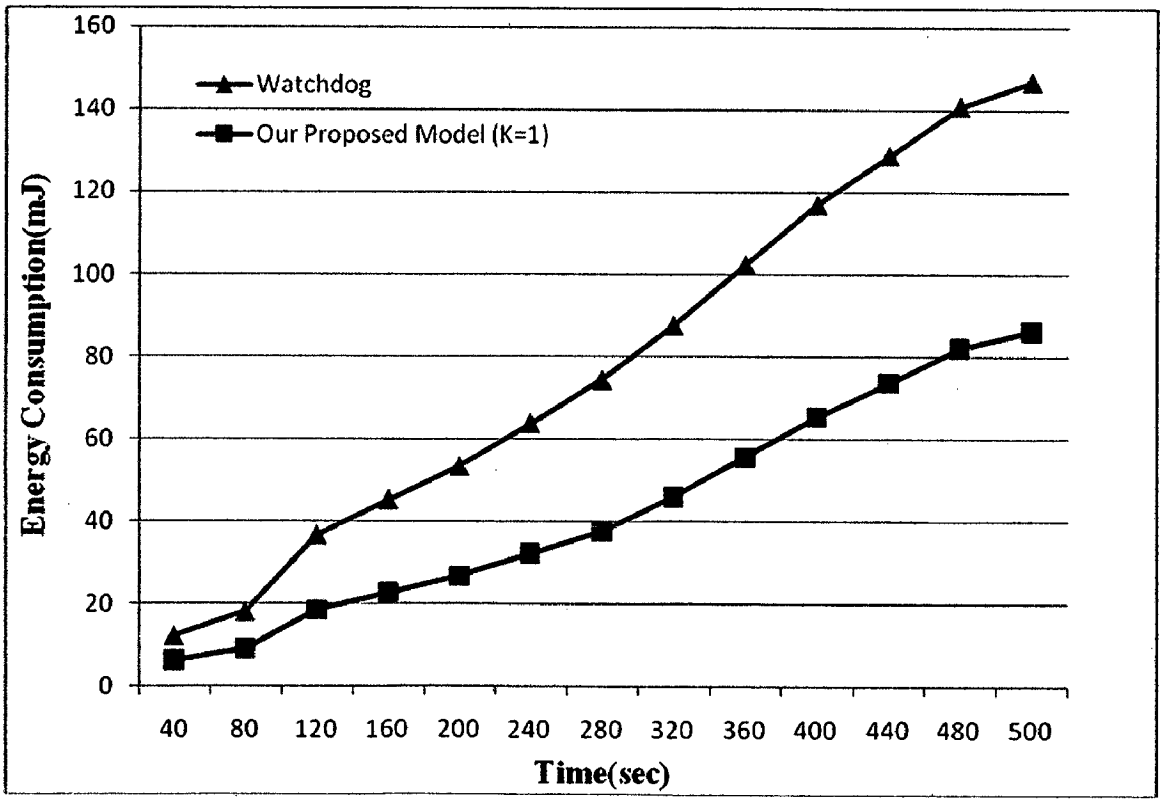

Figure 6.3 Energy consumption of monitor nodes

We obtain the energy consumption of monitor nodes as shown in Figure 6.3. For all four attacks, simulation gives the same result. Here we showed energy consumption of monitor nodes for monitor degree one. As the monitor degree increases, number of monitor nodes per sensor node in the network increases and energy consumption of monitor nodes also increases. It is apparent that our model consumes less energy than watchdog mechanism. As the number of monitor nodes in our proposed model are lesser, energy consumed by monitor nodes gets reduced as compared to watchdog, in which every node participates in monitoring the neighbouring nodes.

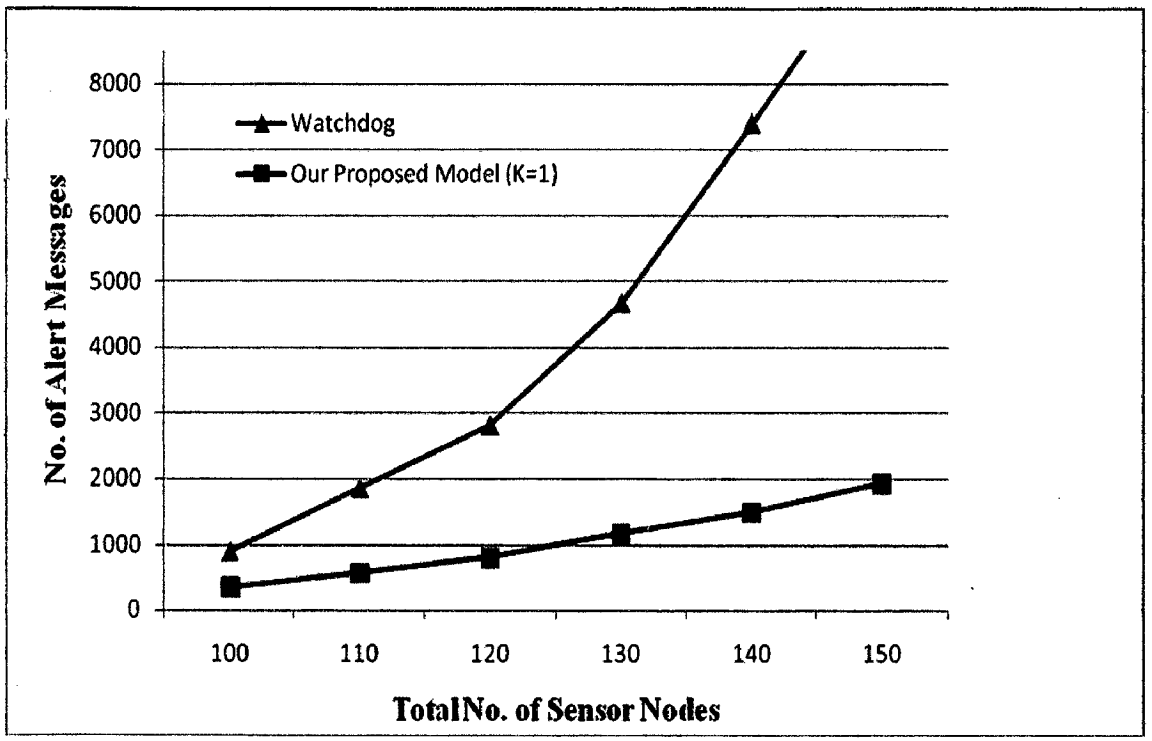

Figure 6.4 Number of alert messages with diffrent network density

Monitor nodes monitor neighbouring nodes' behaviour. If there ia any misbehaviour in neighbouring node, monitor node detects it as an intrusion and sends the alert message to the base station or sink node. In case of watchdog, every node monitors neighbouring nodes. If an intrusion is detected, each monitor node monitoring that specific node sends alert messages to base station or sink node i.e number of alert message increases exponentially as the monitor node increases. In our model, monitor nodes are fewer than in watchdog mechanism. Hence the number of alert messages

are also less than number of alert messages in watchdog as shown in Figure 6.4. For all four attacks, simulation gives the same result.

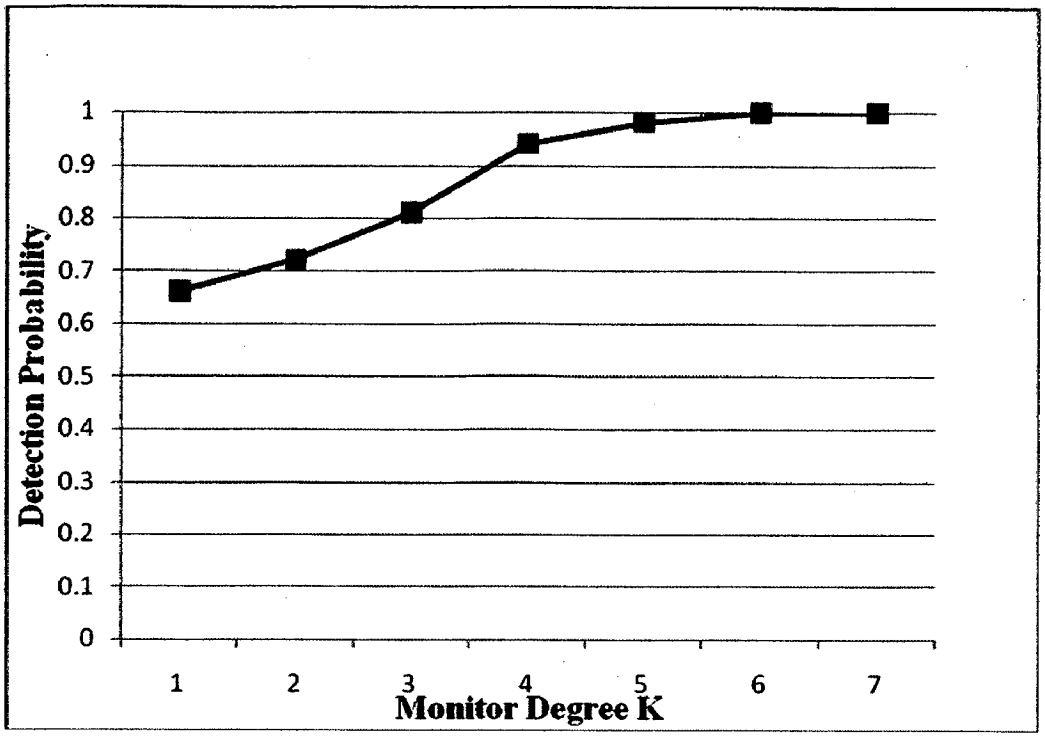

Figure 6.5 Monitor Degree vs. Detection Probability

Detection probability of intrusion detection increases as the number of monitor node per sensor node (monitor degree K) increases. The probability of detection is close to 1 if the number of monitor degree is more than 5. For all four attacks, simulation gives same result as shown in Figure 6.5.

Chapter 7
CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

In this dissertation, we focused on IDS model which consists of two intrusion detection modules: local IDS agent and global IDS agent. Local IDS agent monitors the information sent and received by the sensor. Global IDS agent monitors the neighbouring nodes' behaviour. Local IDS agent remains active all time where as global IDS agent gets activated according to the K_m-monitor eligibility algorithm. In K_m–monitor algorithm, node decides that its global IDS agent should be activated on the basis of its one hop and two hop neighbouring nodes' global IDS agent state and monitoring degree provided by base station. We simulated the proposed IDS model in NS-2. Our model minimizes sensor nodes activating global intrusion detection modules thus reduces energy usage on other nodes as compared to *watchdog* and enhances the network lifetime. It also keeps broadcast alert messages to a minimum and reduces collisions in the network. Our model had much better performance than *watchdog technique*, which has been applied in many previous works.

7.2 Suggestions for Future Work

Future work may consider

1. A dynamically change of monitor degree to suit each particular situation. Given that nodes can turn on IDS fairly quickly, it is natural to consider adaptive strategies in responding to the threat as it develops.

2. In this dissertation, we considered static sensor network. There is a scope for extending this work to use it in mobile sensor network.

3. There is a scope to investigate light-weighted decentralized approaches, and systematically analyze its benefits and inherent weakness when compared with centralized approaches.

REFERENCES

[1] A. Perrig, J. Stankovic, and D. Wagner, "Security in Wireless Sensor Networks", Communications of the ACM, Vol. 47, No. 6, June 2004, pp. 53-57.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," IEEE Communication Magazine, August 2002, pp. 102-114.

[3] A. P. Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. C. Wong, "Decentralized Intrusion Detection in Wireless Sensor Networks", In Proceedings of the 1st ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks, Q2SWinet '05, Montreal, Quebec, Canada, October 2005, pp. 16-23.

[4] M. Tubaishat, and S. Madria, "Sensor Networks: An Overview", IEEE Potentials, Vol. 22, No. 2, April/May 2003, pp. 20-23.

[5] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and Countermeasures", In Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications, Anchorage, AK, USA, May 2003, pp. 113-127.

[6] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil Attack in Sensor Networks: Analysis and Defenses", In Proceedings of the 3rd ACM International Symposium on Information Processing in Sensor Networks, Berkeley, California, USA, April 2004, pp. 259-268.

[7] A. Wood and J. Stankovic, "Denial of Service in Sensor Networks", IEEE Computer Society, Vol. 35, Issue 10, October 2002, pp. 54-62.

[8] E. Shi and A. Perrig, "Designing Secure Sensor Networks", IEEE Wireless Communications, Vol. 11, Issue 6, December 2004, pp. 38-43.

[9] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks", In Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, SenSys'03, Los Angeles, California, USA, November 2003, pp. 28-39.

[10] Tim Crothers, "Implementing Intrusion Detection Systems: A Hands-On Guide

for Securing the Network", 1st edition, WILEY Publisher, 2003.

[11] B. Sun, L. Osborne, Y. Xiao, and S. Guizani, "Intrusion Detection Techniques in Mobile Ad Hoc and Wireless Sensor Network", IEEE International Conference on Wireless Communication, Vol. 14, Issue 5, October 2007, pp. 56-63.

[12] R. Roman, J. Zhou, and J. Lopez, "Applying Intrusion Detection Systems to Wireless Sensor Networks", In Proceedings of 3rd IEEE Consumer Communications and Networking Conference, CCNC 2006, Las Vegas, Nevada, USA, January 2006, pp. 640-644.

[13] I. Onat and A. Miri, "An Intrusion Detection System for Wireless Sensor Networks", In Proceedings of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WiMob 2005, Los Alamitos, California, USA, August 2005, pp. 253-259.

[14] S. Banerjee, C. Grosan, and A. Abraham, "IDEAS: Intrusion Detection Based on Emotional Ants for Sensors", In Proceedings of 5th International Conference on Intelligent Systems Design and Applications, ISDA 2005, Wroclaw, Poland, September 2005, pp. 344-349.

[15] P. Techateerawat and A. Jennings, "Energy Efficiency of Intrusion Detection Systems in Wireless Sensor Networks", In Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence and International Agent Technology Workshops, WI-IATW'06, Hong Kong, China, December 2006, pp. 227-230.

[16] W. Du, L. Fang, and P. Ning, "LAD: Localization Anomaly Detection for Wireless Sensor Networks", In Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium, IPDPS'05, Denver, Colorado, USA, April 2005, pp. 41-47.

[17] Q. Zhang, T. Yu, and P. Ning, "A Framework for Identifying Compromised Nodes in Sensor Networks", In Proceedings of 2nd IEEE Communications Society/CreateNet International Conference on Security and Privacy in Communication Networks, SecureComm 2006, Baltimore, Maryland, USA, August 2006, pp. 1-10.

[18] S. Cheng, S. Li, and C. Chen, "Distributed Detection in Wireless Sensor Network", In Proceedings of 7th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2008, Paris, France, December 2008, pp. 401-406.

[19] Y. Huang and W. Lee, "A Cooperative Intrusion Detection System for Ad Hoc Networks," In Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks, SASN '03, Fairfax, Virginia, USA, October 2003, pp. 135-147.

[20] I. Khalil, S. Bagchi, and N. B. Shroff, "LITEWORP: A Lightweight Countermeasure For The Wormhole Attack in Multihop Wireless Networks", In Proceedings of International Conference on Dependable Systems and Network, DSN 2005, Yokohama, Japan, June 2005, pp. 612-621.

[21] T. H. Hai, F. Khan, and E. Huh "Hybrid Intrusion Detection System for Wireless Sensor Networks", In Proceedings of International Conference Computational Science and Its Applications, ICCSA 2007, Kuala Lumpur, Malaysia, August 2007, pp. 383-396.

[22] Q. Wang and T. Zhang, "Detecting Anomaly Node Behavior in Wireless Sensor Networks", In Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops, AINAW-2007, Niagara Falls, Ontario, Canada, May 2007, pp. 451-456.

[23] G. Huo, X. Wang, "DIDS: A Dynamic Model of Intrusion Detection System in Wireless Sensor Network", In Proceedings of International Conference on Information and Automation, ICIA 2008, Hunan, China, June 2008, pp. 374-378.

[24] R. Chen, C. Hsieh, Y. Huang, "A New Method for Intrusion Detection on Hierarchical Wireless Sensor Networks", In Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication, Suwon, Korea, January 2009, pp. 238-245.

[25] Kevin Fall, and Kannan Varadhan, "The ns Manual", VINT Project, April 2002, pp. 1-366.

[26] The VINT Project, "The Network Simulator - ns-2". [Online], Available: http://www.isi.edu/nsnam/ns/.

[27] I. Downard, "Simulating Sensor Networks in NS-2", Technical Report NRL/FR/5522-04-10073, Naval Research Laboratory, Washington, D.C., USA, May 2004, pp. 1-9.