修士論文

# A Study on Efficient Architecture for Intrusion Detection System in Ad Hoc Networks

# アドホックネットワークにおける侵入検知システムのためのアーキテクチャの研究

東京大学大学院 新領域創成科学究科 基盤情報学専攻
学籍番号 47-66346

Ritonga Muhammad Arifin

指導教員　中山 雅哉 准教授

2008年1月29日 提出

# Abstract

In recent years, security has become a major concern in ad hoc network technology. It has unique characteristics that make new challenges arise compared to its wired networks or traditional wireless networks counterpart. Due to these unique characteristics, the traditional intrusion detection techniques for wired networks and wireless networks are not well suited for ad hoc networks.

In this document, we analyze the system architecture that fits into ad hoc network environment and propose a Manager-based architecture for intrusion detection system in ad hoc network. We have evaluated the performance of proposed system through simulation-based experiment and compared it with other existing methods. The result shows that our method gives the better performance.

# Acknowledgements

I would like first to thank my advisor, Prof. Masaya Nakayama, for providing invaluable guidance in my research. His broad knowledge and deep insights help me to carry out this research.

I would also like to thank Prof. Yasushi Wakahara, Dr. Fumitaka Nakamura and Dr. Yuji Sekiya for giving me invaluable advice to my study in the laboratory meeting.

I also wish to acknowledge all my colleagues at the campus network laboratory, for their insightful discussions on research.

Finally, I am also grateful to the staff of the Information Technology Center for their support especially the secretary Yoshizawa and the secretary Kawasaki.

# TABLE OF CONTENTS

# Figure List

# Table List

# Chapter 1

# INTRODUCTION

## 1.1 Introduction and Motivation

History has noted the proliferation of microprocessor. It has been a trend to embed it into everything: cars, mobile phones, refrigerators, digital cameras, etc. It is predicted that short-range wireless transceiver will follow microprocessor to be a great trend in the near future. Many electronic devices will become more useful and effective by forming a network and communicating with each other. It is wireless ad hoc network technology that supports this new paradigm of networking.

Ad hoc network is a type of network that allows the members of the network to directly communicate to each other within the network without any fixed infrastructure such as access points or base stations. In this network, one node functions as router as well as the end point. Due to this special characteristic, ad hoc network experiences more vulnerabilities that bring more security concerns and challenges compared to other infrastructured networks.

As the popularity of this network increases, vulnerabilities of this network are also predicted to increase and eventually leads to emergence of many types of new attacks specified for this type of networks. Therefore, the security research in ad hoc network environment is important. Moreover, as the lessons from wired network technology, security in ad hoc network should also be in-depth security, which means that there should be mechanism to detect the attacks when the first security shield is broken. Therefore, we want to focus our work on intrusion detection approach.

In the situation of providing intrusion detection system in ad hoc network, among many challenges that ad hoc network faces, the unavailability of infrastructure seems to be the main problem. Thus, in this work, we plan to create a foundation of realizing intrusion detection system in ad hoc network, in other words, a system architecture for implementing the intrusion detection system in ad hoc networks. The architecture should fit to the special characteristics of ad hoc network.

## 1.2 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 gives an introduction to ad hoc networks and the security goals and challenges in ad hoc networks. Two type of approaches to secure ad hoc network are presented, along with some examples of each. A brief explanation about the background of intrusion detection system is also presented in this chapter.

Chapter 3 presents the related works in the field of architecture for intrusion detection system in ad hoc network. In the same chapter, two existing works are brought up as examples and also for subject of comparison later in this thesis.

Chapter 4 explains about the design of the proposal system. In order to give a clearer view of the system, the algorithm is explained with examples and flowcharts.

Chapter 5 presents the evaluation of our proposal system. The evaluation is divided into two parts: evaluation in static networks and dynamic networks.

Lastly, chapter 6 concludes this thesis along with an extension plan for the future.

# Chapter 2

# AD HOC NETWORKS AND SECURITY

In this chapter, we first illustrate ad hoc networks, the type of network that we base our work on, and the challenges to provide secure communication on that type of networks. We then illustrate why Intrusion Detection Systems (IDSs) are necessary when we deploy ad hoc networks in reality. After brief introduction on IDSs in infrastructure network, we then discuss why these IDSs are not applicable to ad hoc networks. Finally, some research questions to be answered in order to develop IDS for ad hoc networks will conclude this chapter.

## 2.1 Ad Hoc Networks

In the recent years, wireless networks have gained a tremendous popularity in both research and industry. There are at least two variations of mobile networks. The first is known as infrastructure networks because they have fixed and wired infrastructure such as gateways or routers that connect them to other networks. The bridges in these networks are also known as base stations. In an environment like this, a node is able to move freely and establish a connection with the nearest base station that lies within its communication range. When the mobile node moves out of the range of one base station that it was connected with, it goes into the range of another base station. A hand-off process that occurs between the old base station and the current one enables the node to continue communication seamlessly through the network. These types of networks are most widely applied in daily life such as mobile telephone networks and the wireless local area networks (WLANs) in office areas.

The second type of wireless networks is the *infrastructureless mobile network* that is also known as ad hoc network. Ad hoc networks have no fixed routers or base stations and the participating nodes are capable of movement. Due to the limited transmission range, multiple hops may be required for nodes to communicate across the ad hoc network. Routing functionality is incorporated into each host, thus ad hoc networks can be characterized as having dynamic, multi-hop, and constantly changing topologies. Example scenarios for the application of ad hoc networks include search and rescue operations, meetings or conven-

tions in which persons wish to quickly share information and data acquisition operations in inhospitable environments.

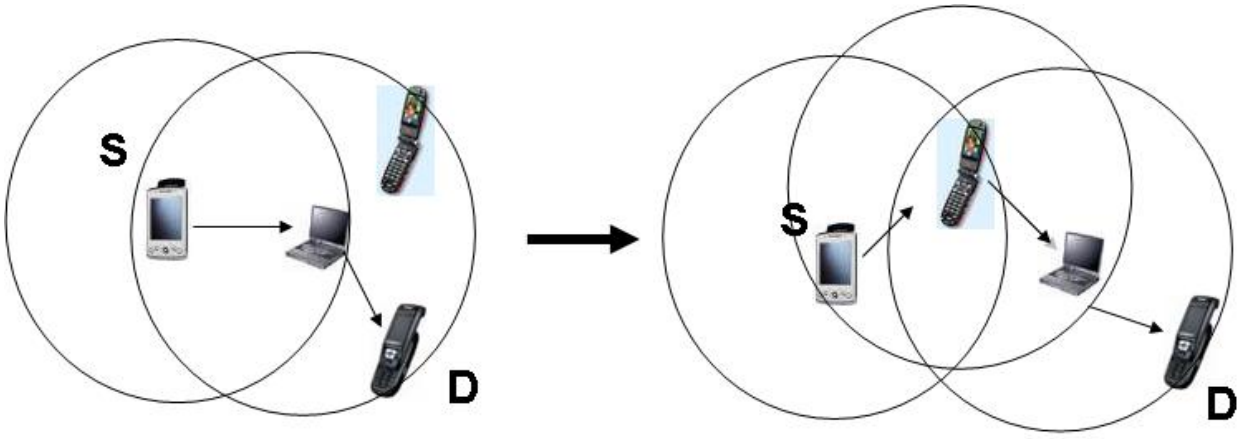



Fig. 2.1 Ad Hoc Network Example

Figure 2.1 illustrates an example ad hoc network. The network is formed by independent mobile nodes such as PDA, mobile phones, and laptop that have wireless transceivers. Each circle illustrates the communication range of the node in its center. In the left side of this example, we can see that PDA that acts as source communicates with a destination, mobile phone, outside its communication (transmission) range through an intermediate node, a laptop, that locates within transmission range of source node and destination node. Moreover, mobile nodes that construct this ad hoc network can move freely inside the network. This mobility results to the dynamic change of the network topology, as shown in right side of Figure 2.1. The participating nodes in ad hoc networks act both as end hosts and routers, forwarding traffic from the source to the destination.

### 2.1.1 Properties of Ad Hoc Networks

Generally, there are 4 major properties of ad hoc networks. Brief explanation of each property is given in following subsections.

#### Infrastructureless

As already mentioned, ad hoc networks do not rely on pre-existing infrastructure and this may be their most distinguishing attribute. Instead ad hoc networks are formed by individual nodes when they come to close proximity and need to communicate with each other. This implies that there is no need for stationary components such as routers, bridges and cables and of course central administration is not required.

Due to the lack of stationary infrastructure, the participating nodes in the ad hoc network have to forward traffic on behalf of other nodes that are not in close proximity to the destination node. If they deny participating in the routing process, the connectivity between

nodes may be lost and the network could be segmented. Therefore, the functionality of an ad hoc network heavily depends on the forwarding behaviour of the participating nodes.

#### Shared Wireless Medium

Hosts that construct an ad hoc network typically communicate with each other using wireless channels; they can also communicate with each other by using other hosts as intermediate hops in the communication path (multi hops communication).

#### Dynamic Network Topology

Another important property of ad hoc networks is their dynamic topology. Since the topology arbitrarily changes due to node mobility and changes of the surrounding environment, the utilised routing protocols have to be able to adapt to the dynamic topology. Traditional wired routing protocols like OSPF do not incorporate in their normal operation support for frequent network topology changes. Thus, the routing protocols that are currently utilised in ad hoc environments have specifically been designed to handle node mobility and rapidly changing topologies.

#### Resource Constraints

The devices that are usually employed in the ad hoc networks have their own limitations. Since the only hardware component that is required to connect a device in an ad hoc network is a wireless interface, PDAs and mobile telephones can also be utilised. Furthermore, differences in the radio transmission ranges and reception equipment sensitivities may lead to unidirectional links which could complicate routing in the ad hoc networks. Apart from the communication differences between the nodes, ad hoc networks suffer from limited hardware resources like limited battery, constrained CPUs and small memory capacity.

### 2.1.2 Ad Hoc Routing Protocols

In order to adapt to the special properties explained in the previous section, ad hoc networks adopt special routing protocols. Many routing protocols have been proposed for ad hoc networks. In general, these protocols could be divided into two categories: proactive and reactive. Proactive routing protocols, such as Destination-Sequenced Distance Vector routing protocol (DSDV) and the Wireless Routing Protocol (WRP), waste limited bandwidth by continuously maintaining the complete routing information about the whole network. They react to topology changes, even if there is no traffic. They are also called table-driven methods. The protocols in this area differ in the number of tables maintained, the information each table contains as well as the details of how they are updated. Reactive routing protocols, such as Ad hoc On-demand Distance Vector routing protocol (AODV), and the Dynamic Source Routing protocol (DSR), are based on demand for data transmission. They

can significantly reduce the routing overhead when the traffic is lightweight and the topology changes less dramatically, since they do not need to periodically update route information and do not need to find and maintain the routes when there is no traffic. The differences among reactive routing protocols lie in the implementation of the path discovery mechanism and optimizations to it.

## 2.2 Ad Hoc Network Security

As in infrastructured networks, ad hoc networks also suffer from security threats. Moreover, since recently this type of networks gains more popularity, it is predicted that in near future, more threats and attack schemes will emerge. Therefore, ad hoc networks need to take the countermeasures for these threats. In this section, we describe about security requirements that ad hoc networks need to take in order to secure the networks. We then describe the challenges to provide security due to the special properties of the networks. Finally, we describe some approaches to increase the security in ad hoc networks.

### 2.2.1 Security Requirements

The security services in ad hoc networks are not different from any other infrastructured networks. The goal of these services is to protect information and resources from attacks and misbehavior. In dealing with network security, the following requirements should be met in order to ensure an effective security system.

- **Availability :** Ensures that the desired network services are available whenever they are expected, in spite of the presence of attacks. Systems that ensure availability in ad hoc networks are the ones that can counter denial of service and sleep deprivation) attacks as well as selfish legitimate nodes that refuse to forward packets.
- **Authentication :** Ensures that communication from one node to another is genuine. In other words, it ensures that a malicious node cannot masquerade as a legitimate nodes.
- **Data Confidentiality :** Ensures that a given message cannot be understood by anyone other than the designated recipients. Data confidentiality is typically enabled by applying symmetric or asymmetric data encryption.
- **Integrity :** Ensures that a message has not been altered by malicious node during its transmission.
- **Non-repudiation :** Ensure that a node cannot deny the message it has sent. Usually, digital signatures are used to ensure this.

### 2.2.2 Security Challenges

Due to the special characteristics explained in Section 2.1.1, ad hoc network experiences more vulnerabilities that bring more security concerns and challenges compared to other infrastructure networks. Explanation of the security challenges in each ad hoc networks property are given in the following subsections.

#### Infrastructureless

Central servers, specialized hardware, and fixed routers are not available. The unavailability of such infrastructures prevent security solution that has been deployed in wired networks ineffective in ad hoc network. Firewall, which is a powerful security tool in wired network is also not available due to unavailability of infrastructure. PKI (Public Key Infrastructure), which is a reliable infrastructure to provide authentication is also not available. Therefore, new methods that is not relying on centralized method should be deployed in ad hoc networks.

#### Shared Wireless Medium

Wireless link usage renders ad hoc networks susceptible to attacks. Unlike wired networks, in which an adversary must gain physical access to the network's wires or pass through several lines of defense at firewalls and gateways, attacks on a wireless ad hoc network can come from all directions and target any node. Therefore, ad hoc networks will not have a clear line of defense, and every node must be prepared to defend against threats. Moreover, the MAC protocols used in ad hoc networks, such IEEE802.11, rely on trusted cooperation in a neighborhood to ensure channel access, which leads to high vulnerability.

#### Dynamic Network Topology

Mobile nodes are generally autonomous units that are capable of roaming independently. This means that tracking down a particular mobile node in a large-scale ad hoc network cannot be done easily. Moreover, nodes are allowed to enter and leave the network spontaneously, i.e. to form and break links unintentionally. Therefore, the network topology has no fixed form regarding both its size and shape. Any security solution must take this feature into account.

Another problem that has to be considered seriously is internal attack. Since nodes in this network are usually mobile devices that move in and out the network, a malicious user can capture and tamper them outside, and once they get back into the network, they can be used to access the network because they have the standard security keys for the network. This makes even a perfect crypto system will not be sufficient to secure ad hoc networks.

#### Resource Constraints

There are many problems caused by this characteristic. The first problem is due to power limitation. Ad hoc enabled mobile hosts are small and lightweight, and they are often supplied with limited power resources. This limitation causes a vulnerability, for example, attackers may target some nodes' batteries to disconnect them, which may lead to a network partition. This is called sleep deprivation attack.[1)]

The second one is due to memory and computation power limitation. Mobile nodes have limited storage devices and weak computational capabilities. Consequently, high complexity security solutions, such as symmetric or asymmetric data encryption, are difficult to implement.

The last one is due to mobile devices physical vulnerability. Mobile devices used in ad hoc network, and in mobile networks in general, are lightweight and portable. This represents a vulnerability, since the devices and the information stored in the devices can be easily stolen. Mechanisms for protecting both devices and information should be employed.

### 2.2.3 Security Schemes

In the realm of computer network security research, there are two main approaches in securing the network. The first is intrusion prevention approach. Researches on secure routing are main representatives of this approach. This approach aims in designing and implementing routing protocols that have been designed from scratch to include security features. Mainly the secure protocols that have been proposed are based on existing ad hoc routing protocols like AODV and DSR but redesigned to include security features. The second approach is the intrusion detection approach that aims in enabling the participating nodes to detect and avoid malicious behaviour in the network without changing the underlined routing protocol or the underline infrastructure. Although the intrusion detection field and its applications are widely researched in infrastructure networks it is rather new and faces greater difficulties in the context of ad hoc networks.

In the following section we briefly present the two approaches in realizing security schemes that can be employed in ad hoc networking environments.

#### Intrusion Prevention Approach

Many proposed approaches are applicable to secure routing in ad hoc networks. Secure AODV[6)] and Ariadne[2)] are some of them. This approach is efficient to deal with external attacks as we can assume that the attackers don't have access to the network cryptosystem, e.g. they don't have private key that is accepted within the network. Thus, digital signatures scheme can be used to protect information authenticity and integrity. In such scheme, a pair of private-public key is needed to sign and verify the data. While the robustness and the efficiency of the key is important in ad hoc networks, the key management also plays a great

role to realize a secure system. One approach to do the key management is by establishing a Certification Authority (CA) to issue the key. However, since it is hard to centralize CA in ad hoc networks, key management is also actively researched recently.

Zhou and Haas propose distributed CA function over multiple nodes by employing *(t, n) threshold cryptography*.[5)] The system can tolerate t-1 compromised servers. However, this scheme doesn't describe how a node can contact t servers securely and efficiently in case the servers are scattered in the whole area. Luo, Kong, and Zerfos propose a localized key management scheme called URSA.[3)] In this scheme all nodes are servers. The advantage of this scheme is efficiency and secrecy of local communication as well as system availability. On the other hand, it reduces the system security especially when nodes are not well protected.

The usage of CA is reasonable if nodes have permanent address. However, addressing in ad hoc networks can also follow recent trends towards dynamic address allocation and auto-configuration. In that case, one feasible solution would be to pick a key pair, and map the public key to the address in some deterministic way. O'Shea and Roe propose Cryptographically Generated Address (CGA).[4)] This is relatively secure, although potentially expensive in computation.

However, intrusion prevention approach is not sufficient to secure ad hoc networks. Internal attack that is launched by malicious user through a compromised node is difficult to prevent because the compromised node usually has the access to the network's cryptosystem, that can be keys or passwords. Another reason why ad hoc networks need a second wall of defense is because there can be problems even to seemly perfect system due to unexpected even or bugs in the program, especially when the program is large and complex.

#### Intrusion Detection Approach

In recent networks, intrusion detection approach is absolutely needed as the second line of defense, completing intrusion prevention approach to realize more secured system. In the recent time, as ad hoc networks gain more popularity, special attacks that are crafted for this network also increase, make the former intrusion prevention approach ineffective. Therefore, we also need intrusion detection method to reactively detect attack to gain more time for improving the intrusion prevention method without resulting so many destruction in the network. As the intrusion detection will be the main of this report, we will discuss it more detail in the next section.

## 2.3 Intrusion Detection System

Intrusion Detection System (IDS) is a reactive method against intrusion. In the contrast of preventive method, which by analogy attributes to walls and locked doors, IDS serves as burglar alarms. It reacts when an intrusion have been or is occuring on the system. That is why IDS is also called the second line of defense. Generally speaking, an IDS is

not an antivirus program designed to detect malicious softwares such as viruses, Trojans, and worms. It is also not a network logging system used, for example, to detect complete vulnerability to any DoS attack across a network.

In this section, we explain briefly about the intrusion detection system in infrastructure networks which is the original point of such system. Then, we show some challenges why it is difficult to create an intrusion detection system for ad hoc networks. Finally, we discuss about research questions need to be solved in order to develop a viable intrusion detection system for ad hoc networks.

### 2.3.1 Intrusion Detection System in Infrastructure Networks

The pioneering work on IDS was done in 1980 by James Anderson. He wrote report for US Air Force proposed a method for filtering computer audit trails and detecting unusual usage patterns through statistical analysis. After that, the research on this field is getting more and more active, and in 1990 a team in University of California Davis developed NSM (Network System Monitor) which is the first IDS to analyze network traffic.

Intrusion detection can be classified based on audit data as either host-based, network-based, or the mixed approach of host-based and network based. Host-Based IDS (HIDS) monitors for attacks at the operating system, application, or kernel level. HIDS has access to audit logs, error messages, service and application rights, and any resource available to the monitored host. Network-Based IDS (NIDS) monitors traffic as it flows to other hosts.

IDS can also be classified, based on the detection method, into three categories.

- **Anomaly detection method:** In this method, a baseline profile of normal system is created and saved in the system. Then, the captured data which describes the current condition of the system will be compared with this profile. Some threshold value are used to determine whether the current condition can be judged as anomaly or accepted as normalcy. The difficulty to set the threshold is one disadvantage of this method. If the threshold value is set too high, it will increase the false positive, that is the anomaly which is detected as normalcy. In other hand, low threshold value will increase the false negative, that is the normalcy which is detected as anomaly. Moreover, anomaly that is not caused by intrusion also flagged as intrusive in this method.

- **Misuse detection method:** In misuse detection (also called signature-based detection), decisions are made on the basis of knowledge of the attack model. The system keeps the signatures of known attacks. Then, the captured data will be compared to these signatures and any matched pattern is treated as an intrusion. While this method is able to determine intrusion with relatively low false positive and false negative rate, it cannot detect new type of attacks. Moreover, the system needs a relatively larger memory to store the attack signatures and it keeps increasing as new signatures is inputted to the system.

- **Specification-based detection method:** In this method, the system defines a set of constraints that describe the correct operation of a program or protocol, and monitors the execution of the program with respect to the defined constraints. The capability to detect unknown attacks and relatively accurate detection are surely the advantages of this method. However, it only reacts to violations to the specified protocol or program. Another type of intrusions are more likely to be neglected.

an IDS should also, although not necessarily, have intrusion response system, preferably without human intervention. The type of intrusion response depends on the type of intrusion, the network protocols and applications in use, and the confidence in the evidence. The response system can also inform the user who may in turn do more investigations and take appropriate action.

### 2.3.2 Challenges of Intrusion Detection System in Ad Hoc Networks

Many intrusion detection systems (IDSs) have been proposed in wired networks. However, applying the research of wired network to ad hoc networks is not easy because of architectural differences. Among them the main difference is the lack of fixed infrastructure. The challenges of designing IDS for ad hoc networks can be attributed as follows:

- **Infrastructureless :** In wired networks, all traffic must go through switches, routers, or gateways. Hence, IDS can be implemented in those devices and audit data can be collected easily. We can use the previous analogy of burglar alarms. Normally, sensors are placed at common points of entry and exit. Logically, this strategy focuses on what it deems the weakest points in the structure and thus the most vulnerable to an intruder's attack. However, in ad hoc networks, there is no fixed infrastructure as the centralized audit points that acts as the entry and exit to the network. Every node is independent and possible to become door to outside of the network. Therefore, securing every node in ad hoc networks becomes crucial.

- **Dynamic Network Topology :** The algorithm that IDS uses must be distributed in whole networks, and should take into account the fact that a node can only see a portion of the network traffic. Moreover, since ad hoc networks are dynamic and nodes can move freely, there is possibility some nodes are captured and compromised, especially if the environment is hostile, such as battle field. If the system use cooperative algorithm among the nodes, then it is necessary to ensure which nodes one can trust.

- **Resource Constraints :** In detecting intrusions, ad hoc networks cannot communicate as frequently as wired networks in order to conserve bandwidth and other resources such as battery. Moreover, mobile devices have limitations in computation

and memory. Therefore, a heavyweight system is difficult to implement in mobile devices. These limitations also should be put into considerations in developing IDS for ad hoc networks.

These challenges lead us to some research questions that have to be answered in order to develop a reliable intrusion detection system for mobile ad hoc networks. The following lists the research challenges in developing a viable intrusion detection system for ad hoc networks:

- What is a suitable system architecture for building intrusion detection systems that fits to the characteristics of ad hoc networks?
- What are the suitable data sources to provide information about network condition? How do we detect anomaly based on partial, local audit traces - if they are the only reliable audit source?
- What is a good model of activities in ad hoc network environment that can separate anomaly when under attacks from the normalcy?

In this report, we mainly focus our target to solve the first question, an efficient system architecture for ad hoc networks. We discuss about existing works in this area in the next chapter.

## 2.4 Summary

In this chapter, we introduce about ad hoc network and security challenges it has due to its special characteristics. We also presented the security goals and challenges that the ad hoc networking faces in order to focus the research target on the security of ad hoc network fields. While most research about security in ad hoc networks focus on intrusion prevention approach, intrusion detection approach is definitely needed as this network type is gaining more popularity to attract many attacker to threaten the security of the networks. Besides, the lesson in wired network also teaches us not to only rely on one approach, instead, the usage of defense in depth. We presented three research questions to answer to develop a viable system.

# Chapter 3

# AD HOC NETWORK INTRUSION DETECTION SYSTEMS

The works on intrusion detection system (IDS) for ad hoc networks are relatively new. The first work known to this subject was done by Yongguang Zhang and Wenke Lee[9] in the year of 2000. Moreover, as ad hoc networks gain popularity and many specific applications are developed for the networks, researches on intrusion detection system for ad hoc networks start getting attention among researchers.

We focus our work to develop a system architecture that fits the characteristics of ad hoc networks to answer the first research questions stated in Section 2.3.2. Therefore, in this chapter, we firstly explain three types of system architecture for ad hoc network IDS, along with the merits and demerits of each type. Then, we also explain some study cases of existing works to provide more information about the real works in this field and position our work on the big map of researches in the field of IDS in ad hoc network.

## 3.1 Architectures of Intrusion Detection System in Ad Hoc Networks

First question that needs to be answered in order to develop a viable intrusion detection system in ad hoc network is the architecture of system. In this section, we will explain three types of architectures that have been proposed for IDS in ad hoc networks.

### Host-based Architecture

The main characteristic of this architecture is that every node runs an intrusion detection system agent and independently determines intrusions as shown in Figure 3.1. Every decision made is based only on information collected at local node, since there is no cooperation among nodes in the network. In this architecture, there is no data related to intrusion detection exchanged among other nodes in the network. Hence, nodes in the same network do not

know anything about the situation on other nodes in the network since no alert information is passed.

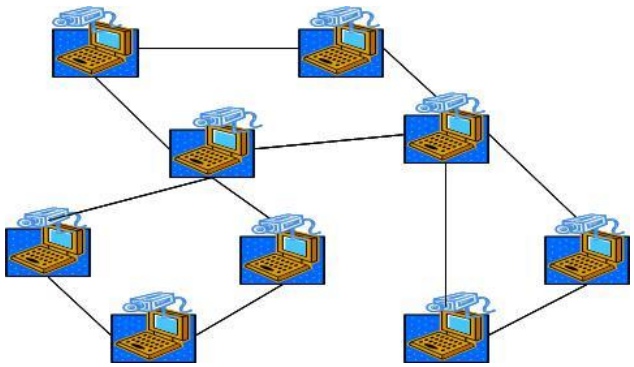


Fig. 3.1 Host-Based Type Architecture

The merits that can be expected with this type of architecture is that there is no network overhead for the intrusion detection process such as audit data exchange. Thus, we can expect the fast detection time since there is not necessary to wait for the data transfer. Moreover, this system could reduce the risk from the attacks type where attackers accuse legitimate nodes misbehaving in purpose to have those nodes excluded from the network.

However, this architecture has limitations to be implemented in real environment because in most of attacks type, information on each individual node might not be enough to detect intrusions.

### Hierarchical Architecture

The second type of architecture is hierarchical model. In hierarchical architectures, networks are divided into smaller sub-networks (clusters) with one or more clusterheads that are responsible for the intrusion detection in the networks. Figure 3.2 shows an example of hierarchical architecture with one clusterhead. This model differs from host-based architecture in the way that not all nodes need to host IDS agents to reduce the burden of nodes in the network. In this system, clusterheads are responsible to perform the intrusion detections in the network by intercepting all packets that are sent to their clusters and gaining local data from each of their cluster members.

This type of architecture is the most suitable architecture in term of information completeness. Moreover, the idea of reducing the burden of hosting IDS agent in some nodes helps the system to conserve overall energy. However, this has to be paid for the network overhead to form clusters and audit data exchange, not to mention the relatively long detection time as the data exchange is needed to perform the detection. Moreover, malicious nodes that are elected as clusterheads could result to the devastation of the networks.

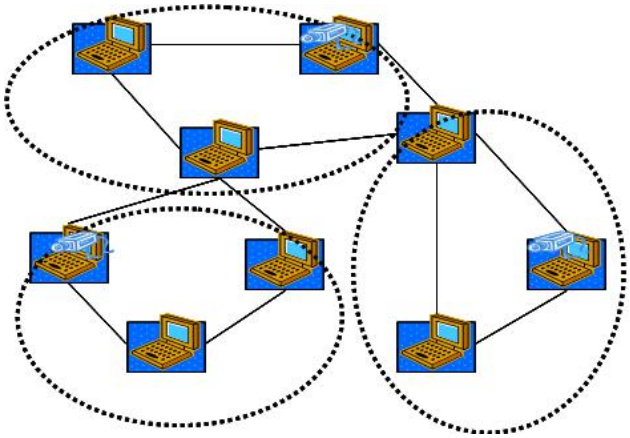


Fig. 3.2 Hierarchical Based Architecture

### Distributed and Cooperative Architecture

The third type of architecture is distributed and cooperative model. Since the nature of ad hoc networks is distributed and requires cooperation of other nodes, Zhang and Lee[9] have proposed that the intrusion detection system in ad hoc networks should also be distributed and cooperative as shown in Figure 3.3. Similar to host-based architecture, Every node participates in intrusion detection and response by having an IDS agent running on them. An IDS agent is responsible for detecting and collecting local events and data to identify possible intrusions, as well as initiating a response independently. However, neighboring IDS agents cooperatively participate in global intrusion detection actions when the evidence is inconclusive through voting mechanism.

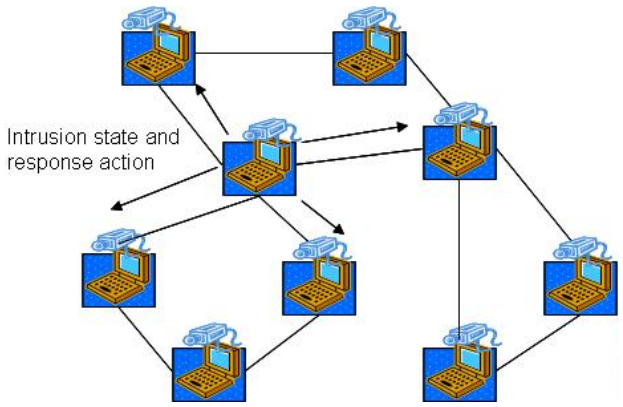



Fig. 3.3 Distributed and cooperative IDS architecture

This model in some extent absorbs some merits from the previous two architectures. Network overhead can be reduced by exchanging data only when it is needed. The lack of completeness of the local audit data can also be compensated by asking the intrusion status in neighboring nodes.

## 3.2 Case Studies

In this section, we briefly explain two related works to be compared with our proposal work later in this thesis. The first method represents host-based architecture and the other represents hierarchical architecture.

#### Cross-Feature Analysis for Detecting Ad Hoc Routing Anomalies

Yi-An Huang et al proposed a technique for automatically constructing anomaly detection models based on host-based architecture.[17] They introduce a new data mining method that performs cross-feature analysis to capture the inter-feature correlation patterns in normal traffic. This anomaly detection system is installed in every node in ad hoc network to provide the node with mechanism to detect intrusion that enter the network by sensing the anomaly in the routing mechanism, e.g. there are more route changes happened in the last five minutes compared to the normal model that has been learned in advance. This calculation is performed with machine learning tool such as RIPPERs and SVMs. They show that the system can sufficiently detect intrusion even with host-based architecture. In this document, we will compare this host-based architecture with our system in term of detection speed. Host-based architecture has the merit of fast detection as the time for data exchange can be eliminated. However, we will show that in some conditions, our proposal will work more efficiently. In addition, we also compare with another hierarchical based architecture to show that within the realm of hierarchical architectures, our method can outperform the others.

#### Connectivity-Based Method

Oleg Kachirski et al proposed an architecture of intrusion detection system based on hierarchical architecture.[15] They proposed an algorithm to select nodes which will host network monitoring and decision making agents in the network based on voting scheme. By doing so, the total resource of the network can be preserved. The clusterhead is selected from all nodes in its neighbor who has the most connections to other nodes, calculated as connectivity index. While this method gives good performance to broaden monitoring in the network, this method can't be as efficient if selected clusterhead is incapable to perform its task due to the lack of resource. This is one of the tasks we want to solve on this research. Later in this thesis, we will compare our proposal method with Connectivity-Based Method in order to show the significance of selecting clusterhead with more power instead of connections.

## 3.3 Summary

In this chapter, we explain several architecture types for intrusion detection in ad hoc network along with each merits and demerits. We also bring up two related works that will be compared with our proposal to evaluate the effectiveness of our work later in this thesis.

# Chapter 4

# MANAGER-BASED INTRUSION DETECTION SYSTEM

As stated before, the main topic of this thesis is to answer the first research question explained in Section 2.3.2, about the development of architecture for intrusion detection system that fits in ad hoc network environment. In this chapter, we discuss our proposal method, a manager-based architecture for intrusion detection system in ad hoc networks. This architecture belongs to hierarchical type architectures that is explained in the previous section.

## 4.1 System Overview

In this section, we discuss the overview of our proposal system. This covers the basic assumption of network model, problems of implementing intrusion detection system in such model, and the objectives that we want to achieve by implementing our work. We also describe the general idea of our architecture in this section.

### 4.1.1 Network Model

We assume the network model in our problem domain as a network with diverse type of components. We believe that this assumption is valid in the real life application that utilizes ad hoc network environment. The network is constructed by different types and performances of mobile nodes, ranging from low performance nodes to high performance nodes. We assume low performance nodes (e.g. sensor and mobile phones) are nodes with special purpose embedded processors that only have tens of megahertz of CPU clock size and short range wireless transceivers that can only transfer data at tens of kilobits per second. In contrast, we assume high performance nodes (e.g. laptop) are more powerful nodes, equipped with more sufficient processing power, e.g. few gigahertz of CPU clock size, sufficient battery power and capability of transferring data at a few megabits per second. In this thesis, we use the term of "weak nodes" for low performance nodes and "strong nodes" for high performance nodes interchangeably.

These nodes are grouped into some smaller sub-networks (clusters) each consists up to tens of nodes. These sub-networks are interconnecting each other with the same routing infrastructure. For simplicity, we assume that the communication between two nodes can be done in two ways (bidirectional), ignoring the fact that transmission range of each node differs from another. Imposing more security to the network, we only allow a cluster to be formed by nodes within one hop links.

### 4.1.2 Problems and Objectives

Among related works explained in previous chapter, there is no method that put the existence of low performance nodes (weak nodes) into considerations. They consider all nodes in the network to have the same capabilities, thus every node has the same responsibilities for the role in the networks. In the case of Connectivity-Based Method,[15)] all nodes have the same chances of being elected as Managers of clusters. As stated previously about our network model, various type of nodes could exist in a network. Selecting a sensor node whose CPU clock is only 10 MHz is apparently not realistic in the term of efficiency. Not only because the weak nodes battery will be drained faster because the heavy load of network intrusion detection agent and the frequent communication they need to perform in order to stabilize the cluster, but they might not be able to detect intrusions, which is the main problem, because of the limited computation power.

The same logic also applies to both host based and distributed and cooperative based method because in these architectures, every weak node also needs to host IDS agent. While it can reduce the performance of the weak node itself, an independent weak node is relatively easier to be compromised by attackers to perform internal attacks that eventually leads to bigger problems in the network.

Another problem in our field of interest is regarding the network overhead due to the cluster generation in hierarchical based architecture. As stated in the Section 3.1, generally hierarchical-based architectures have the merit of having more access to broader and more complete data to increase the detection rate. However, the process to form clusters usually include data transfers and/or other mechanism that lead to more overhead to the network. Therefore, it is essential for the systems that run in ad hoc network environment, that has limitation in resource including the bandwidth, to be efficient in term of producing less data packet for being viable to be implemented in real applications of ad hoc network.

The objectives of our proposal system is to solve the problems aforementioned. More powerful nodes should be in charge to detect the intrusion as well as preventing other nodes from attacks. Moreover, network overhead caused by clustering generation should be as low as possible to reserve the network resources.

### 4.1.3 Manager-Based Architecture

#### General Ideas

We propose a manager-based architecture for intrusion detection system in ad hoc network (Figure 4.1) that belongs to hierarchical architecture model. We divide the nodes that construct the network into two types: Regular Node and Manager. The composition of these nodes are given as 1 Manager for $N$ Regular Nodes (RNs) ($N \geq 0$), together they form a smaller sub-network that is called zone.

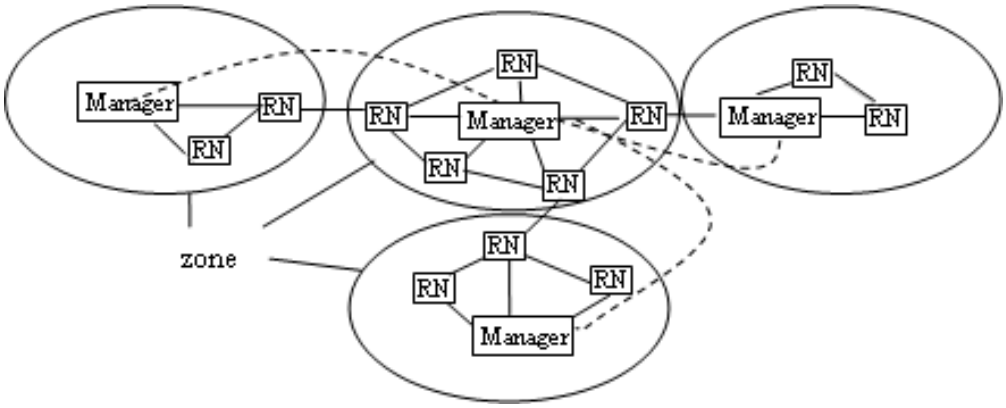



Fig. 4.1 Manager-Based Architecture

RNs function as sensors whose tasks are collecting intrusion data locally specified on the detection algorithm that is utilized in the network. These data can be raw data such as application log files in each node (e.g. the tcpdump data and login history, etc) or crafted data such as the number or percentage of route change occurred in the last 2 minutes, etc. On the other hand, Managers function as the heads of zone to perform the intrusion detection in their zone based on the data collected from Regular Nodes added with its local data. They perform the analysis of the data and send back the result in a form of alert information to every regular nodes in their zones. Since ad hoc network doesn't have any fixed infrastructure, it is difficult to aggregate all intrusion data occurred in the network to one place without cooperation of all nodes. Therefore, in this architecture, all Managers should cooperate to provide the network with more complete data for an accurate and efficient detection. The relation among nodes is best described in Figure 4.2.

The timing of data collection is also decided in specified application or network environment. When Managers are not in a hurry to analyze the local data, Regular Nodes can only send their data periodically. However, when Managers detect an anomaly in the network and need to perform further analysis, they can request the data from Regular Nodes.

In the next subsection, we explain the general requirements in order to create the zone in our system.

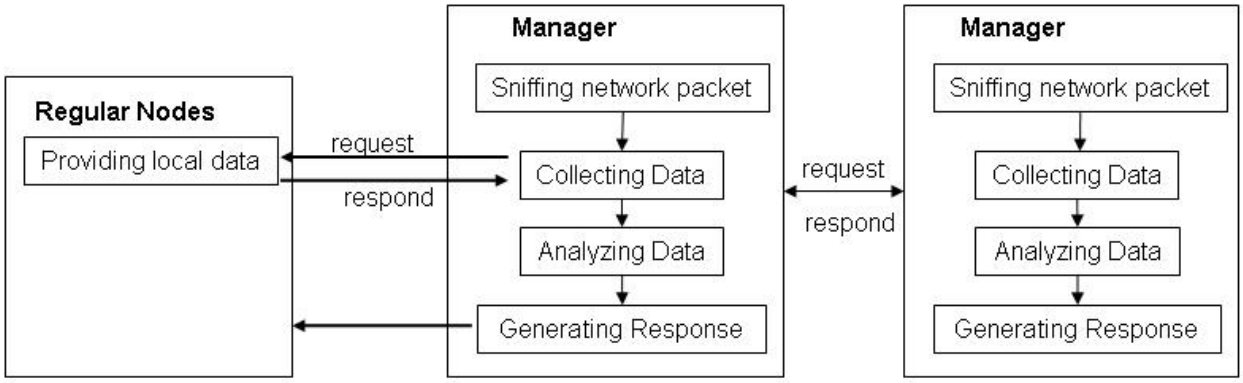



Fig. 4.2 Relationship Among Nodes

### General Requirements

First, it is necessary to make sure that all member of the networks are covered by the system. As stated previously about the network model, we assume a network that needs a high degree of protection, thus all nodes should have their Managers within their 1-hop vicinity. This means that Manager is within the transmission range of one node. By doing so, reliable and fast communication between RN and Manager can be established to provide more accuracy to the system. However, this will result to more frequent change of zone members in the presence of mobility.

The second requirement is an assurance that one node affiliates to only one zone. Zone is a non-overlapping area and it is important to make sure that there is no overlapping zone (i.e. one node affiliates to multiple Managers at a same time) to increase the communication efficiency. Therefore, every node should affiliate to the "best" node in its neighbor. For all nodes in order to have the same definition of quality of a node, there should be parameters to specify that. In this research, we introduce the usage of weight value. Weight value is calculated independently at each node based on specified parameter(s) of that node. This parameter can be one of or compound of computation power, memory size, battery power, and so on that describe performances of a node. It should be a relative value decided for specific quality of a nodes. For example, if we are interested in fast detection time that generally can be achieved by high computation power, we can use CPU clock frequency as the parameter to express the weight of a node. For example, node whose CPU clock frequency more than 1 GHz is given 1000 weight value, hundreds MHz is 100, tens MHz is 10, and so on.

It is also preferable to make sure that Managers are well-distributed throughout the network so that communication efficiency among Managers can be increased as well as broadening the monitored area. One method to do that is by preventing two or more neighboring nodes to become Managers. However, as we will show later in this paper, neighboring Managers should be allowed to prevent "incapable" nodes to become Manager. Incapable node is defined as node whose weight value is below a specific threshold value. Letting this node

to become Manager is not efficient, not only for the nodes, but also to the whole network. Nodes that construct ad hoc network in our problem domain consist from weak nodes and strong nodes, therefore, it is necessary that the algorithms can accommodate this diversity characteristic.

## 4.2 Manager Selection Algorithm

We have developed an algorithm to achieve the objectives and fulfill the requirements explained in the previous section. In this section, we explain about the core part of our proposal architecture system: algorithm of Manager selection. In our system, Managers are the center of zones. Choosing a Manager of a zone is equal to create the zone itself. In the next section, we explain the basic operation of the algorithm, and tell more detail in the section after that.

### 4.2.1 Basic Operation

The Manager Selection Algorithm (MSA) is constructed by 5 functions and 3 control messages. Functions are executed at each node triggered by these specific control messages or messages from other mechanism (e.g. routing mechanism, etc) that tell about disconnection of neighbor nodes. An example of such message is HELLO message of AODV routing protocols.

These specific control messages in this algorithm are transmitted via broadcasting to generate and control members of a zone. These messages and their brief purposes are described as follows:

- **Manager Declaration Message,** $M(X, W_X)$**,** is used by a node $X$ with weight value $W_X$ to declare that it has become a Manager and transmitted periodically by Manager to control the zone.
- **Regular Node Submission Message,** $RN(X, Y)$**,** is used by a node $X$ to declare that it will affiliate to node $Y$'s zone.
- **Incapability Declaration Message,** $INC(X, W_X)$**,** is used by a node $X$ to declare that it is an incapable node, a node whose weight value is too small to become Manager of other nodes.

One message above is sent in order to correspond another type of messages received. Upon receiving one message, a node will execute one of 5 functions below. The functions and the brief explanation about their purposes are as follows :

- **Initialization Function** is performed by a node when it enters a network at the first time, or when it lost the previous Manager.

- **Manager Update Function** is performed by a node to update its role or Manager upon receiving Manager Declaration Message.
- **Regular Node Update Function** is performed by a node to update its role or Manager upon receiving Regular Node Submission Message.
- **Incapable Function** is performed by a node upon receiving Incapability Declaration Message from weak nodes, which is defined as nodes whose weight value is below threshold, to inform its surroundings about its incapability.
- **Link Error Function** is performed by a node after receiving a message about disconnection to neighboring nodes from another mechanism such as HELLO messages of AODV routing protocol.

We will explain these messages and functions in more detail in the next section.

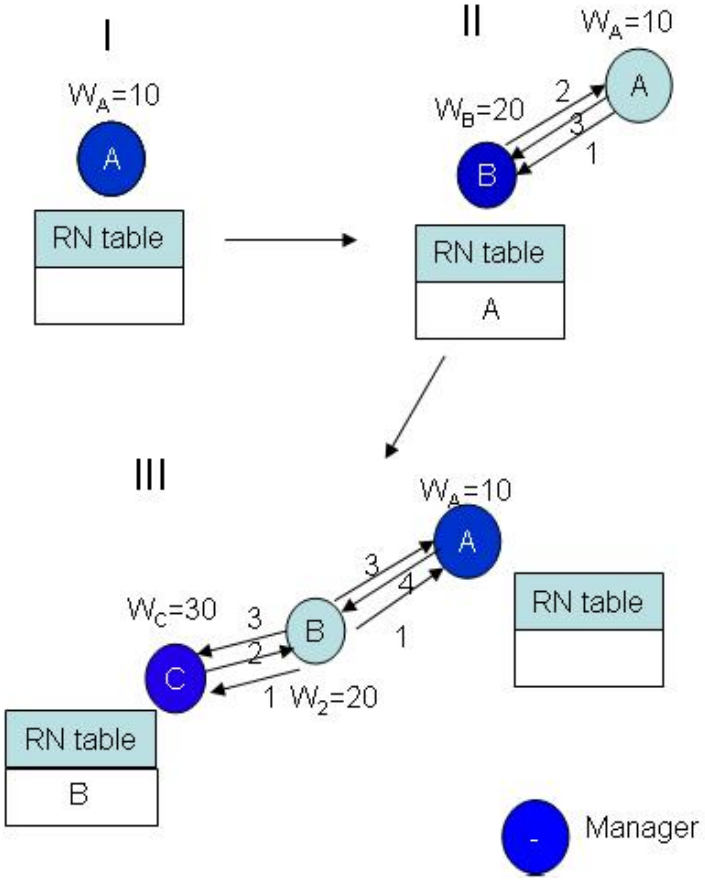



Fig. 4.3 Example Scenario of Manager Selection Algorithm

For further understanding about Manager Selection Algorithm, example scenario of Manager Selection Algorithm shown in Figure 4.3 is used. In this figure, there are 3 steps, starting from I and ending at III. Node is added one by one at every step. The process on how Manager is selected in every step is explained below.

- **Step I**
  Node A enters the network without any other nodes in its neighbor. Thus, after waiting for several times, this node automatically becomes Manager.

- **Step II**
  Node B enters the network within transmission range of node A, hence B receives message $M(A, W_A)$ (indicated with number 1 in Figure 4.3) that is sent by Manager A. Since node B is having bigger weight value (20) than A (10), B declares himself to become Manager by sending $M(B, W_B)$ (indicated with number 2). Upon receiving this message, node A stops being Manager and sends $RN(A, B)$ (packet number 3) to ask node B to become its Manager. B then registers A as its Regular Node.

- **Step III**
  Node C with weight value 30 enters the network within transmission range of node B. After receiving periodical $M(B, W_B)$ (number 1) from B, it sends $M(C, W_C)$ (number 2) because its weight value is larger than B's. Meanwhile, Node A upon receiving $M(B, W_B)$ doesn't do anything. Then, node B sends $RN(B, C)$ (number 3) to become node C's Regular Node. Node A who listens this message, starts becoming Manager again and transmits $M(A, W_A)$ (number 4). Upon receiving this message, node B doesn't do anything since its Manager, node C, has bigger weight value than A.

Based on example above, we can understand that Managers will not be laying side by side. However, in the presence of weak nodes, this algorithm permits adjacent Managers to help those weak nodes. This will be further discussed in the next section.

### 4.2.2 Detailed Descriptions

#### Message Types

As explained in the previous section, this algorithm utilizes specific control messages to select Manager and create zones. The format of messages is shown in Figure 4.4.

#### Manager Declaration Message

We use the format in Figure 4.4 to explain Manager Declaration Message (M Message), $M(X, W_X)$. It has message Type 0. The Message ID is set by sender, incrementing the last value of the same type of packet it sent before. Node ID field is set to the address and Node Weight Value is set to the weight value of the sender. The last field, Manager Node ID, is left empty in this type of message.

#### Regular Node Submission Message

By using the same format, Regular Node Submission Message (RN Message), $RN(X, Y)$, has message type number 1. The Message ID field is set to increment of the last RN Message

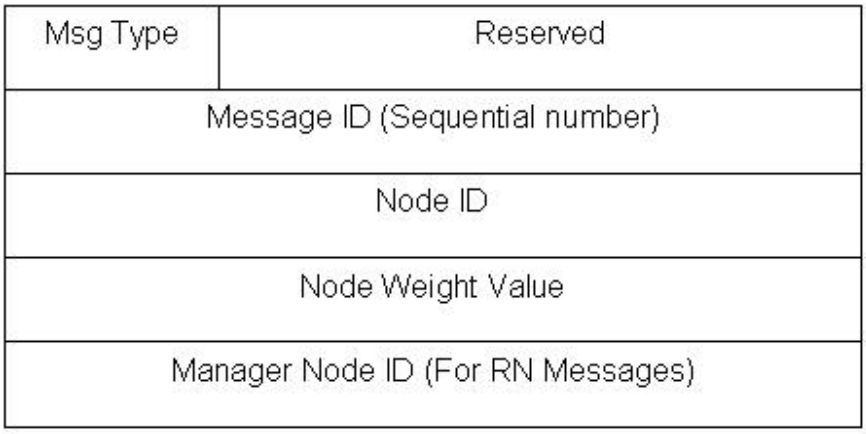



Fig. 4.4 Control Message Format

sent by this node. Node ID and Node Weight Value fields are set to this node's address and the weight value. The last field, Manager Node ID field is set to the address of this node's Manager.

### Incapability Declaration Message

Incapability Declaration Message (INC Message), $INC(X, W_X)$, has message type number 2. The message fields in this message resemble M Message, filling Node ID field and Node Weight Value to its own address and weight value. The last field is also left empty.

### Function Types

Functions in this algorithm are triggered by either one of control messages explained above, or messages from another mechanism such as routing protocol that tells about the condition of neighbor nodes. In this subsection, we will explain in detail every function in Manager Selection Algorithm (MSA). We use flowchart to explain the detail of MSA.

In each figure, there are two parts of algorithm, the part that is only surrounded by dashed line, called MSA I, and the whole part, called MSA II. As stated before, this algorithm can be used to prevent weak nodes to become Manager by allowing adjacent Managers. While in MSA I (dashed line part in the flowchart), there is no special process to prevent incapable node to become Manager of other nodes. However, in MSA II (the whole part of the flowchart), neighboring Managers are allowed to prevent incapable node to become Manager of other nodes.

### Initialization function

The flowchart of Initialization Function is shown in Figure 4.5. When a node $Self$ enters a network or when it can't decide its role in the network, it will wait for $M(X_i, W_{X_i})$ messages during a specified time $T_{wait}$. From all messages it receives during $T_{wait}$, it will choose the

neighbor node $X_k$ with the biggest weight value that is bigger than its own value, then it will broadcast $RN(Self, X_k)$ messages and affiliate to $X_k$'s zone. In the case where multiple nodes have the same weight value, $X_k$ with the smallest $k$, i.e. the node whose messages arrive in first, will be chosen. If there is no Manager whose weight value is bigger than itself, it will broadcast $M(Self, W_{Self})$ message and become Manager. If there is no Manager in its neighbor and its weight value is smaller than pre-defined threshold value $W_{Th}$ that permits one node to become a Manager, it will broadcast $INC(Self, W_{Self})$ message and perform initialization function again after pausing for a specified time $T_{reinit}$.

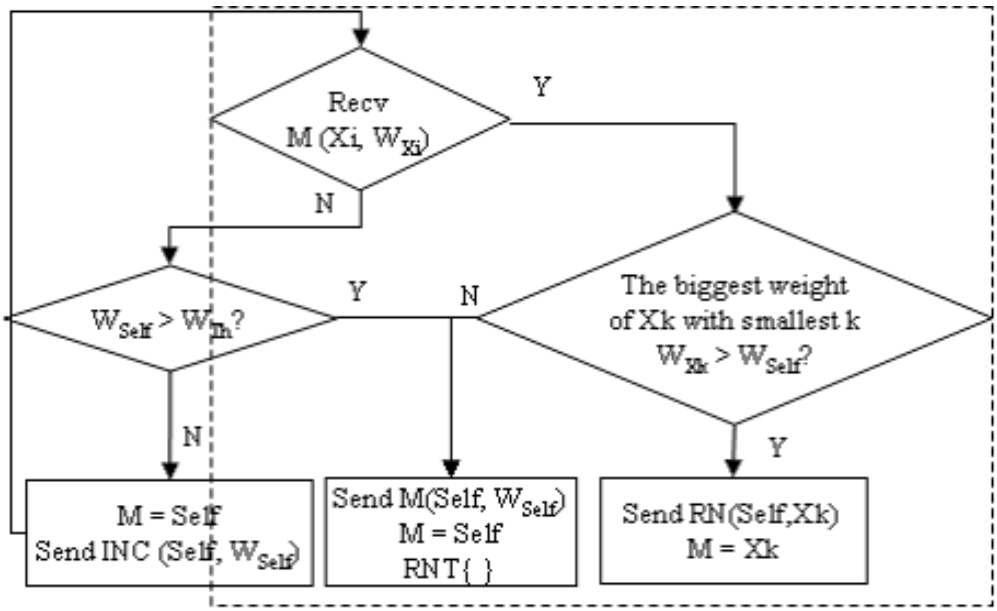



Fig. 4.5 Initialization Function

### Manager Update Function

The flowchart of Manager Update Function is shown in Figure 4.6. This function is executed when a node receives $M(X, W_X)$ message. In MSA I, Node $Self$ simply compares the newly received Manager's weight value with its current Manager's and if new Manager's weight value is bigger, regardless its previous role, $Self$ will join $X$'s zone. However, in MSA II, $Self$ needs to check its current role before joining $X$'s zone because it can't join $X$'s zone if it's currently a Manager of any incapable node(s). The status of Manager of incapable nodes is indicated by $SP\_FLAG = 1$. In other hand, when it receives this $M(X, W_X)$ message from its own RN, $Self$ considers that node $X$ has become Manager of other incapable nodes and releases $X$ from its Regular Node Table (RNT).

### Regular Node Update Function

The flowchart of Regular Node Update Function is shown in Figure 4.7. This function is executed when a node receives $RN(X, Y)$ message. The general principle of this function

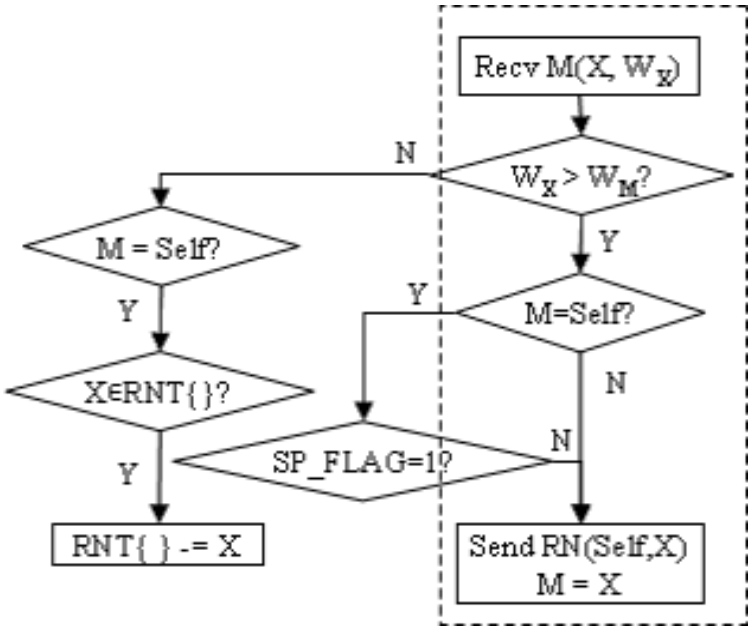



Fig. 4.6 Manager Update Function

is to check the relation between executing node $Self$'s current role with nodes $X$ and $Y$ in the received message. There are three possibilities: 1)If executing node $Self$ is RN of node $X$ then it needs to find a new Manager, 2) If node $Self$ is same as node $Y$, i.e. node $X$ is joining $Self$'s zone, then $Self$ will add $X$ to its RNT, and 3) If $X$ is a RN of $Self$ that leaves to join another Manager $Y$'s zone, then $Self$ will erase $X$ from its RNT.

In a special case of MSA II where $SP\_FLAG = 1$ (i.e. executing node $Self$ is Manager of incapable node(s)), node $Self$ checks if node $X$ resides in its HLT and leaves to join another Manager $Y$'s zone. In the affirmative, $Self$ will erase $X$ from its HLT, and if HLT becomes empty, it will set $SP\_FLAG = 0$ and execute Initialization function to find a new Manager.

### Incapable Function

The flowchart of Incapable Function is shown in Figure 4.8. This function exists only in MSA II, executed by a node upon receiving $INC(X, W_X)$ message. In order to help incapable node, the general principle of Manager's distribution is relaxed and neighboring Managers are allowed. When a Node $Self$ receives $INC(X, W_X)$, first it checks its current role in the network. If it is a RN, it checks its own weight value. If it's bigger than threshold, it will set up $SP\_FLAG = 1$ and insert $X$ in its Helped List Table (HLT). After that, it broadcasts $M(Self, W_{Self})$ message and becomes a Manager.

### Link Error Function

The flowchart of Link Error Function is shown in Figure 4.9. Whenever a node detects link error that is checked periodically, this function will be executed. In this function, the role

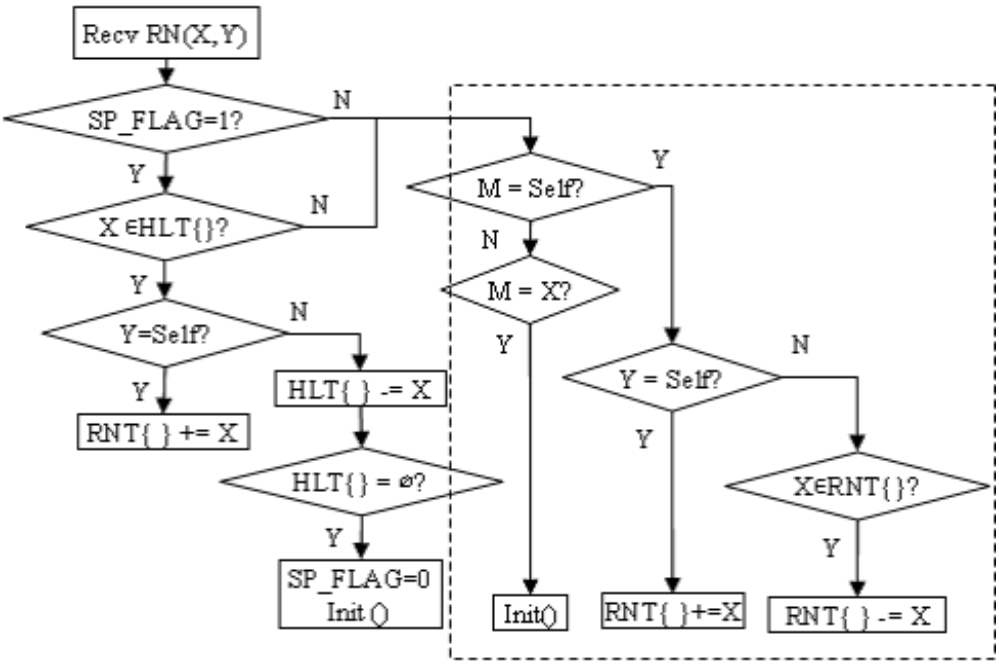



Fig. 4.7 Regular Node Update Function

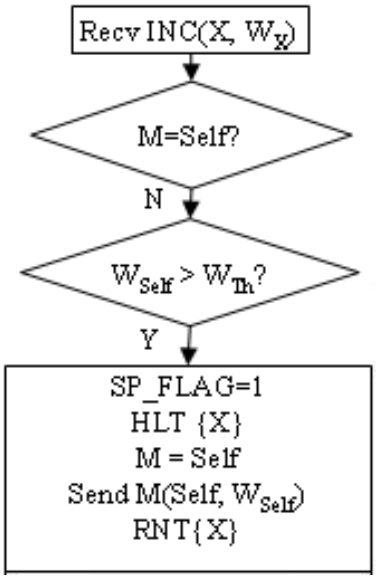



Fig. 4.8 Incapable Function

of missing node and executing node are checked. Basically, there are two possibilities: 1) If missing node $X$ was Manager of executing node $Self$, then $Self$ will execute Initialization function to select a new Manager. 2) If missing node $X$ was Regular Node of executing node $Self$, then $X$ should be erased from $Self$'s Regular Node Table (RNT). Moreover, in MSA II, this is followed by checking $SP\_FLAG$ value and whether $X$ was exist in $Self$'s HLT. If $X$ exist in HLT, it should be erased from there. After that, if HLT entry become empty (i.e. this node is no longer a Manager of any incapable node), then $SP\_FLAG$ will be set to 0 and Initialization function will be executed to select a new Manager.

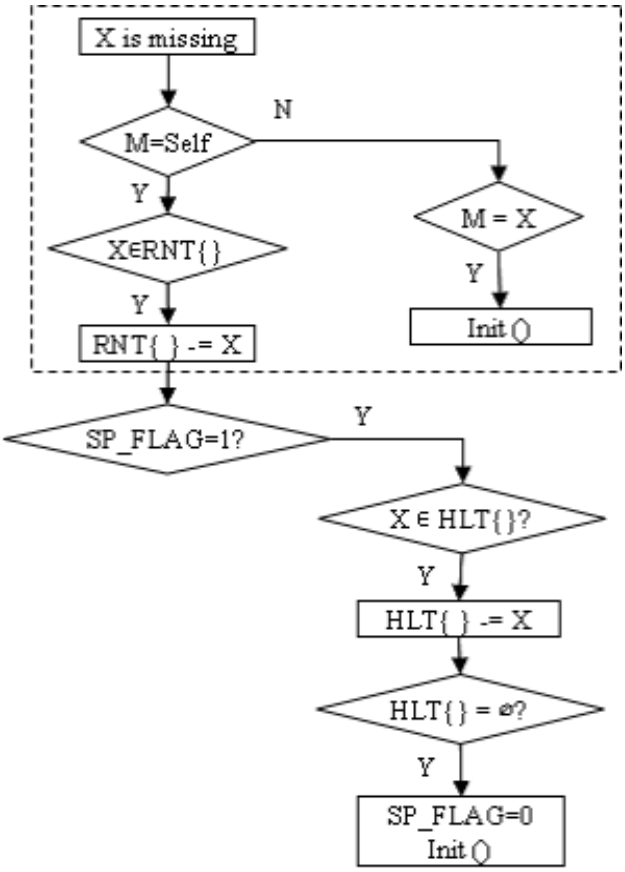



Fig. 4.9 Link Error Function

## 4.3 Summary

In this chapter we presented the detail of our proposal system, a Manager-based architecture. This proposal system is considered to be able to solve the first problem by recruiting as powerful nodes as possible as the nodes in-charge of intrusion detection in the network. Moreover, it's also expected to achieve the second objective, reducing the network overhead, by letting each node manage its own Manager selection, without the need of voting scheme. These two points will be evaluated in the next chapter.

# Chapter 5

# EVALUATION

In order to study the feasibility and effectiveness of our system architecture, we carried out simulation experiments. We conducted two type of simulations: simulation in static network and dynamic network condition. The simulation in the static condition is aimed to study the efficiency of the system architecture in the term of intrusion detection time. Meanwhile, the simulation in the dynamic condition was performed to check the effect of mobility to the network overhead as the result of Manager selection and cluster formation process. In this chapter, we first describe our simulation approach and then present the simulation results respectively.

## 5.1 Simulation in Static Condition

### 5.1.1 Simulation Method

#### Network Modeling

We model the network as an $M \times N$ array with 1 distance-unit between cells, as shown in Figure 5.1. Each ad hoc node is locating in one cell and have bidirectional links with its neighbors. It means that if node A is able to send packet to node B then node B is also able to send packet to node A. Neighbors are defined as the nodes that are locating within transmission range of a node. In Figure 5.1, if transmission range of one node that resides in cell $(i, j)$ is 2, then its neighbors are the nodes that resides in cell $(i-2, j-2)to(i+2, j+2)$ (the gray area of Figure 5.1).

In this simulation, special case of ad hoc network is assumed. All nodes enter the network and communicate with each other in ad hoc mode, a direct communication among nodes without any infrastructure. However, it is assumed that there is no mobility in the network after nodes form the network. Once the clusters are formed and Managers are elected, the network topology remains the same until all nodes leave the network.

Every node is given different ID number and placed in one of the cells randomly. This node has weight value that is also given randomly. 4 type of weight values are utilized in

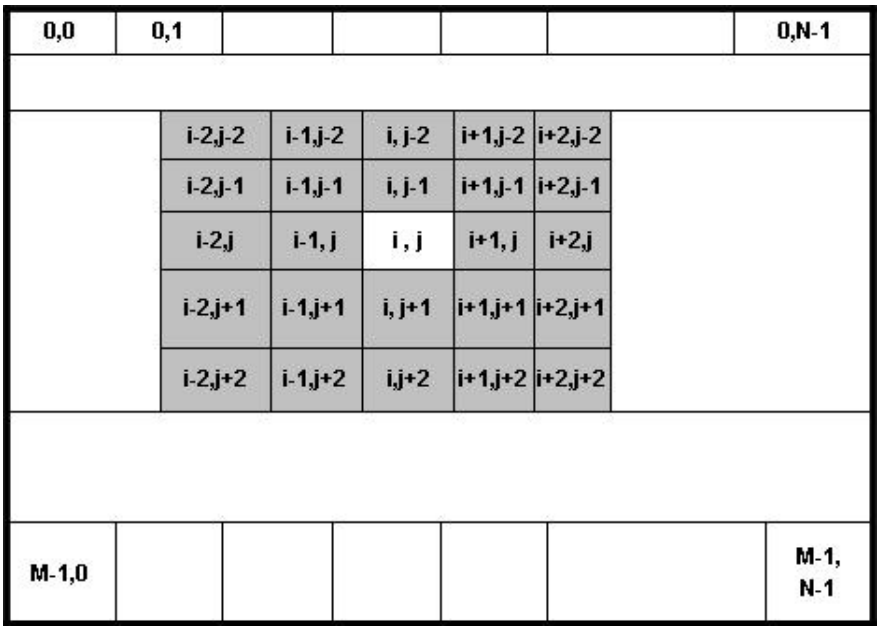



Fig. 5.1 Network Model

this simulation: namely 10, 50, 100, 1000. This weight values represent the computation ability of each nodes. For example, nodes with weight value 1000 can perform 20 times computation of nodes with weight value 50. Although node is given randomly, however, in one simulation, the composition of the number of nodes with specific weight values are kept constant. We use two out of four weight value in one simulation, one weight value represents weak nodes and the other represents strong nodes. In MSA II, weak nodes are incapable of becoming Manager of other nodes.

After nodes enter the network, they run the algorithm to generate the zones. As the target of comparison of clustering algorithm, we choose the Connectivity-Based Method that is explained in Section 3.2 of this thesis. We also pick up one host-based method as the target of comparison in this simulation. The reason we also pick up a host-based method which is seemingly unfair to be compared with our hierarchical based methods due to different methodology is because in this simulation we use one host-based detection algorithm to verify the efficiency of our method.

After Manager selection, simulation enters the intrusion detection phase. In each simulation, one victim node is set, and the detection time of the victim by its Manager is calculated. As the detection algorithm model, we assume cross-feature analysis anomaly detection method,[17] a host-based detection algorithm that is briefly explained in Section 3.2. As explained before, the detection time with this algorithm is linearly proportional to the amount of data. Thus, the trade-off between data processing and data transfer becomes crucial to evaluate this comparison. The next sub-section is dedicated for more detail explanation.

### Evaluation Metrics : Detection Speed

The objective of this evaluation is to determine the impact of proposed architecture on metric, as described in this sub-section.

In this simulation, we consider a metric of detection time $T$ to evaluate this system. Detection time is defined as the time needed by Manager $i$ to collect data from its Regular Nodes (RNs) and analyze them to find the evidence of intrusion.
We express this as follows. (the complete meaning of notations in this section is shown in Table 5.1)

$$T_i = \underbrace{\sum_{j=1}^{N_i} t_j}_{\text{Data Collecting Time}} + \underbrace{\frac{\alpha}{W_i}\{\sum_{j=1}^{N_i}(\beta_j \times t_j) + \gamma_i\}}_{\text{Data Analysis Time}} \tag{5.1}$$

| Notation | Meaning |
|---|---|
| T | Detection time |
| t | Data transfer time from Regular Nodes |
| i | Manager ID |
| j | Regular Node ID |
| $N_i$ | Number of Regular Nodes of Manager $i$ |
| $\alpha$ | Time per byte data by maximum weight node (sec/byte) |
| $\beta$ | Bit-rate (bit/sec) |
| $\gamma$ | Feature data size (byte) |
| E | Expectation value |
| p | The number of all nodes |
| q | The number of Managers |

Table 5.1 Simulation Metrics Notation

The first half of the equation describes data collecting time. Data collecting time of Manager $i$, which has $N_i$ number of Regular Nodes (RNs), is comprised of data transfer time $t(sec)$ from every RN $j$ within its zone. Due to collision if more than 1 node sending data at the same time, data collecting time is bigger than or equal to the total data transfer time of all RNs. However, for the sake of simplicity, we assume that there is a mechanism to queue all RNs to send their data one after another in order to avoid collision, thus data collecting time equals to the total data transfer time of all RNs.

The second half of the equation describes data analysis time, i.e. time needed by Manager to analyze all received data in order to find intrusion evidence. This time is proportional to the amount of data and the capability of one node to process the data. The amount of data is the sum of received data, expressed as product of bit-rate $\beta(bit/sec)$ and data transfer time $t(sec)$, from every RN $j$ and the data $\gamma$ owned by Manager $i$ itself.

Generally, processing capability of one node to process data depends on CPU power of the node. In this paper, we use weight value as the parameter to express this processing capability. Moreover, we introduce a parameter $\alpha(sec/byte)$ that is defined as the time needed by a node with maximum weight value to analyze 1 byte of data. As explained before, weight is a relative value of specific parameter(s) to specify one node's quality. Here, we use CPU clock frequency as the parameter to decide weight value, therefore dividing $\alpha$ with weight value of Manager $i$ yields the processing capability of that Manager.

Based on the metric of detection time, we particularly interested in two values:

• Expectation value $E$ of the detection time. As explained previously in Section 5.1.1, one victim is assumed to be exist in each simulation. This expectation value describes the most probable outcome of one randomly chosen node's detection time in a network with $p$ number of nodes.

To calculate this expectation value of detection time, we use Equation 5.2 where $p$ is the number of all nodes in the network, $q$ is the number of Managers in the network, $N_i$ is the number of Regular Nodes that belong to Manager $i$'s zone, and $T_i$ is detection time of Manager $i$ that is calculated from Equation 5.1.

$$E = \frac{1}{p}\sum_{i=1}^{q}(1 + N_i) \times T_i \tag{5.2}$$

• Maximum value $M$ of the detection time. This value describes the worst probable outcome of one randomly chosen node's detection time in a network, i.e. the detection time when a Manager with minimum weight of value becomes a victim of an attack.

### Parameters Setting

We implemented the network model as specified previously in this section in C computer language. The intrusion detection algorithm itself is not directly installed on the system, as we focus our work on system architecture side, instead, parameters are set to imitate the action on the intrusion detection algorithm previously explained in previous section. The main characteristic of the intrusion detection algorithm is that time needed to analyze the feature data is linearly proportional to the data size. We refer to[17] for more detail explanation.

Feature data size is set to 1250 bytes that can be transferred within 0.01 second with 1 Mbps wireless transmitter. This data is processed by nodes with maximum weight, i.e. 1000, within the same as the data transfer time, giving the data transfer and data processing ratio to be equal. Parameter values are summarized in Table 5.2. Based on these parameters, the calculation of evaluated metrics is performed.

As we are interested in observing the result in different situation, we set some variables for the simulation, shown in Table 5.3.

| Parameter | Value |
|---|---|
| M × N Network Size | 20×20 |
| Number of nodes $p$ | 100 |
| Transmission Range | 2 distance units |
| Data transfer time ($t$) | 0.01 sec |
| Data analysis time ($\alpha$) | $8 \times 10^{-6}$ sec/byte |
| Bitrate ($\beta$) | 1 Mbps |
| Feature data size ($\gamma$) | 1250 byte |

Table 5.2 Simulation Parameters for Evaluation in Static Condition

| Name | Values |
|---|---|
| Number of Weak Nodes | 10, 20, .., 100(%) |
| Weak Nodes Weight | 10, 50, and 100 |

Table 5.3 Simulation Variables for Evaluation in Static Condition

### 5.1.2 Simulation Result

The result of simulation in static condition is discussed in this section. Figure 5.2, 5.3 and 5.4 are showing the expectation value of detection time in case of simulating nodes with weight 1000 and 10 (Figure 5.2), weight 1000 and 20 (Figure 5.3) and weight 1000 and 100 (Figure 5.4). The best result of using this algorithm is when performance ratio of weak nodes and strong nodes is high. In the first figure when the performance ratio is 100, MSA II gives the fastest detection time among other algorithms, even from host-based architecture. Meanwhile, MSA II can accommodate the presence of weak nodes until the ratio achieves 70% to give the same performance as host-based architecture. After that, due to inexistence of mechanism to prevent weak nodes to become Managers, it will deteriorate because more weak nodes are elected as Managers. On the other hand, as the subject of comparison, Connectivity-Based Method, detection time increases linearly as the number of weak nodes increase. We can notice that, when all of the nodes are weak nodes, MSA I converges close to Connectivity-Based Method.

When the ratio of weak nodes and strong nodes performance is low, as shown in Figure 5.3 and 5.4, host-based architecture is giving the fastest detection time in most cases. However, our system outperforms the other hierarchical based architecture.

The result of the simulation, where the chosen victim gives the maximum detection time (worst case), is shown in Figure 5.5, 5.6 and 5.7. This maximum value is achieved when weak nodes become Manager. Moreover, this will be exacerbated by additional nodes that become the member of the weak nodes' Regular Nodes. In Figure 5.5 and 5.6, we notice that at the first ratio where only 10 nodes are weak, proposal method outperforms host-based architecture in detecting intrusions. This happens due to the inexistence of weak

nodes that become Manager. Even MSA that doesn't have any mechanism to prevent weak nodes to become Managers can achieve this result because there are many strong nodes in the network and every weak nodes can affiliate to one of them. However, as the ratio of weak nodes in the network increase, there exist weak nodes that could not be covered by strong node Managers due to random placement in the network. In the comparison with Connectivity-Based Method, proposal methods consistently give the faster detection time.

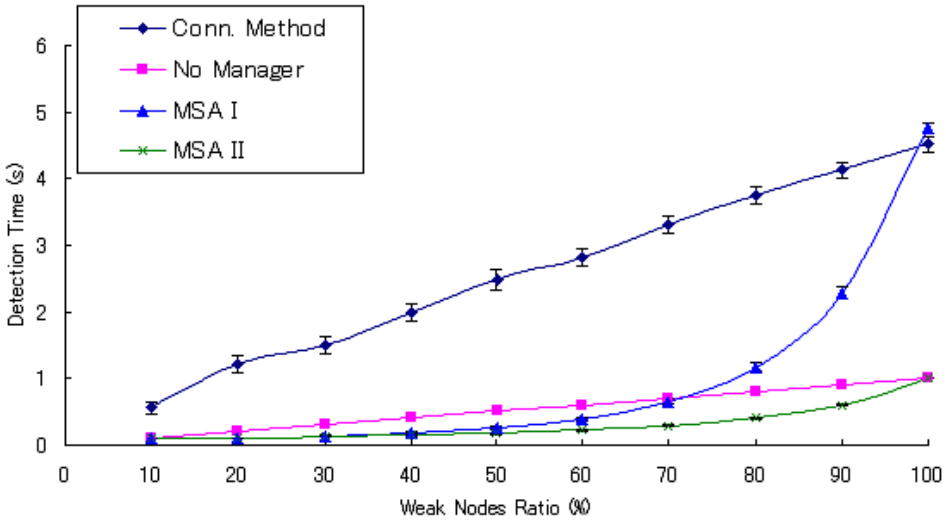



Fig. 5.2 Average detection time versus number of weak nodes in the network when the weight of Weak Nodes : Strong Nodes = 1 : 100

## 5.2 Simulation in Dynamic Condition

### 5.2.1 Simulation Method

#### Network Modeling

The second type of evaluation is performance checking in term of network overhead in a dynamic network environment. The basic model is same as the model used in the previous simulation. In addition to that, we introduce simple mobility scenario into the network (Figure 5.8). As the mobility model, we use random walking model with a constant speed, 1 distance unit per second. For example, a node initially is in cell $(i, j)$, sets its destination to cell $(p, q)$, it then walks one cell per second as pointed by arrow in Figure 5.8. Once it arrives at $(p, q)$, it directly sets to another destination, namely $(r,s)$ and then moves to the new destination. This process is iteratively performed by mobile nodes that are randomly chosen among nodes in the network until the simulation finishes.

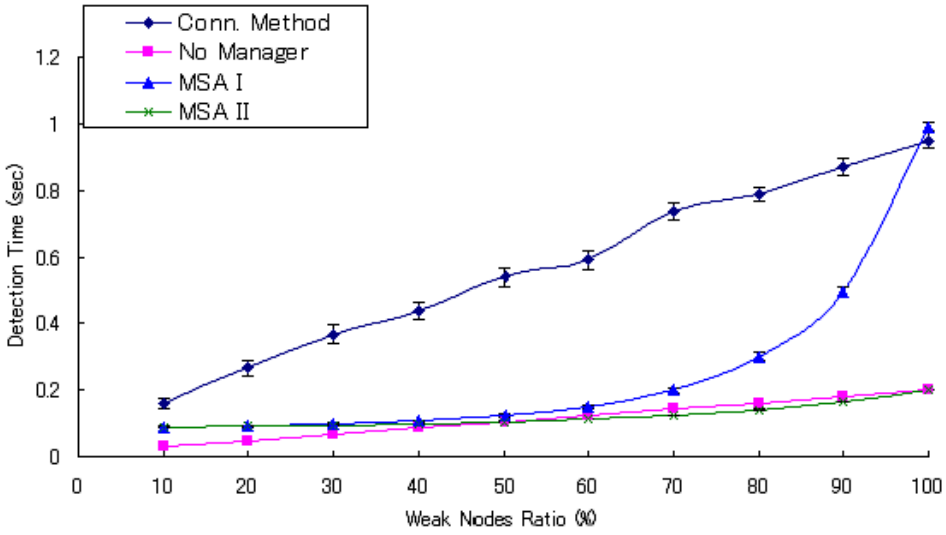



Fig. 5.3 Average detection time versus number of weak nodes in the network when the weight of Weak Nodes : Strong Nodes = 1 : 20

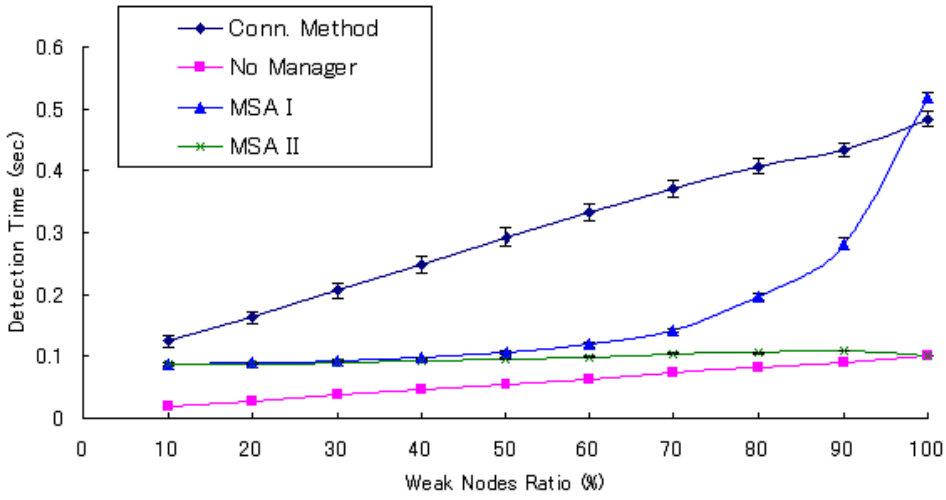



Fig. 5.4 Average detection time versus number of weak nodes in the network when the weight of Weak Nodes : Strong Nodes = 1 : 10

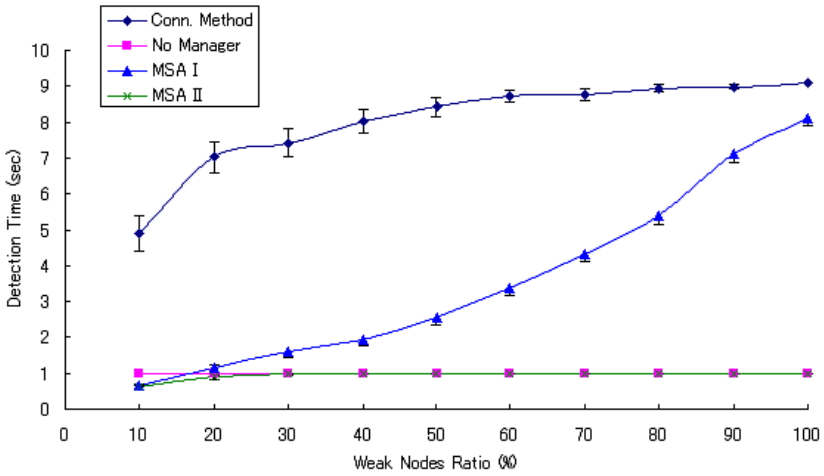



Fig. 5.5 Worst case of detection time in the network when the weight of Weak Nodes : Strong Nodes = 1 : 100

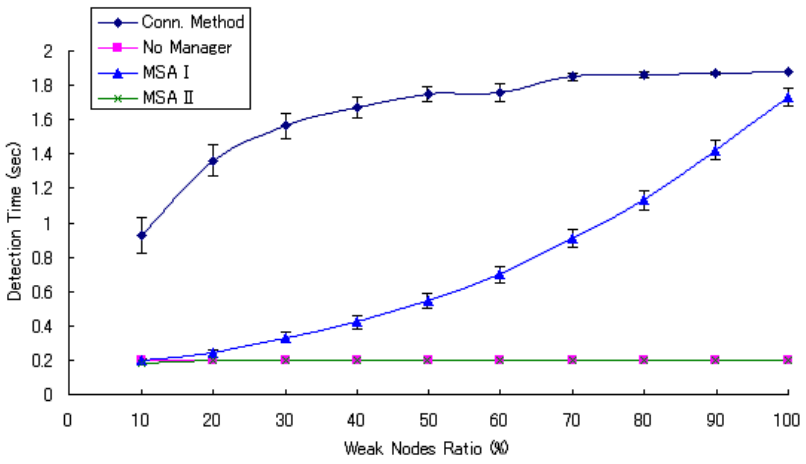



Fig. 5.6 Worst case of detection time in the network when the weight of Weak Nodes : Strong Nodes = 1 : 20

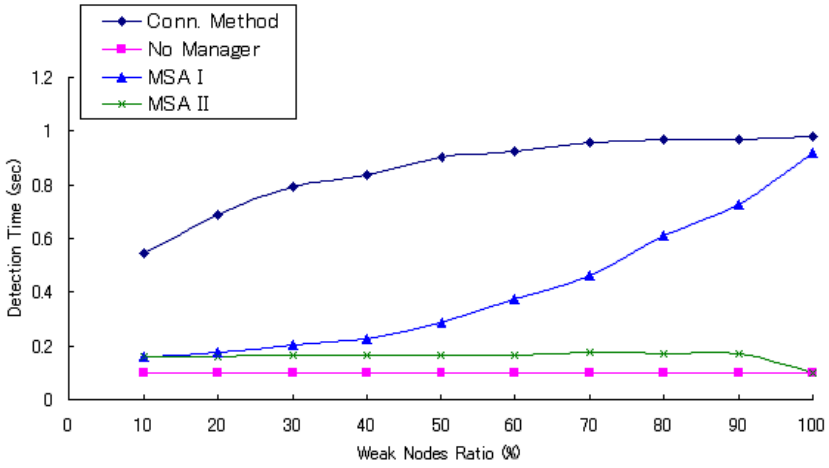



Fig. 5.7 Worst case of detection time in the network when the weight of Weak Nodes : Strong Nodes = 1 : 10

In this network, we also classify the nodes as weak and strong nodes. However since there is no weight-related calculation, there is no variation of weight value. Here, all weak nodes are assumed to have weight value 10, and strong nodes's weight value is 1000.

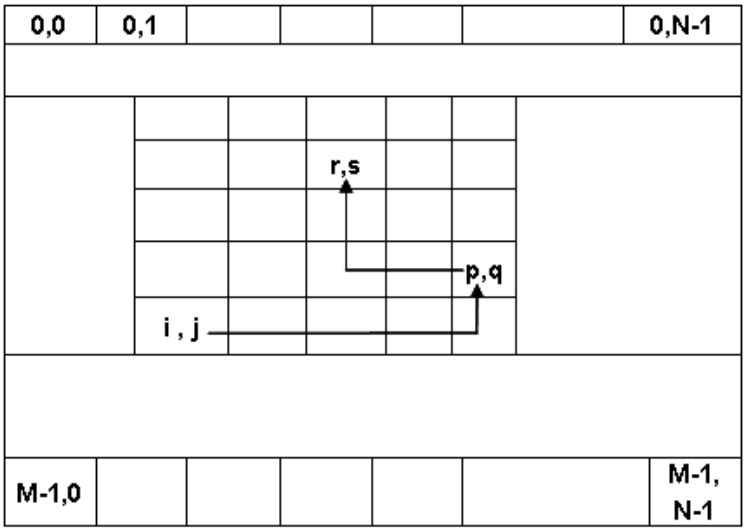



Fig. 5.8 Network Model for Dynamic Condition Evaluation

### Evaluation Metrics : Network Overhead

In this simulation, we consider a metric of network overhead to evaluate this system. In hierarchical architecture-based intrusion detection system, it is necessary to form the cluster before nodes can interact together in detecting intrusions that come to the network. The

cluster generation imposes overhead to the network because of the control packets. In, Connectivity-Based Method, voting scheme is used to select Managers and form the clusters, and in our proposal method, 3 type of control messages are transmitted by every node to select Managers. These messages are the source of overhead to the network. In this simulation, the increase of packet number needed to form cluster when nodes move is counted and compared to show the level of efficiency of proposal methods.

**Parameter Settings**

Basically, simulation parameters are also same as the simulation in static condition. However, minor changes such as transmission range parameter is moved from parameter table (Table 5.4) to simulation variable (Table 5.5) because we can also evaluate the effect of network density toward the network overhead. Besides network density, the effect of the number of mobile nodes and the number of weak nodes are also considered in this evaluation.

| Parameter | Value |
|---|---|
| M × N Network Size | 20×20 |
| Number of nodes $p$ | 100 |
| Bitrate ($\beta$) | 1 Mbps |
| Move speed | 1 distance-unit/sec |

Table 5.4 Simulation Parameters for Evaluation in Dynamic Condition

| Name | Values |
|---|---|
| Number of Weak Nodes | 10, 20, .., 100 |
| Transmission Range | 2, 4, 6, and 8 (distance-unit/sec) |
| Number of Mobile Nodes | 10, 20, ..., 100 |

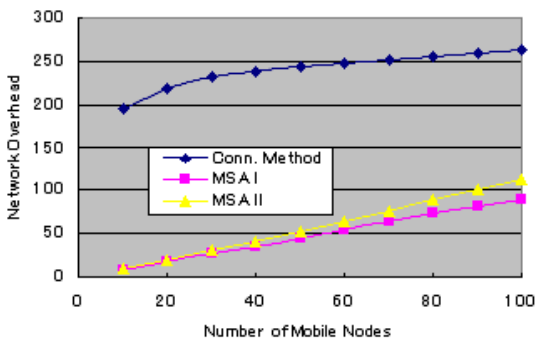
Table 5.5 Simulation Variables for Evaluation in Dynamic Condition

### 5.2.2 Simulation Result

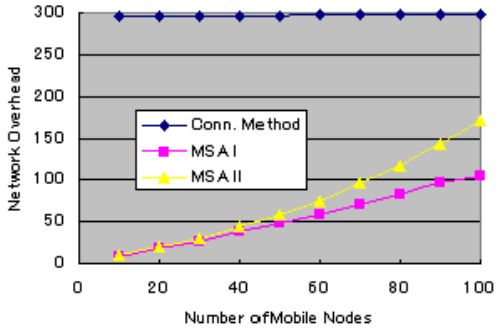
The first simulation is to evaluate the effect of transmission range and the number of mobile nodes toward the network overhead. In this simulation, the variable weak nodes number is fixed to 50 and two variables, namely transmission range and mobile nodes number, is evaluated. The result is shown in Figure 5.9.

In all cases, proposal method system gives the better result, in term of low network overheads. As stated before, network overhead here is the increase packet number as nodes move. When transmission range is 2, we can see that the increasing of mobile nodes number increase the overhead in all architectures. However, after transmission range increase to 4,

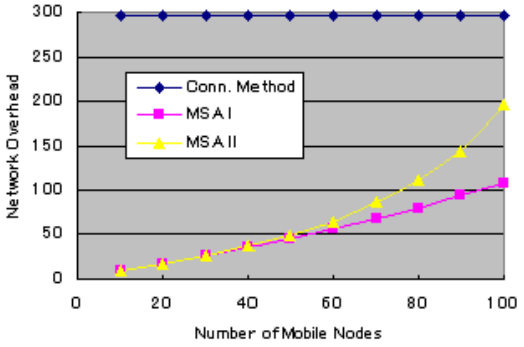
the overhead in Connectivity Based Method seems to be static. This is because it has reached the maximum packet number it can transmit in one Manager selection, which is 3×N with N is the number of all nodes. For more detail about Connectivity-Based Method,[15] is referred. The increase of transmission range result to more nodes affected when nodes move, thus the network overhead also increase. From the graph we can see that the increase of mobile nodes number gives bigger effect to MSA II rather than MSA I. Number of weak nodes in this simulation is set to 50, making the movement of a node make a bigger probability for a node to lose its Manager. When a weak node loses its Manager, it will perform Initialization function and transmit Incapability Declaration Message. This message increase the overhead of the network.



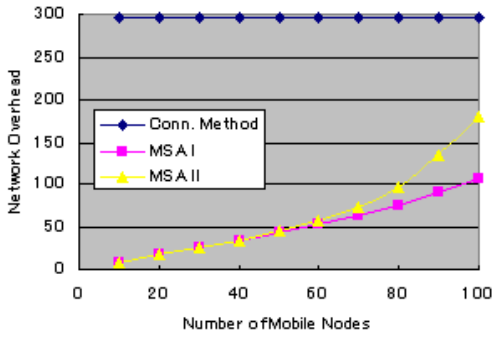

(a) Transmission Range = 2




(b) Transmission Range = 4




(c) Transmission Range = 6

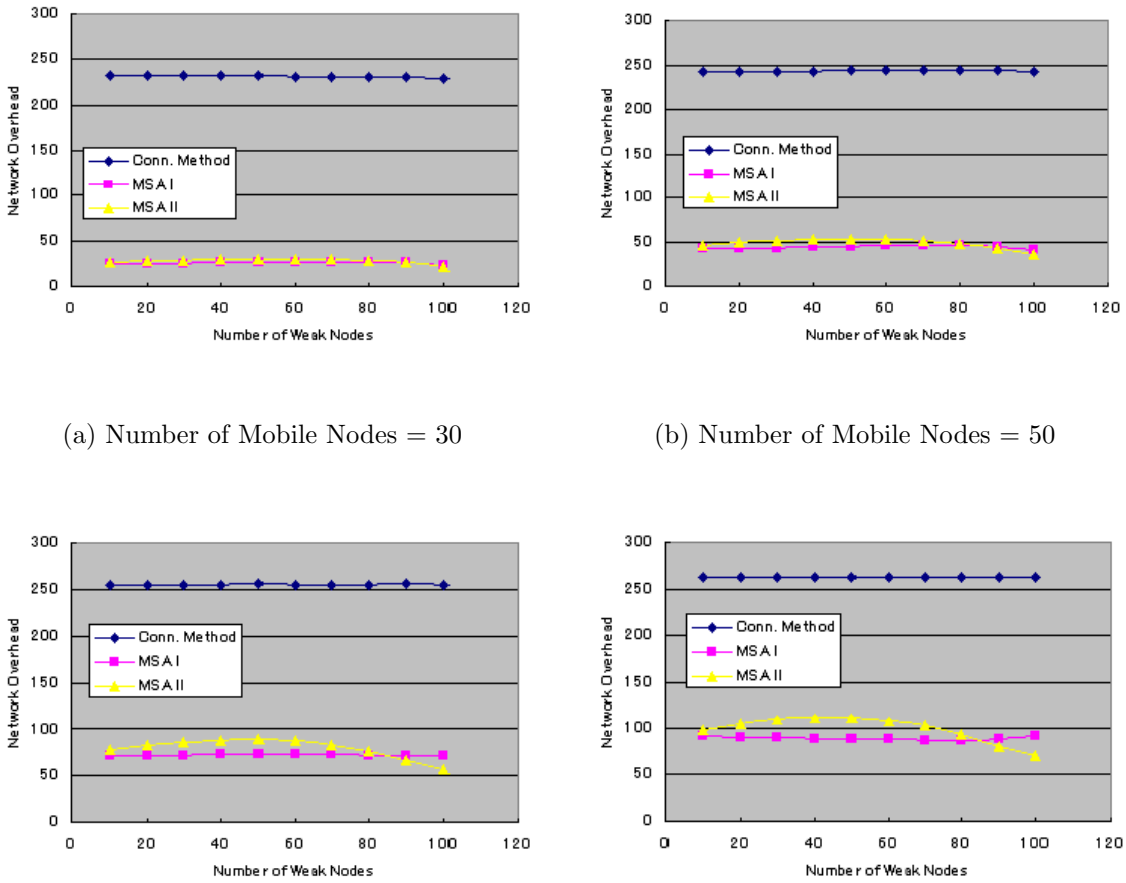



(d) Transmission Range = 8

Fig. 5.9 Number of Weak Nodes = 50

The second simulation is to evaluate the effect of the number of mobile nodes and the number of weak nodes to the network overhead. In this simulation, the variable transmission

range is fixed to 2. The result is shown in Figure 5.10. From this graph, and also from the previous graph (Figure 5.9), we can see that mobile nodes number increase the network overhead for all type of architecture system. However, we can see in all graphs of Figure 5.10(a) that the number of weak nodes only affect MSA II.




(a) Number of Mobile Nodes = 30

(b) Number of Mobile Nodes = 50

(c) Number of Mobile Nodes = 80

(d) Number of Mobile Nodes = 100

Fig. 5.10 Transmission Range = 2

## 5.3 Discussion

Both of simulations are conducted to evaluate the proposal system in term of intrusion detection time and effect to network overhead, and compare our work with previous related works. As stated before in previous chapter, we believe that hierarchical based architecture is more suitable in ad hoc network where various type of nodes exist, because besides more data can be collected from many nodes to increase the detection rate, weak nodes can also be helped from hosting intrusion detection agent that could be computationally expensive for such weak nodes.

From the first simulation in Section 5.1, we notice that proposal architecture gives faster detection time, even from the host-based method in some conditions. While we can claim that our result is valid for comparison with host-based architecture, because host-based method couldn't collect more data, which could lead to higher detection rate, than hierarchical architecture, there is possibility that Connectivity-Based Method collects more data than proposal system. Therefore, we need to check the number of Regular Nodes per Manager to verify the validity of detection result. The data is shown in Figure 5.11 and 5.12.

Figure 5.11 shows the number of Regular Nodes per Manager in the network that are used for simulation in Section 5.1. Figure 5.12 shows the percentage of Managers in the same network. Having this result as reference, we can be sure about the validity of simulation result in Figure 5.2, 5.3, and 5.4. Moreover, we can conclude from this graph that when all nodes are weak nodes, MSA I algorithm is giving the same performance, in term detection speed, with Connectivity-Based Method.

From this graph we can conclude when all nodes are weak, MSA II gives the same result with host-based method. As the future work, we could make a mix system of MSA I and MSA II due to this characteristic, to prevent the architecture to be fully host-based system.

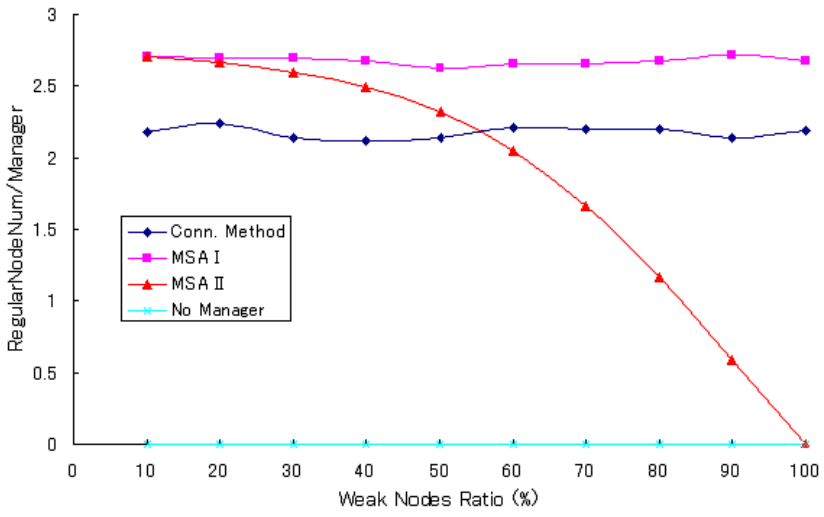



Fig. 5.11 The Number of Regular Nodes per Manager in The Network

In the simulation in Section 5.2, we can conclude that proposal architecture transmit less packet to select Managers. This due to the management of Manager selection is fully given to every node. Meanwhile, in Connectivity-Based Method, when a node moves to a new place, it will initiate a voting request so the voting scheme can be performed immediately.

This feature gives the merit of fast cluster generation in Connectivity-Based Method. Meanwhile, in proposal method, when a node come to a new place, it will wait, for a definite time, a Manager Declaration Packet from existing Managers.

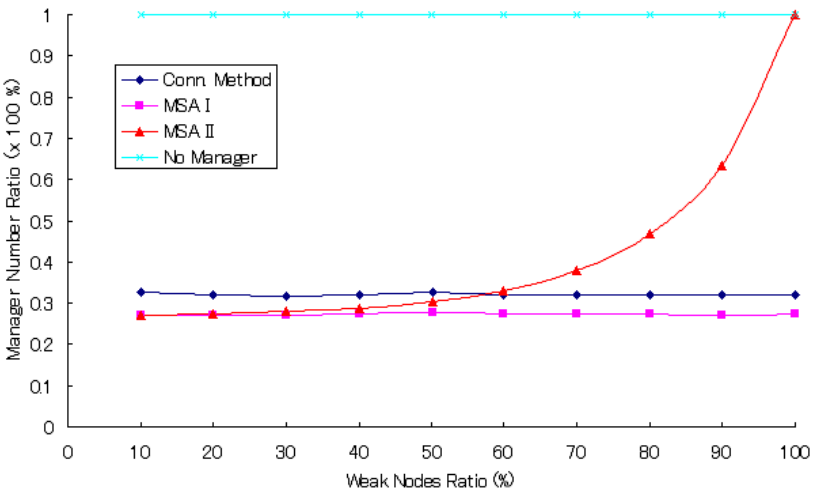



Fig. 5.12 The Number of Manager Ratio in The Network

## 5.4 Summary

In this chapter, we explained about our evaluation for our Manager-based architecture, along with comparison with other existing methods. We can conclude that our proposal system is efficient in term of speed detection and low network overhead compared to other methods.

# Chapter 6

# CONCLUSION AND FUTURE WORKS

## 6.1 Conclusion

In this thesis, we proposed an architecture of intrusion detection system to realize the fast detection time and low network overhead in ad hoc network environment. Since ad hoc networks are usually constructed by various type of mobile devices, asking for "better" nodes to perform intrusion detection is instinctively a good idea. We proposed two type of algorithm to select Manager, that is responsible for intrusion detection in the network, based on weight value, a parameter to specify one node's quality. The performance of proposed system has been verified in a simulation-based experiment and also compared with other related works. The result shows that our method gives the better performance under several conditions.

## 6.2 Future Work

One of the challenging part for this work to be applicable in real environment is that we need to guarantee that the IDS cannot be compromised. Malicious nodes can falsify its weight value to be elected as Manager that eventually results to the inefficiency of the system. Utilizing reputation mechanism, such as the works in[12] and,[11] is one feasible solution in term of setting additional constraints for nodes to be selected as Manager. This approach in some extent can prevent malicious nodes to become Manager. Another feasible solution is utilizing a mechanism to ease the detection of malicious Manager. Thus, logical malicious nodes do not want to be Manager. The next step of our work is to investigate the effective way to increase the security of the proposed system.

# Bibliography


1) Frank Stajano and Ross J. Anderson, "The Resurrecting Duckling: Security Issues for Ad Hoc Wireless Networks," Security Protocols, 7th International Workshop Proceedings, pp. 172-194, 1999.

2) Y. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure On-Demand Routing Protocol for Ad hoc Networks," Proceedings of the 8th Annual International Conference on Mobile Computing and Networking (MobiCom'02), pp. 12-23, September 2002.

3) H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang, "URSA: Ubiquitous and Robust Access Control for Mobile Ad-Hoc Networks," IEEE/ACM Transactions on Networking, December 2004.

4) G. O'Shea and M. Roe, "Child-proof Authentication for MIPV6 (CAM)", ACM Computer Communication Review, April 2001.

5) Zhou, L. and Haas Z., " Securing Ad Hoc Networks, " IEEE Network Magazine, vol. 13, no. 6, November/December 1999.

6) M. G. Zapata, "Secure Ad Hoc On-Demand Distance Vector (SAODV) Routing," ACM Mobile Computing and Communication Review (MC2R), Vol. 6, No. 3, pp. 106-107, July 2002.

7) H. Debar, M. Dacier, A. Wespi, "A Revised Taxonomy for Intrusion-Detection Systems, " ANNALES DES TELECOMMUNICATIONS, Vol. 55, Part 7/8, pp. 361-378, 2000.

8) P. Brutch, C. Ko, "Challenges in intrusion detection for wireless ad-hoc networks," Proceedings in 2003 Symposium on Applications and the Internet Workshops, pp. 368-373, 27-31 January 2003.

9) Y. Zhang, W. Lee, "Intrusion Detection in Wireless Ad Hoc Networks," 6th Int'l. Conf. Mobile Comp. and Net., Aug. 2000, pp. 275-283.

10) S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom'00), pp. 255-265, August 2000.

11) S. Buchegger and J. Le Boudec, "Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes - Fairness In Dynamic Ad-hoc NeTworks)," Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02), pp. 226-336,June 2002.

12) P. Michiardi and R. Molva, "Core: A Collaborative Reputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks," Communication and Multimedia Security Conference (CMS'02), September 2002.

13) P. Albers, O. Camp, J. Percher, B. Jouga, L. M, and R. Puttini, "Security in Ad Hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches," Proceedings of the 1st International Workshop on Wireless Information Systems (WIS-2002), pp. 1-12, April 2002.

14) A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion Detection in Wireless Ad Hoc Networks," IEEE Wireless Communications, Vol. 11, Issue 1, pp. 48-60, February 2004.

15) O. Kachirski and R. Guha, "Effective Intrusion Detection Using Multiple Sensors in Wireless Ad Hoc Networks," Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03),pp. 57-64.1, January 2003.

16) D. Sterne, et al., "A General Cooperative Intrusion Detection Architectures for MANETs", Proceedings on 3rd IEEE International Workshop on Information Assurance, pp. 57-70, March 2005.

17) Y. Huang, W. Fan, W. Lee, and P. S. Yu., "Cross-feature Analysis for Detecting Ad-hoc Routing Anomalies," Proceedings of the 23rd International Conference on Distributed Computing Systems, 99, May 2003.