ABSTRACT

CLUSTER-BASED WIRELESS MOBILE AD-HOC NETWORK SECURITY

by
Mandur Bajra Bajracharya

A wireless ad-hoc network is a cooperative communication network deployed with a specific purpose. It is characterized by dynamically changing topology with no fixed infrastructure which exhibits a perfectly vulnerable state for a number of different kinds of attacks.

This thesis proposes a novel technique of achieving security in wireless mobile ad hoc networks by integrating a simple clustering and a voting mechanism together. The purpose of forming a cluster is to facilitate security in the network and the voting is performed to support the decision making process in the cluster. A procedure of forming small homogenous clusters throughout the network has been proposed and used to suggest different ways of intrusion detection and response. An effort has been made to show how these nodes can be dynamically grouped into clusters by using only the MAC layer.

The fundamental premise of the proposed security scheme is based on the policy "do not let neighbors cheat". Other key issues addressed in the thesis are managing the cluster, assessing performance of cluster members by performing comparative analysis of recorded statistics with the pre-existing values, and voting to decide the action required to be taken to keep the network safe.

Cluster based security is obtained in wireless ad hoc networks by employing a distributed Intrusion Detection System (IDS). Each of the mobile nodes 'listens' to the

traffic constantly and collects local data to form a table. Since no cluster heads are maintained in a cluster, whenever a node detects an anomaly in the network, it consults its cluster members and derives a decisive response. Consultation is done primarily by a voting process. Each of the cluster members vote on the basis of statistics they have collected in their local statistics table. Majority voting is accepted and action is taken under the predefined law. Similarly, when a node experiences pressure because of network traffic or other network activities, it asks for assistance from its cluster members, and assistance is provided as per directives given to each node.

OPNET, a network simulation tool, has been used to simulate the formation of a cluster and measure the time required under various conditions.

CLUSTER-BASED WIRELESS MOBILE AD-HOC NETWORK SECURITY

by
Mandur Bajra Bajracharya

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Telecommunications

Department of Electrical and Computer Engineering

May 2005

APPROVAL PAGE

CLUSTER-BASED WIRELESS MOBILE AD HOC NETWORK SECURITY

Mandur Bajra Bajracharya

Dr. Nirwan Ansari, Thesis Advisor — Date
Associate Chair - Graduate Studies
Professor of Electrical and Computer Engineering, NJIT

Dr. Swades De, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

Dr. Roberto Rojas-Cessa, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Mandur Bajra Bajracharya

Degree: Master of Science

Date: May 2005

Undergraduate and Graduate Education:

- Master of Science in Telecommunications,
  New Jersey Institute of Technology, Newark, NJ, 2005

- Bachelor of Engineering in Electronics and Communication Engineering,
  Dr. Ambedkar Institute of Technology, Bangalore, India, 2001

Major: Telecommunications

To my mother for her unconditional love,

To my father for his guidance,

To my brother and my friends for their encouragement and support.

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to Dr. Nirwan Ansari, who has not only served as my research supervisor, providing valuable and countless resources, insight, and intuition, but has also constantly given me support, encouragement, and reassurance. Special thanks are given to Dr. Swades De and Dr. Roberto Rojas-Cessa for actively participating in my committee.

Many of my senior colleagues in the Advanced Networking Laboratory are deserving of recognition for their support. I would like to especially thank my senior colleagues Mr. Li Zhu and Ms. Yuanqiu Luo for their immense help during practical difficulties. Also, I would like to give thanks to Ms. Brenda Walker and the entire staff of the Department of Electrical and Computer Engineering at NJIT. Last, but not the least, I would like to thank Mr. Rohan Bafna, Mr. Sarabjit Khanuja and Ms. Savita Raina– all your understanding and support saw me through difficult times.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES
(Continued)

CHAPTER 1

INTRODUCTION

It is easier to deploy wireless communication network than conventional wired networks. They provide seamless connectivity giving unrestricted worker mobility within the coverage area. With the increasing data transfer rate ever offered, there could be hardly anyone who would be willing to be denied the comfort and ease delivered by the wireless communication technology.

As the industry standards are maturing and the availability of lightweight wireless networking hardware is growing, wireless local area networks are being rapidly deployed in industrial, commercial, and home networks. As different standards come into effect, compatibility between different vendors is increasing, making the wireless devices more affordable and readily available in the market [14]. As a result, use of wireless communications is increasingly becoming pervasive in our daily lives.

Cellular networks are the earliest form of Wireless networks in which users are connected using access points (base stations), and the backbone network. A user can roam around through hand-offs between different access points. This kind of practice of forming a wireless environment has been widely deployed and commercially used all around the globe. The limitation of cellular network is its fixed topology of network at its access point level, limiting the mobility of mobile nodes within the coverage area of its access points. An ad hoc network overcomes this limitation and offers each mobile node to act as a router, thus changing the topology rapidly and unpredictably. Besides, mobile

nodes in such systems could be autonomous systems deployed on an ad hoc basis or could be part of the global network connected to the larger Internet.

Mobile Ad Hoc Networks (MANETs) are more vulnerable to attacks because of their lack of a fixed infrastructure over the wireless environment. Owing to the broadcasting nature of MANETs, it is extremely easy for any active node to access the network. Any mobile node within the radio range of another node can always 'listen' to what is being broadcasted, thus violating the privacy of the broadcasting node. It is also easy for any malicious node to broadcast false information and disturb the operation of the network. In view of this context, unlike wired networks, where monitoring and ensuring of data integrity are done at higher layers of the network protocols (assuming the lower layers are protected by wires for transferring data [4]), MANETs demand security to be addressed at the datalink layer.

Another property of MANETs posing challenging threats to its security is its constantly changing topology. The nodes in MANET are expected to join on the fly as they move in and out of the network. There are no central certification authorities who could perform an administrative task for certification of any of the nodes. Whenever a new node joins the network, there is no way of knowing whether the node is malicious or not. Again, unlike in the wired network, where an intruder could be a host inside or outside the network, in MANETs, an intruder is necessarily a part of the network infrastructure. Thus, it is even easier for an intruder to manipulate the message, and the routing.

It is difficult to identify a malicious node in a network as long as it does not perform any undesirable activity. It is only when these nodes start showing aberrant

behavior that a malicious node can be detected, and immediate measures can be taken to secure the network. There are basically three different ways of detecting malicious node in a network as categorized by [15]. They are classified as anomaly detection, signature detection and misuse detection. However, irrespective of how a malicious node is detected, limitation of not being able to identify a malicious node in the very beginning causes most MANETs operate in the mode "trust no peer" [1].

There has been a tremendous amount of research, in the field of security in ad hoc networks. This includes developing a secure routing protocol, developing different cryptographic methods to maintain the integrity of data, and developing different policies for security. Yet, there still exist a number of areas which require further attention.

1.1 Research Objectives

This thesis aims at finding the solutions for various network security issues in mobile ad hoc networks.

The first objective of this work is to explore the feasibility of forming a cluster network to defend against security attacks. Clustered architecture has been used for routing [16, 25], frequency and code distribution [16, 21], and to provide an individual node a scaled down view of a network [16, 28]. Different from the above applications, this research aims to form clusters to perform 'neighborhood based watch' [1], in which each cluster member monitors the activity of the other cluster members. The second objective of this thesis is to explore and propose possible solutions for different kinds of attacks employing homogenous clusters.

1.2 Thesis Organization

This thesis is organized as follows. Chapter 2 discusses various attributes of ad hoc network securities and taxonomy of attacks. It also provides a brief overview on a few research works related to this thesis. In Chapter 3, detailed explanation of the proposed 'homogenous cluster' is presented. It also introduces the concepts and policies that are used to form a cluster. Chapter 4 presents the explanation of information maintained by each node and the intrusion recovery concept in details. It starts with the explanation of information maintained by each node and finally encompasses the intrusion recovery concept in details. Four different cases of attacks and possible solutions for them are discussed to illustrate the possible use of homogenous cluster-based architecture for intrusion detection and response. Chapter 5 presents the simulation model adopted in this thesis. Chapter 6 presents the simulation results for different scenarios. Chapter 7 concludes this thesis and discusses the future research directions.

CHAPTER 2

BACKGROUND AND PRIOR WORK

This chapter briefly describes the existing works in the field of security in ad hoc networks. Section 2.1 describes different security attributes, and threats related to wireless networks are discussed in Section 2.2. Section 2.3 surveys three research works, which provide the useful technical background leading to this thesis.

2.1 Security Attributes

The following attributes help to classify different security needs in ad hoc networks [11].

Authentication: This term describes the need of recognizing users with the right identity for data transmission. It is required to make sure that the network is not deceived by malicious terminals, which provide false information or intercept data from genuine users.

Availability: This term describes the need of presence of ad hoc terminals when they are desired by other fellow terminals. It is required to make sure that the network acts in a cooperative fashion to achieve the total functionality.

Confidentiality: This term describes the need of confining information among end users. It is important to make sure that information about the network is not exposed to malicious terminals. This goal is usually achieved by a cryptographic technique.

By using these attributes, each node in an ad hoc network needs to possess the following characteristics for complete security [11, 19].

Byzantine robustness: Being able to perform properly even when the network is under attack.

Certainty of discovery: Making sure that the presence of a node is correctly detected.

Integrity: Maintaining certainty that the data transmitted/received are not tampered.

Isolation: Being able to isolate malicious nodes from good ones.

Lightweight computations: Having tasks with low computational complexity to perform any operation.

Location privacy: Making sure that locations of nodes are not exposed to malicious nodes.

Non-Repudiation: Making sure that nodes in the network do not deny services requested to them.

Self-stabilization: Being able to recover from an attack after a certain amount of time.

2.2 Threats and Attacks

In wireless networks, an attack can be targeted at either the performance of the network or the data in the packets. If an attack is made to the network, it can bring down the performance of the whole network by damaging its throughput and efficiency. If an attack is made to the data packets, it can harm an individual node and its performance. Moreover, it is easier to launch an attack on wireless ad hoc networks because they are deployed in public places where threats of eavesdropping are prominent. Attacks are categorized as active or passive depending upon whether the attacker only 'listens' to the network activity or takes part in network traffic by actively broadcasting packets. A passive attacker silently listens to the data packets, and analyzes them to find network

topologies or crypto-graphical keys. The following are different kinds of attacks that can occur in ad hoc wireless networks [12].

Routing Loop: In this attack, an attacker broadcasts false routing routes to induce a routing loop. This consumes network resources and results in degraded network performance.

Black Hole: In this attack, the attacker either becomes a black hole by directing packets from different sources to itself and dropping them all, or directs all packets to a certain node and makes it drop all the packets. This attack can decrease the throughput of the network.

Partitioning: It is caused by attacking a single node that connects two set of nodes. The attacker analyzes the network topology, and compromises the linking node to partition the network.

Rushing Attack: This kind of attacks is made by forging the sequence number in routing packets. The attacker broadcasts forged packets with a higher sequence number and the source address of the victim node. When the destination node receives the genuine packet with a lower sequence number, it drops those packets thinking they are duplicate packets. This attack can hamper route discovery.

2.3 Literature Survey

This section briefly introduces three research works, which discuss various techniques of intrusion detection and recovery in wireless ad hoc networks. This thesis is motivated by these ideas, and has adopted some parts of them to build a novel architecture of an

Intrusion Detection System (IDS). The following sections discuss only the essence of these papers that is directly or indirectly related to this research.

2.3.1 Intrusion Detection Using Mobile Agents in Wireless Ad Hoc Networks [2]

This paper discusses a distributed intrusion detection system for ad hoc wireless networks based on the mobile agent technology. The authors proposed a multi-sensor intrusion detection system which employs cooperative intrusion detection. This design has a modular IDS architecture and distributed intelligent mobile agents in a small number of nodes to achieve an adequate degree of intrusion detection.

The agents are categorized as action agents, decision agents and monitoring agents. Monitoring agents are further classified into packet monitoring sensors, user activity sensors, and system-level sensors. These agents cooperatively monitor both the network and the hosts, and take decision accordingly. For example, packet monitoring sensors monitor packets in the network for any trace of attacks, and the host monitoring sensors constantly look for any suspicious activities on the host node, such as unusual process memory allocations, CPU activity, I/O activity and user operations (invalid login attempts with a certain pattern, super-user actions, etc). Each sensor monitors either the network or the host actively, and reports to the decision agent. Furthermore, they form a cluster with a clusterhead that monitors every member in the cluster. The cluster heads are formed by voting, and they are kept within either one hop or two hops distance depending upon the depth of security required in the network.

In this design, the authors assumed that both the decision agent and the packet monitoring agent are present in the same node. The decision agent works completely based on evidence of anomalous activities gathered by each node from its local

monitoring agents and the packet-monitoring sensors. The IDS was proposed to work as a local anomaly detection model so that any thing that falls outside the preset profile is classified as a possible intrusion.

2.3.2 Architecture of the Mobile Ad-hoc Network Security (MANS) System [1]

In this paper, authors proposed a Law-Governed fully decentralized scalable security policy with the Local Collaborative Group function. It is assumed that the IDS is installed in every node collecting local data from both the host node and neighboring nodes. The data are then used to identify any malicious activities in the network.

The authors proposed to form a Local Collaborative Group with well defined laws to be followed by each node in the neighborhood. In this work, each node is associated with a "group" to cast votes among its neighbors, and it can pass "messages" to share information.

The foundation of this scheme is a neighborhood-based 'watch', which is a process of monitoring the neighbors either periodically or triggered by some events. All nodes can simultaneously and independently derive conclusion of any security breach in the network. By forming the local collaborative group, nodes perform majority voting among members within the same group to derive conclusions about whether a node being voted is compromised or not. If there is a tie, it is repeated again for a certain number of times, after which the node is decided to be bad. Implementation of this policy derives three different security levels: benign, suspicious and compromised. Finally, if a node is found to be compromised, it is barred from the network by global declaration that the node is malicious.

2.3.3 Intrusion Detection in Wireless Ad-Hoc Networks [9]

In this paper, the authors proposed an architecture, in which IDS is installed in each node so that each node can detect the signs of intrusion locally and independently. These signs are analyzed and investigated in a broader range by neighboring nodes collaboratively. The simplified version of IDS is conceptually structured into six pieces as shown in Figure 2.1.

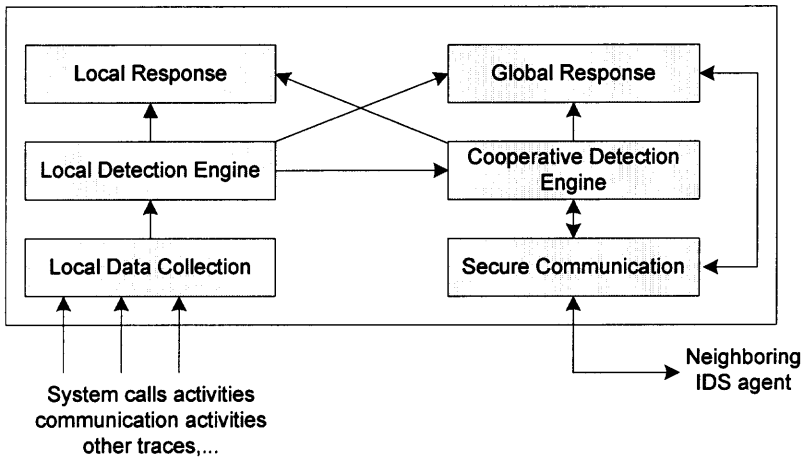

Figure 2.1 Architecture of IDS.

Here, a local data collection table gathers streams of real-time data from various sources. These data are analyzed for any trace of attacks in each node with a certain percentage of confidence by the local detection engine. If there is any warning of malicious activities, it first triggers the local response. If the warnings remain inconclusive, agents seek cooperative help by triggering cooperative detection engine. Cooperative detection is done with the aid of other nodes in the network. Nodes are

suggested to derive conclusions about the global intrusion detection by using majority voting. Finally, secure communication is achieved by nodes performing global response to the threat.

CHAPTER 3

HOMOGENOUS CLUSTERS

Homogenous clusters are the novel kind of clusters proposed in this thesis to address the security issues in a wireless ad hoc network. A homogenous cluster, which does not have a cluster head, consists of a group of nodes in the vicinity of one another (within one hop distance). They form a logical bond with each other and share equal responsibilities to keep the network secure.

This chapter begins with an introduction to the basic information that each node must maintain in order to form a cluster. It then describes how a homogenous cluster is formed in an ad hoc wireless network by using only the data extracted from the MAC layer, independent of any higher layer routing protocols. Finally, it describes the process of breaking a cluster with size greater than five into two clusters – each with a size less than five.

3.1 Information Maintained by Each Node to Form a Cluster

Figure 3.1 shows the data – Cluster Maintenance Table and Precedence Queue list maintained by each node to form a cluster.

Cluster Maintenance Table

Cluster members	Permanent/Temporary	Node been verified/unverified

Precedence Queue

Member1	Member3	Member2	Member4	(NEW)

Figure 3.1 Cluster maintenance table and precedence queue list.

Cluster members: A homogenous cluster consists of a number of cluster members. The cluster member field enlists the addresses of all the cluster members in the cluster. Heuristically, a homogenous cluster is suggested to have three to five nodes at any given time. Hence, each cluster member only needs to monitor a small number of nodes, and this reduces the overhead of monitoring. Once a node becomes a member of the cluster, it must maintain a specific set of information about all the other members in the cluster.

Permanent/Temporary: This field indicates whether the cluster membership is permanent or temporary. A cluster member is designated as "permanent" if it falls within the radio broadcast range of itself and as "temporary" otherwise. This field is required mostly in the scenario when nodes are sparsely distributed or when they lie at the extreme ends of the network as discussed in Section 3.2.2. This field can have two values - TRUE or FALSE.

Node been verified/unverified: A new node is allowed to join the cluster as soon as the cluster allows it to do so. Once it joins the cluster, each member verifies its neighboring nodes. This field indicates whether verification of each cluster member has been

completed. It could be used when weighted voting is adopted or an extensive security is desired. This field has two values - TRUE or FALSE.

Precedence Queue: This field is required to give priority to nodes when more than one node request for voting for the same reason. While forming the cluster, it gets filled on a first-come-first-serve basis. Any new node joining the cluster is appended to the first vacant position from the beginning of the list. The first position of this queue is called the 'notch' and it bears the highest priority. Precedence queue is a circular linked-list, and is updated when the 'notch timer' expires. The notch timer is the time duration, for which each member gets privilege of being the 'notch' in the precedence queue. Each member gets an opportunity to become a 'notch' with the period as defined by the timer value in the 'notch timer'. If any of the node member moves out of the cluster, the precedence queue is updated so that all members in the precedence queue behind it advances one step forward.

3.2 Voting Policy

The homogenous cluster follows a majority voting policy. Nodes in the cluster decide any action to be taken by voting. Any node that detects an anomaly in the network can request for a 'vote' from its cluster members. Each cluster member casts its vote by referring to its own data. The member, who requests for the 'vote', collects the responses from all other members and decides the result, which is derived by a majority voting.

3.3 Forming a Cluster

3.3.1 Flow Diagram of Forming a Cluster

The procedure of forming a cluster is shown in Figure 3.3. Two different flow diagrams are merged together by a dotted line.

The left portion of the diagram denotes the flow for a node that requests to join the cluster, while the other portion shows the flow for the members of the cluster to which a new node is asking permission for membership. For example, considering Figure 3.3, the left side of the flow in Figure 3.2 is for node A, and the right side of the flow is for one of the nodes in cluster X. Each cluster is kept very small in size so that every cluster member only needs to actively monitor a minimum number of nodes around itself.

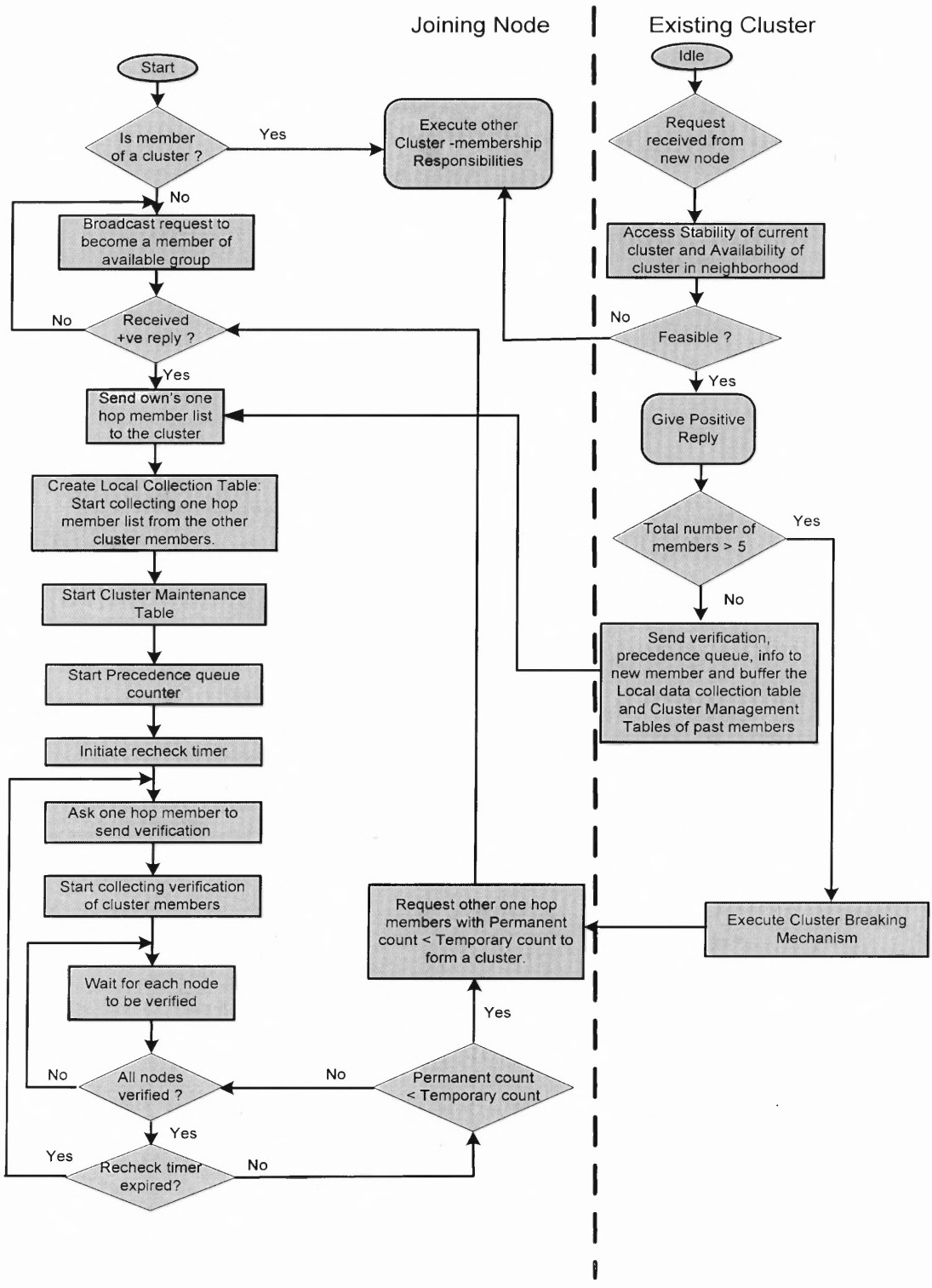

Figure 3.2 Flow diagram for forming a cluster.

3.3.2 Illustration of Forming a Cluster

With reference to the flow diagram in Figure 3.2, a simple illustration of how a cluster is formed is given in Figure 3.3..

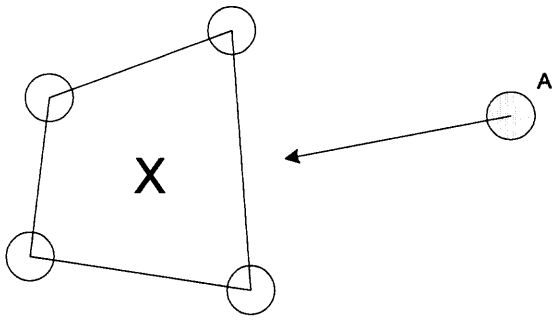

Figure 3.3 New node joining the cluster.

Consider that a node A wants to join cluster X in Figure 3.3.

- Node A broadcasts a request to become a member of cluster X.

- One of the members in the cluster will respond to the request of node A by requesting votes from other cluster members to let it join the cluster. The initiation of the voting process could be done by any of the cluster members as soon as they get the request. However, only the node which initiates the voting process can collect the vote and take the decision as resulted by the vote. If the node initiating the voting process leaves the cluster, the issue is handled by the notch in the precedence queue.

- If voting in cluster X permits, node A is granted membership and cluster X (node initiating voting) sends precedence queue to node A. A simple criterion for voting in this case is that if all the members in cluster X could 'hear' node A and the total number of cluster members in the cluster does not exceed the maximum limit, the cluster will vote for accepting Node A as a cluster member.

- Node A sends its one-hop member list to all members in the cluster, creates the local data collection table, requests all the one-hop members for verification, and waits for all the fields to get filled. This verification contains history of the new node, if it is from the last cluster member. The last cluster member sends the last recorded 'local data collection table' (explained in Chapter 4, Section 4.2) of the respective node along with the verification.

- Once all the cluster members and the new node have exchanged one hop member list, the new node is considered as a part of the cluster. The verification process could be run whenever the network resources are free and can even be completed when the cluster performs the 'ping' operation as described in Chapter 4, Section 4.3.1.

- When it receives reply from node A, each node in cluster X forms the local data collection table and sends verification requests to its corresponding one-hop members.

- As each node receives complete verification from all the one-hop members of node A, they are marked as verified.

- In extreme cases, if the network is sparsely distributed and one or two of nodes cannot form a cluster of three, then cluster may be formed so that not all the members are at one-hop distance. In this situation, Permanent/Temporary table is used to mark each other in the cluster table. If a node can hear another node, it is marked as "permanent"; "temporary", otherwise.

- Nodes have a tendency to keep the maximum "permanent" members and to maintain stability. Each node uses this criterion while voting to break the cluster.

- Once the cluster is formed, nodes exchange beacons periodically to keep the cluster informed about its presence.

3.4 Breaking the Cluster

When a new node approaches to join a cluster with five members already, a voting procedure starts in the cluster to grant it a membership. Each node gives preference to form the smallest possible cluster and votes accordingly. If the voting is successful, the new node is granted a temporary membership and added to the end of the precedence queue temporarily. When the precedence list consists of six members, a voting starts to break the cluster into two clusters – one containing the first three members and the other with the last three members of the precedence queue. Each node votes based on stability and Permanent/ Temporary count. If there is a tie in voting, voting starts to break the cluster between 2^{nd} -3^{rd} -4^{th} and 5^{th} -6^{th} -1^{st} members of the precedence queue. This

process continues until the voting is successful. If voting is still unsuccessful for the fifth time, the cluster breaks between the first three and last three members of the 'notch' queue.

3.5 Non-overlapping Concept

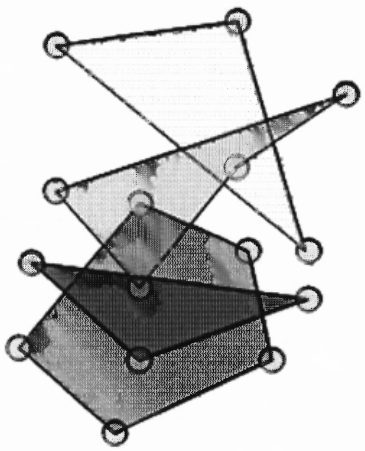

Figure 3.4 Non-overlapping cluster concept.

The homogenous clusters formed for security are non-overlapping. These clusters are logical bonds among nodes, and one node can be a member of only one cluster at a given time. Even when different nodes belonging to different clusters come together physically, they remain distinct in their behavioral property as shown in Figure 3.4. Thus, a node will be actively participating in different processes for only one cluster.

CHAPTER 4

INTRUSION DETECTION AND RESPONSE WITH HOMOGENOUS CLUSTERS

This chapter describes how various kinds of attacks can be handled by the homogenous cluster based architecture. Wireless networks have always been vulnerable to attacks because of their broadcasting nature. Section 4.1 provides an explanation of different assumptions made for the homogenous cluster based architecture. Section 4.2 describes various information required for the intrusion detection. In Section 4.3, intrusion detection for a homogenous cluster is illustrated. A case study of intrusion detection and recovery is presented in Section 4.4.

4.1 Assumptions in Homogenous Clusters

- **Nodes are bidirectional:** Nodes are said to be bidirectional if they both receive and transmit data. Since an ad hoc network relies largely on different cooperative algorithms for routing [9], any node in the path from the source to the destination has to relay data. Hence, nodes with unidirectional property are not acceptable.
- **IDS is installed in each node:** Each of the nodes in the homogenous cluster monitors its cluster members for their behaviors and triggers response accordingly. This procedure requires IDS to be installed in every node.
- **Secured key communication is already available:** This assumption is made to ensure that the MAC layer data available to form a homogenous cluster are accurate and true.

- **Thresholds are pre-analyzed and set:** To identify an attack, all nodes need a predefined set of reference data to indicate normal behavior. These predefined data are the threshold values.
- **Nodes listen to passive acknowledgements:** A passive acknowledgement is a way by which a node in a wireless network confirms the data transmitted by it has been forwarded to its destination by its immediate neighbor. In an ad hoc network, data travel hop by hop, i.e. if a destination lies beyond one's radio transmitting range, some nodes lying between them will relay the data. For example, if node B lies between node A and node C, node B will relay the data. In this process, when node A sends data to node B, it immediately broadcasts it to node C. Node A listens to node B's broadcast to C and ensures that its data have been relayed. This type of listening is called a passive acknowledgement.
- **Nodes use omni-directional antenna with equal power:** This assumption is to make sure that no nodes maliciously use directional antenna for attacks. Attacks by using directional antenna in wireless ad hoc networks are a different field of research that are beyond the scope of this thesis.

4.2 Information Maintained by Each Node for Handling Attacks

Figure 4.1 shows the database table that each node maintains to respond to an attack. Assuming there are at the most five members in a cluster, each node maintains record for five nodes. Therefore, five similar tables are shown in the figure. An explanation of each field in the table follows.

One-hop members (Addresses): This field contains the addresses of the immediate neighboring nodes of each cluster. An immediate neighbor of a cluster member is defined as a node that falls within the radio broadcasting range of each other. Depending upon the density of nodes in a particular scenario, this field can have a varying number of entries and it is updated by listening to the activities of each of the cluster members. For example, if a cluster member transmits data to any node, that node is kept in record as its one hop member. This table can be optimized by keeping records of only the most recent activities. The parameter to define 'most recent' can be characterized by a timer.

Number of packets sent: This is a count of the number of packets sent to a particular address. If this count increases beyond the threshold value for a certain amount of time, the particular cluster member can be suspected of trying to flood the network.

Number of packets received: This is a count of the number of packets received from a particular address to the cluster member. This value may not be very accurate as all cluster members may not be able to 'hear' what all other cluster members have received.

Clustermember 1

Addresses:	Number of Packets sent	Number of Packets received

Clustermember 2

Addresses:	Number of Packets sent	Number of Packets received

Clustermember 3

Addresses:	Number of Packets sent	Number of Packets received

Clustermember 4

Addresses:	Number of Packets sent	Number of Packets received

Clustermember 5

Addresses:	Number of Packets sent	Number of Packets received

Figure 4.1 Local data collection table.

4.3 IDS Concept for Homogenous Clusters

In a homogenous cluster, intrusion detection and response are classified in two different categories: a node detects anomaly in other nodes, and a node experiences attacks from other nodes.

First, if a node observes any malicious activities from any node, it confirms the situation by gathering votes from cluster members. If the gathered votes indicate positive towards anomaly, it sends a complaint to the cluster that possesses the misbehaving node.

Upon receiving the complaint about its cluster member, the cluster will verify the complaint and issue warning to the corresponding node if it is found guilty. If the node continues the misbehavior, it will be banned from the network.

Second, if a node is experiencing an attack, it will ask for 'help' from its cluster members. Upon receiving request for help, cluster members will verify its member's need for help and act to support it.

In any situation, decision making is done cooperatively by voting only. Hence, no individual node is allowed to manipulate the network behavior without consulting the cluster members. For example, no nodes can send any decision making packets such as packets indicating banning a node from a cluster without casting a vote. Any node not following this rule is deemed malicious.

These concepts have been used to illustrate proposals for addressing different kinds of attacks in Section 4.4.

4.4 Case Study

4.4.1 Case I: Finding Hiding Nodes

In a wireless ad hoc network, nodes have limited battery power. It is always in the best interest of a node to save the battery life by contributing the least service to the network. This is, however, not acceptable in an ad hoc network where existence of the whole network depends upon the cooperative activities of every node. If there is a malicious node that selfishly expects its own data packets to be carried by the rest of the nodes in the network, and tries to hide when it is its own turn to contribute to the network by relaying data packets of other nodes, it can be identified by implementing the homogenous cluster.

Selfishly hiding nodes can be traced by implementing a slightly improvised way of pinging. Figure 4.2 illustrates how a hidden node is identified.

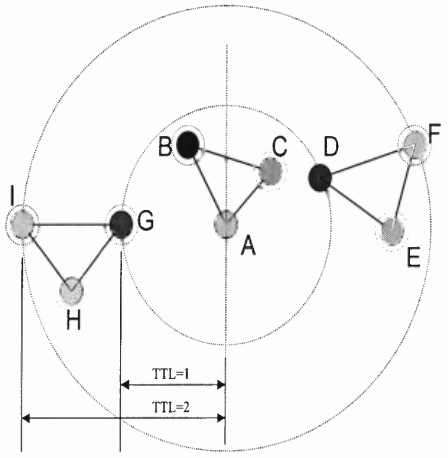

Figure 4.2 Ping with TTL = 2.

Suppose that node A is suspicious about any hiding nodes in its one hop neighborhood. It will simply send a ping request with TTL= 2. TTL is set to 2 so that it will spread to all its neighboring clusters. In return, whoever gets the ping request has to ping back with TTL=1. The node A will listen to all the pings sent by neighboring nodes and identify its one hop neighbors. When a ping request is received, the entire cluster member will make sure that the other members of its cluster also respond. If any of the cluster members does not ping back, it will send a ping request to that particular node. If the node still does not ping back, it is deemed malicious. For example, in Figure 4.2, if node D does not ping back, node E and F will request it to ping. If it still does not ping back, it is considered to be malicious. This pinging procedure can also be used to find one hop members of nodes periodically.

4.4.2 Case II: Stopping Flooding

Flooding can be controlled by a cluster with internal voting itself. If any node is sending excessive data (beyond the threshold limit) to any other node, the cluster will give it a warning. If it continues the misbehavior, it will be banned from the network. Any attempt of radio jamming by any node can be solved only by physically removing the node from the network.

4.4.3 Case III: Coping Black Hole Attack/ Curbing Packets Dropping

Assume there are three clusters ABCD, EFGH and IJKLM in the network as shown in Figure 4.3. Consider that a node in cluster 1 (C) needs to transmit data to a node in cluster 3 (M) via a node in cluster 2 (E). The node C collects passive acknowledgements of its data being relayed further from node E. If node C detects node E is not forwarding its data packets, it will ask to vote to give warning to node E. The corresponding members in cluster 1 will check the local collection table to verify how many packets have been transferred to node E and possibly the number of packets transmitted to node C by E. (Note that the number of packets received by node C may not be very accurate). Upon request for warning, the cluster will vote to give warning, and a warning will be sent to cluster 2 for node E. Once the warning has been sent, nodes in the cluster switch to the mode that they actively monitor all packets sent by node C to E. Cluster members in cluster 1 will make sure that node C is sending packets that are required to be relayed by node E.

In cluster 2, nodes will check their local collection tables on how node E is forwarding its packets to the denoted next hop. If E is found guilty, they will vote to warn

node E. Even in cluster 2, cluster members will switch to the mode where they actively monitor all packets from E checking whether it is forwarding packets or not.

Node C will again try to send packets to node M via E, and both the cluster 1 & 2 will monitor this activity. If node E does not correct itself and continue misbehaving, cluster 1 will again send a complaint to cluster 2, and cluster 2 will permanently block node E and redefine their routing tables if necessary.

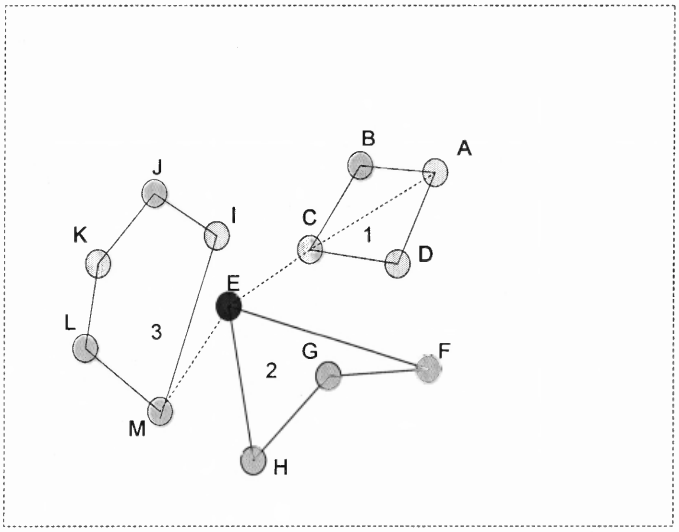

Figure 4.3 Three cluster scenario.

4.4.4 Case IV: Checking DDoS Attack

DoS attack is carried out by flooding the resource so that the network cannot operate correctly. Since the ad hoc network operates in a distributed manner, this cannot be done by a single node. Any node trying to flood can easily be handled as mentioned in case I. If a node can change the routing table of some nodes, it may be able to stage Distributed Denial of Service (DDoS) attacks on any of the nodes in the ad hoc network.

Consider that node M and node C in Figure 4.3 are compromised. Each of them sends packets continuously to node E within the threshold limit of each other. But for node

E, the rate of receiving packets from node C and M together exceeds its basic limit. In this case, it will request its cluster members for help and put a check on nodes in cluster 3 and 1. Nodes in cluster 2 will verify node E's request and request cluster 3 and cluster 1 to check node C and M, respectively. If node M and C have been relaying the packets, the request for check will be relayed to the corresponding source. If the nodes C or M themselves are the source, they can get a warning about ongoing malicious internal activities in themselves. If a node cannot prove that it has been relaying the data or if it cannot change its activities, it will be deemed malicious and proper action will be banned from the network.

CHAPTER 5

SIMULATION MODELS OF HOMOGENOUS CLUSTERS

This chapter describes the simulation models that have been developed to test and evaluate the homogenous cluster-based architecture in mobile ad hoc networks.

5.1 Introduction

The objective of this simulation study is to determine the network overhead of the homogenous cluster-based architecture. All simulations have been performed using the discrete event simulator OPNET Modeler™ version 7.0. OPNET is commonly used for network simulations and provides a rich library of models for implementing wired and wireless simulation scenarios [13]. Section 5.2 begins with a brief discussion of the modeling and simulation methodology used in OPNET. Section 5.3 describes the design and interface of the 'security_processor' used to create homogenous clusters. Sections 5.4 through 5.6 discuss different aspects of the 'security_processor', and Section 5.7 describes the modifications performed in some of the library modules of the OPNET to suit the 'security_porcessor'.

5.2 Modeling Using OPNET

OPNET is a discrete event network simulator used both by the commercial and research communities. It provides a comprehensive engineering system capable of simulating large communication networks with detailed protocol modeling and performance

analysis. It also provides a number of built in libraries for modeling wired as well as wireless network scenarios [13].

Simulation models are organized in a hierarchy consisting of three main levels -

- The simulation network
- The node model
- The process model

5.2.1 The Simulation Network

This is the top level for the simulation of the network scenario. It defines the initial topology of the network and the scale of the network. It also allows the configuration of attributes of the nodes comprising the scenario.

5.2.2 The Node Model

This is the second level in the hierarchy and consists of an organized set of modules describing the various functions of the node. The functionality of each of these modules in the node model is described by their process state as described in the process model.

5.2.3 The Process Model

This is the lowest level in the hierarchy. They contain the finite state machines, definitions of model functions, and a process interface that defines the parameters for interfacing with other process models and configuring attributes. Finite state machine models are implemented using Proto C, which is a discrete event library based on C functions and OPNET kernel itself.

5.3 The Design and Interface of the 'Security_Processor'

The 'security_porcessor' has been designed from scratch in the OPNET to form and operate homogenous clusters. This module works independently of any routing protocol implemented in each node. The basic purpose of this module is to collect data from the MAC layer and populate the 'Local data collection table' and the 'Cluster management table' with the recently available values. These values are then monitored by threshold detectors to detect any aberrant activities in the network.

In a homogenous cluster, each node is assumed to have a security processor that communicates with a security processor of the neighboring nodes via its own security packets. These security packets are encapsulated in the 802.11 MAC layer packets and broadcasted so that all the cluster members can collect them. A conceptual view of inter-security processor communication is shown in Figure 5.1. The details of the security packet format can be found in Section 5.4.

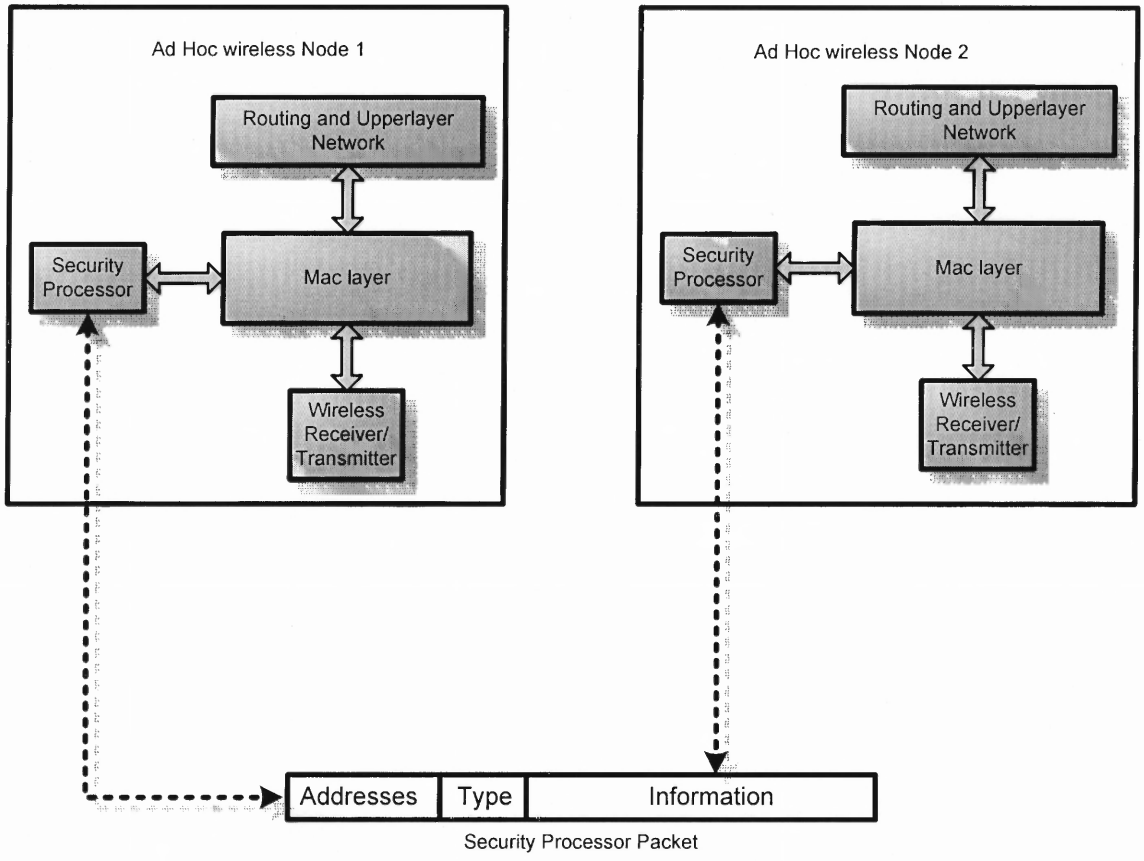

Figure 5.1 Intra-security-processor packet communication.

5.4 Security Packets

Security packets have been created using the OPNET Packet Format Editor. It has seven different fields as shown in Figure 5.2.

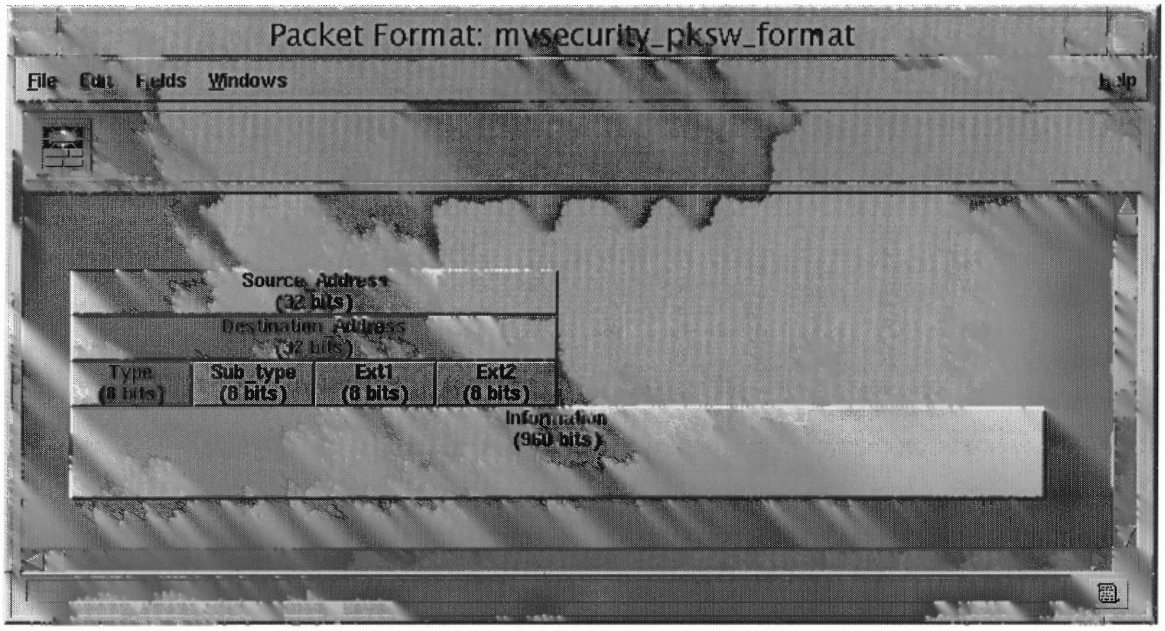

Figure 5.2 Security packet format.

Each of these fields carries the following information.

Source_Address: This field carries the MAC address of the node that is broadcasting the packet.

Destination_Address: This field carries the address of the node in question. For example, if node A is asking vote concerning node B, Source_Address will have address of A, and Destination_Address will have the address of B.

Type, Sub_type, Ext1, Ext2 : These fields collectively denote the kinds of issues for which security packets have been broadcasted. These fields can be populated with new kinds of issues in the course of development. A basic tree structure of the proposed field value is given in Figure 5.3. In this thesis, only the values related to cluster formation activities have been used. An example is given below to illustrate how these values are used.

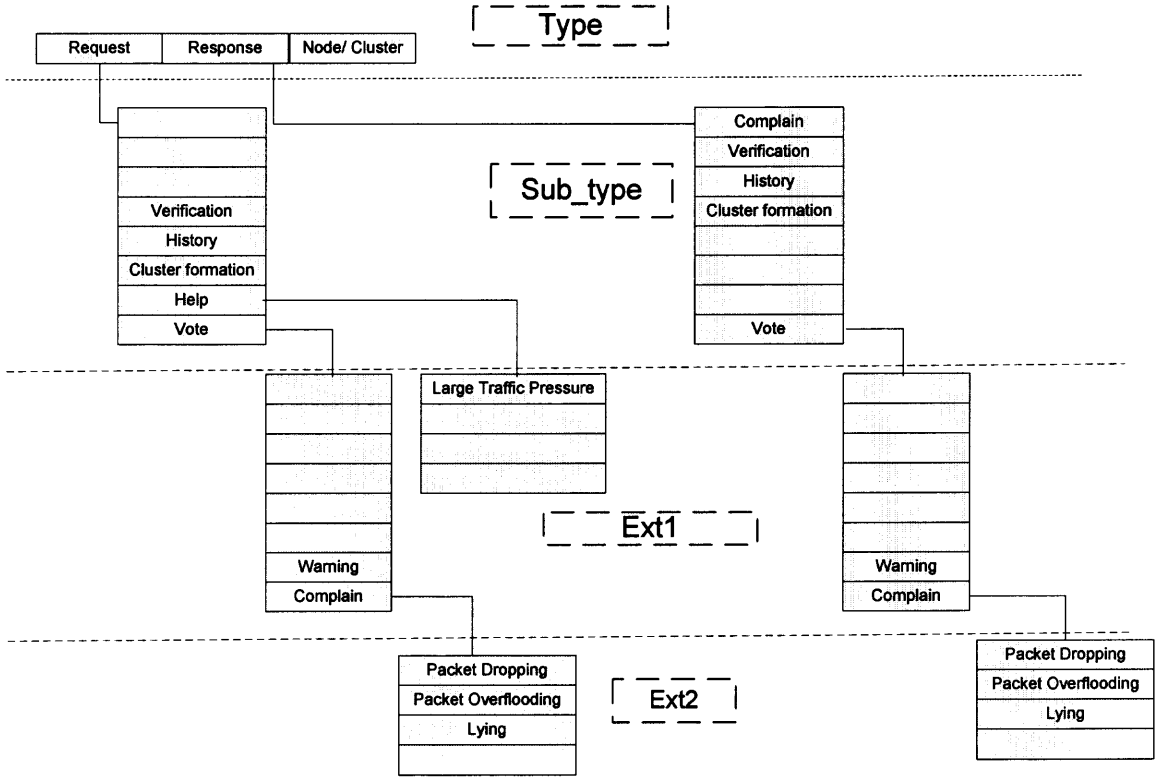

Figure 5.3 Packet type tree.

Example

If node A is requesting a vote complaining over packet flooding by node B, it will have the following values for Type, Sub_type, Ext1, Ext2.

Type : Request

Sub_type: Vote

Ext1: Complain

Ext2: Packet Flooding.

Information: This field carries information present in 'local data collection' or 'cluster maintenance table' as and when required.

5.5 Node Model

The node model developed for this simulation is shown in Figure 5.4.

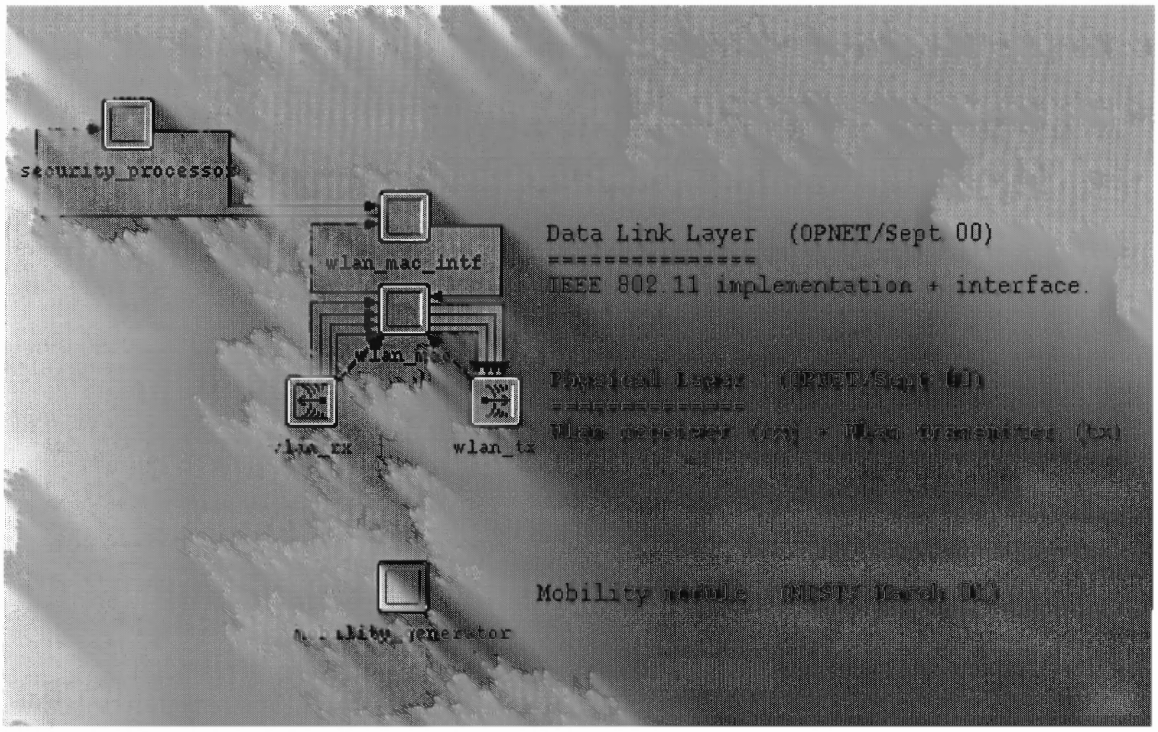

Figure 5.4 Node model of an ad hoc mobile unit with the security processor.

For this simulation, different modules such as wlan_mac, wlan_mac_intf, wlan_rx and wlan_tx have been adopted from OPNET's built-in library with some modifications. The mobility generating module mobility_generator is adopted from a library developed by NIST (National Institute of Standards and Technology) which is available for download from their website [17]. It describes a general billiard mobility in which positions of the nodes are changed periodically after a certain time. In order to program the security processor, detailed studies of different modules provided by NIST have been done. As a result, a few modeling and programming techniques are borrowed from their modules.

5.6 Process Model of Security Processor

The process model of the security processor is developed from scratch to simulate the homogenous cluster. The process model (state machine) designed in the OPNET is shown in Figure 5.5.

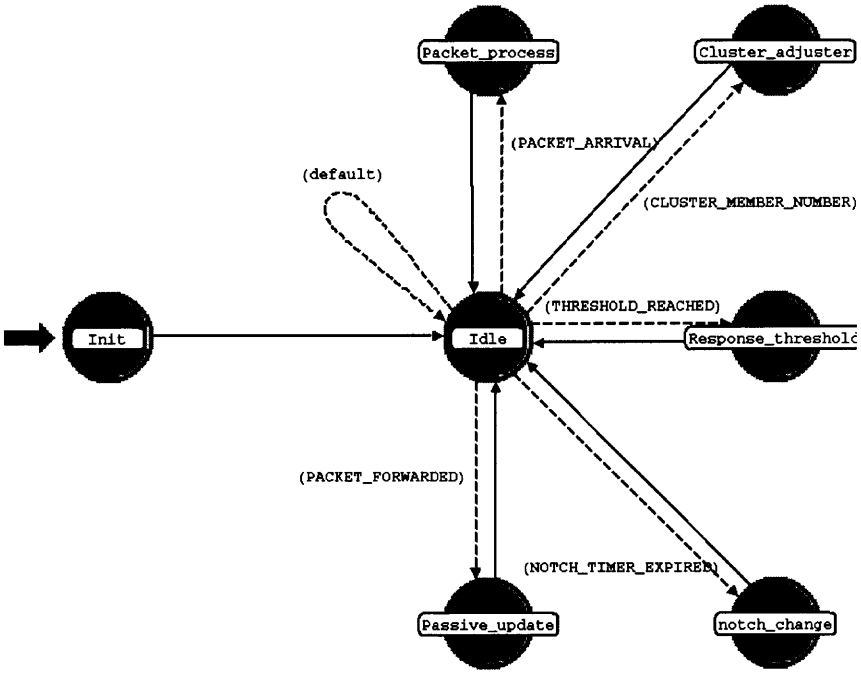

Figure 5.5 Process model of the security processor.

The functions carried out by each of these states are briefly explained as follows.

Init: This is the very initial state where all the state variables, voting table, 'local data collection table', 'cluster maintenance table' and other event handles are initialized. This state prepares a node to perform its job by supplying all the initial attributes.

Idle: This is the state where the process waits for any interrupts to occur. As the processor encounters interrupts by various events, it is directed to one of the states connected to it for processing. For example, if the processor gets 'instream interrupt'

denoting arrivals of packets, control is directed to the packet_processor. As soon as the function is completed, the control returns back to this state.

Packet_processor: This state handles all the incoming packets. It is the main state that carries out most of the functions like deciding about vote, accepting vote, and taking other decisions.

Cluster_adjuster: This state is triggered as soon as the node realizes that it has less than two members in its 'cluster member' list. It can also be programmed to be triggered periodically to find the one hop members by providing counts for the ping counter. This state can request for ping, or broadcast request to become a member of a cluster.

Response_threshold: This state is triggered whenever a node discovers any threshold has been reached. This state handles all threshold counts, and requests vote to cluster members to verify its data. A node can request for only one type of vote at a time. Hence, once it requests for a vote, it cannot request votes for other issues unless the previous request has been resolved. The replied vote goes to the packet_processor, and decision is taken in the same state.

Notch_change: This state bears the notch counter that expires periodically to change the position of notch in the precedence queue.

Passive_update: This state is used for passive listening to fill 'local data collection table'. This state has not been programmed and is left for future development.

5.7 The Modifications Performed in the Built-in Library of OPNET's 802.11 WLAN Module

This section describes the changes that are made in OPNET's 802.11 WLAN module to interface the security processor.

wlan_mac module: In this module, changes are made such that it will collect data for all its cluster members, i.e., if this module receives any packet destined to its cluster member, it will update its cluster member table.

wlan_mac_intf module : In this module, input and output streams are directed towards the security processor to get a free flow of security packets from the security processor.

CHAPTER 6

SIMULATION RESULTS

This section describes the results obtained by simulating the module described in Chapter 4. Section 6.1 describes the simulation variables. Section 6.2 describes the general simulation procedure, and Section 6.3 explains different results in more details.

6.1 Simulation Variables

The different parameters used to produce the simulation results are provided in Table 6.1. All these parameters are common for all the simulations.

Table 6.1 Simulation Parameters

Parameters	Value
Transmitter range	200m
MAC buffer size	256000 bits
Node mobility speed	1 m/s
Simulation time	180s
Data Rate	1 Mbps
Scenario size	500x500m
Number of nodes	3
Physical Characteristic	Frequency Hopping Spread Spectrum
Mobility module	Billard mobility
MAC layer protocol	802.11, ad-hoc mode

6.2 Simulation Process

The simulation starts when all the required parameters are set. The process stops as soon as the goal for the particular scenario is met. Once it is over, graphs are plotted for the analysis of the results. The graphs typically show the number of packets collected by each node in a certain time duration. Note that the packet counts are only the security

packets that have been exchanged, and the count for MAC layer packets like RTS, CTS or ACK packets is not shown.

6.3 Simulation Results

6.3.1 Scenario I: Initial Cluster Formation.

In this scenario, three nodes are turned ON at the same time. As soon as the simulation starts, the nodes actively get in the act to form a homogenous cluster. The simulation stops as soon as their cluster formation process is completed. The initial positions of the nodes in the scenario are shown in Figure 6.1, and the results of the simulation are shown in Figure 6.2.

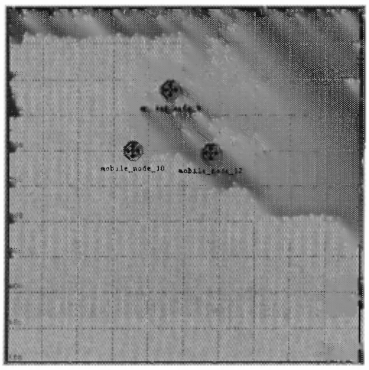

Figure 6.1 Initial positions of the nodes.

The packet counts seen at each node (Figure 6.2) are different because the node turns idle as soon as it successfully becomes a member of the cluster. The six packets received by a node to form a cluster can be justified as follows. First, each of the node broadcasts packets to become a member of a cluster, and this accounts for two packets each node 'hears'. Whenever a node 'hears' request for membership and finds itself in need of a membership, it immediately sends acceptance, and this accounts for two more packets a node 'hears'. Finally, when a node is accepted as a member, it sends its basic

'information' to all other nodes and this accounts for the final two packets. In Figure 6.2, 'mobile_node_12' starts broadcasting its request first. The remaining nodes 'listen' the request broadcasted by the node, and start counting the packets. Thus, the count for received nodes starts for 'mobile_node_10' and 'mobile_node_9' earlier than that for 'mobile_node_12'.

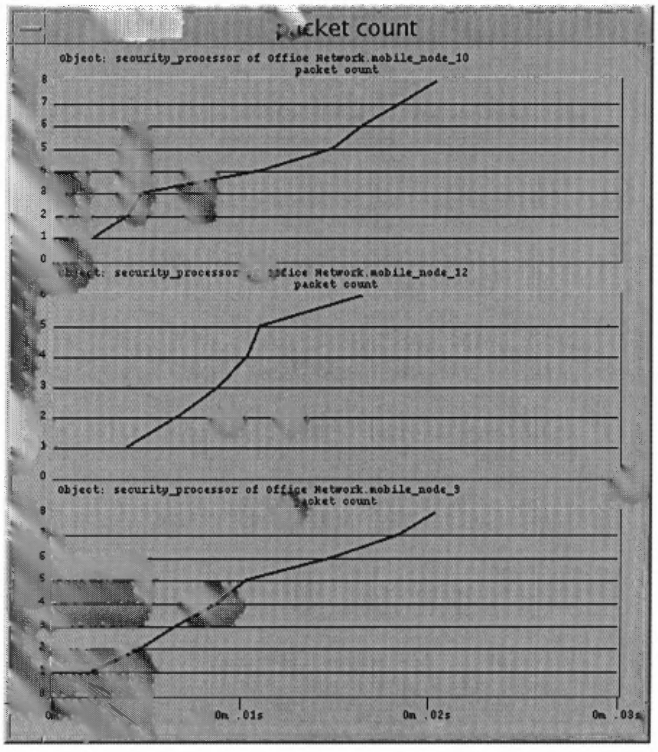

Figure 6.2 The total numbers of packets exchanged and the time taken to form an initial cluster of 3.

6.3.2 Scenario II: New Node Joining Pre-existing Cluster

In this scenario, a new node joins a cluster of three nodes. The cluster is formed before the simulation starts, and the new node is placed in the radio range of all the nodes in the cluster. When the simulation begins, the new node asks for membership and the cluster permits membership after voting. The simulation stops as soon as the new member

successfully joins the cluster. The initial positions of the nodes are shown in Figure 6.3 and the simulation results are shown in Figure 6.4.

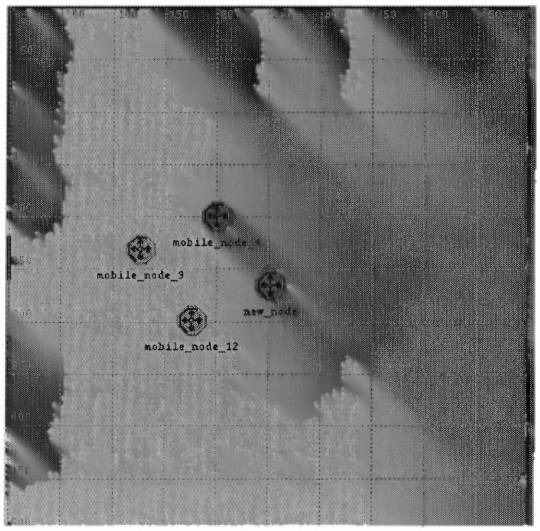

Figure 6.3 Initial positions of the nodes when the 'new_node' asks to join the cluster.

From Figure 6.4, it can seen that the 'new_node' starts broadcasting a request to become a member of a cluster, successfully finishes the process in 0.03s, and becomes idle after exchanging 11 packets. The members of the pre-existing cluster have to exchange more packets because of the voting process. By the time the 'new_node' exchanges 11 packets, the other nodes exchange 15 packets to arrive in the decision process. They 'listen' to two more packets after 0.03s, because one of the packets is the confirmation packet from the 'new_node' and the other one is the confirmation sent by each of the nodes to the 'new node'.

Comparing scenario I and II, it can be seen that joining a pre-existing cluster takes a longer time than forming a cluster initially. This is because scenario II involves the voting procedure being carried out to arrive at the decision about forming a cluster.

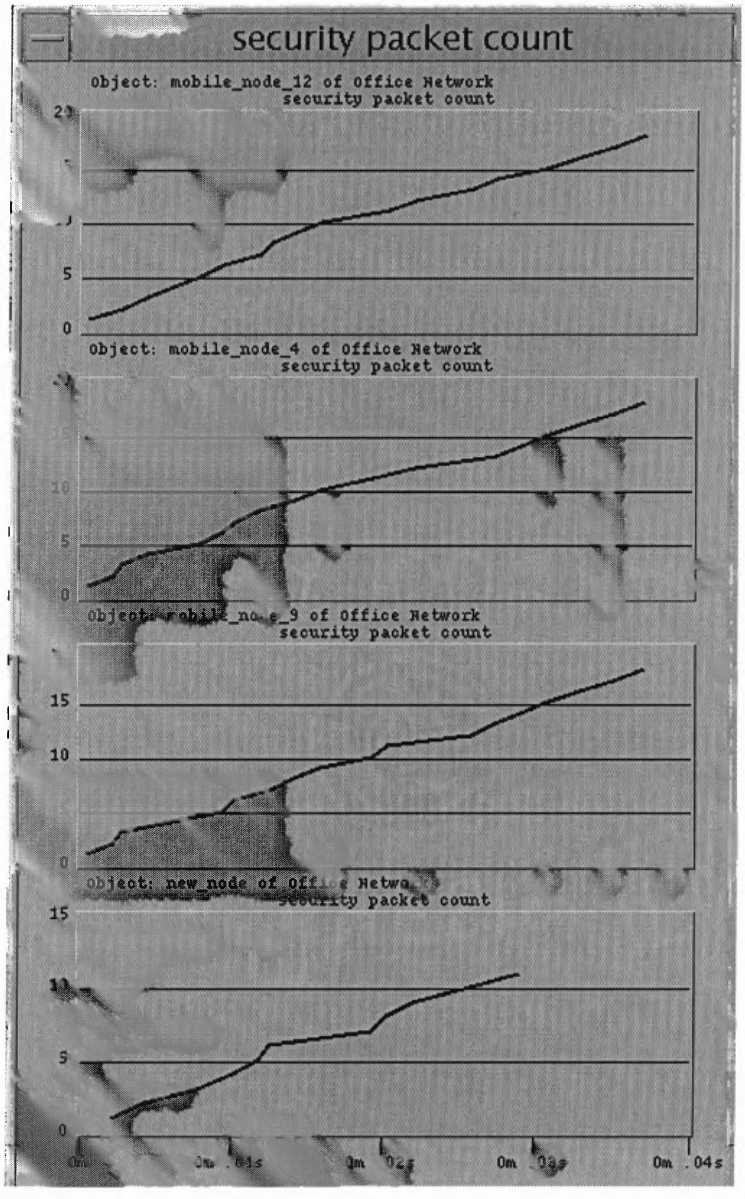

Figure 6.4 The total number of packets exchanged and the time taken for new node to join the cluster.

6.3.3 Scenario III: Detecting an Attack

In this scenario, a simple attack is detected in a cluster of three nodes. One of the nodes constantly sends data packets at the rate of 1 packet/s. Other cluster members are programmed to remove any node, which broadcasts more than 50 packets from cluster member list after consultation. The simulation result shows how cluster members

successfully stop counting any data packets as soon as they receive confirmation of the attack. This scenario is used to test the viability of using homogenous cluster to detect any attack. Figure 6.5 shows the initial positions of the nodes in the scenario. In Figure 6.6, packets received by a good node and a bad node before and after the attack are shown. Figure 6.7 reproduces the graph in Figure 6.6 for a bad node with a different time scale showing the exact time taken for the bad node to be ignored by other good nodes.

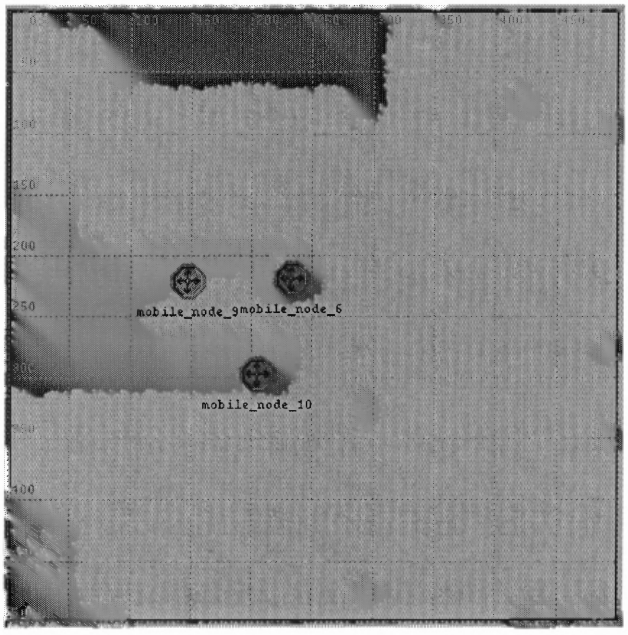

Figure 6.5 Initial positions of the nodes where 'mobile_node_6' is the bad node.

From Figure 6.6, it can bee seen that, as soon as the bad node ('mobile_node_6') broadcasts the 50th packet, the good nodes vote to confirm its activities. They exchange three packets in voting, which is 'heard' by the bad node as seen in Figure 6.6. Once the issue is confirmed, the good nodes stop accepting any packets from 'mobile_node_6', and show the constant number of packets received in their count. When the time scale is changed, it can be seen that it takes only 0.002s for the cluster to make the decision that the bad node has reached the threshold of broadcasting 50 packets.

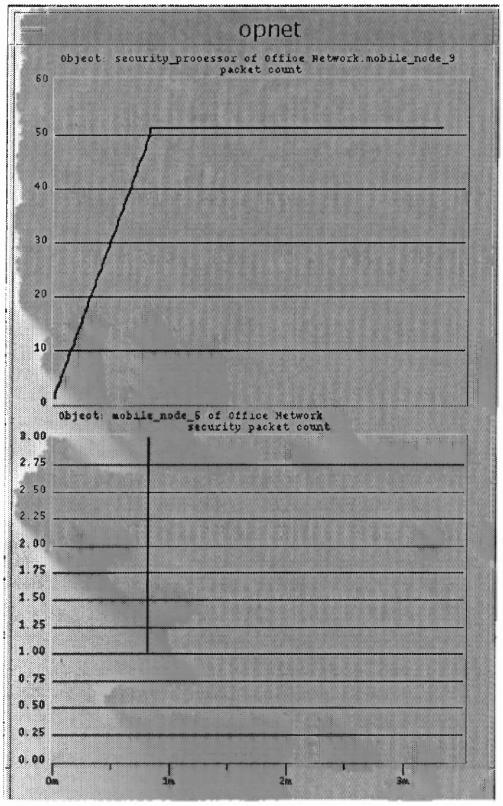

Figure 6.6 The number of packets received by a good node and a bad node in the attack scenario.

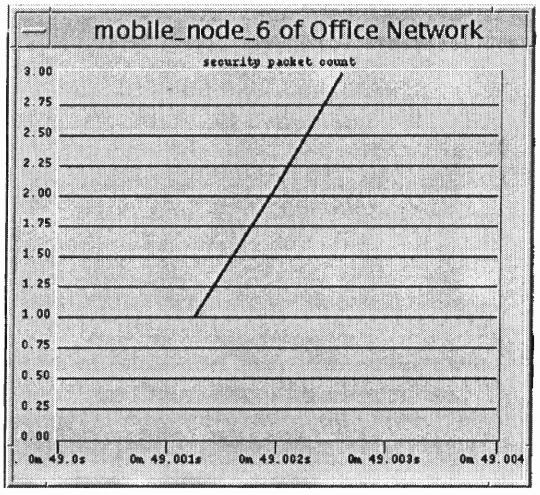

Figure 6.7 The reproduction of graph in Figure 6.6 for a bad node with a different time scale showing the exact time taken for the bad node to be ignored.

CHAPTER 7

CONCLUSION AND FUTURE WORK

This chapter concludes the thesis by 1) summarizing the contributions, and 2) outlining the future research directions.

7.1 Contributions

This thesis proposes a novel homogenous cluster to restrict attacks in wireless ad hoc networks. Compared with other cluster-based schemes [16] [20] [21], this new proposal is characterized by the following aspects. First, no cluster head is required in each cluster. Second, inspired by the work in [6], 'precedence queue' is used to help load balancing and distribute responsibilities among all cluster members. Third, the 'cluster management table' is developed to support the homogenous clusters operation. Various fields in the table are obtained heuristically.

Among many types of attacks in wireless networks [4] [5] [7] [11] [19] [23], some basic attacks have been investigated in detail within the framework of the homogenous cluster-based architecture.

As another goal of this thesis, simulations are carried out to find the feasibility of forming a homogenous cluster for securing a wireless ad hoc network. A mobile node is created with a 'security processor' in OPNET. This 'security processor' can interact with other security processors in other nodes by using security packets. It is found that it takes less than 0.05 second to form a homogenous cluster with less than 18 packets exchanged among all nodes in the cluster at 1Mbps data rate for the given scenario. The delay is

tolerable and the network overhead is acceptable for a network with moderate mobility nodes.

7.2 Future Works

This thesis has created some future research opportunities. First, it is interesting to investigate how to set the different performance matrices such as TTL duration and notch_timer duration so that the system can achieve the optimal performance. Second, it is important to study which voting strategies, weighted voting or majority voting, can achieve better performance. Third, the likelihood of editing the 'cluster member table' and 'local data collection table' with various other fields requires further study. Fourth, a prospect of deriving different threshold values for triggering interrupt in the security_processor based on QoS factors in the network can also be studied for finer granularity. Fifth, more simulations for various attack scenarios need to be done to evaluate the effectiveness of our intrusion detection and recovery scheme.

APPENDIX

THE IEEE 802.11 WIRELESS LAN STANDARDS

This appendix provides a brief introduction to IEEE 802.11.

IEEE 802.11 has defined a set of specifications for both physical and medium access control layer of a network. The specifications allow three different kinds of technologies for transmission of data between nodes, namely, frequency-hopping spread spectrum, direct sequence spread spectrum, or infrared (IR) pulse position modulation.

The MAC layer uses CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) for sharing the common wireless medium. CSMA/CA is different from CSMA/CD (Collision Detection) which is used in Ethernet in the sense that a node willing to transmit a packet does that only when it senses the channel is free. CSMA/CD cannot be implemented in wireless networks because a node cannot hear any other nodes when it is transmitting because signals from other nodes will be overwhelmed by its own signal [24].

Furthermore, 802.11 maintains the order of transmission by using short message packets like RTS (Ready to send), CTS (Clear to send) and ACK (Acknowledgement). Exchange of these messages is necessary in the wireless medium for collision avoidance in hidden nodes. When two nodes are far apart and they cannot listen to each other, they could transmit data any time when they think the channel is free. However, if there is a node amid them and can listen both the far apart nodes, it will constantly experience collision. This kind of problem is called the hidden node problem. Use of short messages is done to avoid this problem.

REFERENCES

[1] Manikopoulos, C.; Ling, L. "Architecture of the Mobile Ad-hoc Network Security (MANS) System," In *IEEE International Conference on Systems, Man and Cybernetics, 2003*, Volume: 4, October 2003, pages: 3122-3127.

[2] Kachirski, O.; Guha, R. "Intrusion Detection using Mobile Agents in Wireless Ad-hoc Networks," In *Proceedings of IEEE Workshop on Knowledge Media Networking, 2002*, July 2002, pages: 153-158.

[3] Mishra, A.; Nadkarni, K.; Patcha, A. "Intrusion Detection in Wireless Ad-hoc Networks," In *IEEE Wireless Communications [see also IEEE Personal Communications]*, Volume: 11, Issue: 1, Feb 2004, pages: 48-60.

[4] Lim, Y. X.; Yer, T. S.; Levine, J.; Owen, H. L. "Wireless Intrusion Detection and Response," In *Information Assurance Workshop IEEE Systems, Man and Cybernetics Society*, June 2003, pages: 68-75.

[5] Parker, J.; Undercoffer, J.; Pinkston, J.; Joshi, A. "On Intrusion Detection and Response for Mobile Ad-hoc Networks," In *IEEE International Conference on Performance, Computing, and Communications*, April 2004, pages: 747-752.

[6] Amis, A. D.; Prakash, R. "Load-balancing Clusters in Wireless Ad-hoc Networks," *Proceedings of 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology*, March 2000, pages: 25-32.

[7] Nadkarni, K.; Mishra, A. "A Novel Intrusion Detection Approach for Wireless Ad-hoc Networks," In *IEEE Wireless Communications and Networking Conference (WCNC 2004)*, Volume: 2, March 2004, pages: 831-836.

[8] Huang, Y.; Lee, W. "Intrusion Detection: A Cooperative Intrusion Detection System for Ad-hoc Networks," In *Proceedings of the 1st ACM Workshop on Security of Ad-hoc and Sensor Networks,* October 2003. pages: 135-147

[9] Zhang, Y.; Lee, W. "Intrusion Detection in Wireless Ad-hoc Networks," In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, August 2000. pages: 275-283

[10] Murthy, C.; Ram, S. "Ad-Hoc Wireless Networks: Architectures and Protocols," Publisher: Prentice Hall PTR, c2004, Upper Saddle River, NJ, Prentice Hall communications engineering and emerging technologies series, ISBN: 013147023X.

[11] "The handbook of Ad-hoc Wireless Networks," edited by Mohammad Ilyas. Publisher: Boca Raton, Fla. CRC Press, c2003. Series: The electrical engineering handbook series ISBN: 0849313325.

[12] "Ad-hoc Networking," Edited by Charles E. Perkins, Publisher: Boston Addison-Wesley, c2001, ISBN: 0201309769.

[13] OPNET Modeler™ 7.0 Online Documentation. OPNET Technologies, Inc.

[14] Vicomsoft®, 'KnowledgeShare - White Papers', Retrieved on March 2005, http://www.vicomsoft.com/knowledge/reference/wireless1.html.

[15] Axelsson, S. "Intrusion Detection Systems: A Taxonomy and Survey," In *Technical Report No 99-15, Dept. of Computer Engineering, Chalmers University of Technology, Sweden*, March 2000.

[16] Yu, J. Y.; Chong, H. J. "A Survey of Clustering Schemes for Mobile Ad Hoc Networks," In *IEEE Communications Surveys & Tutorials,* First Quarter2005, Volume 7, No. 1.

[17] Miller, L. "Simulation Model for the AODV MANET Routing Protocol" http://w3.antd.nist.gov/wctg/manet/prd_aodvfiles.html. Retrieved on 19 January, 2005.

[18] Naldurg, P.; Campbell, R.H.; Mickunas, M.D. "Developing Dynamic Security Policies," *In DARPA Active Networks Conference and Exposition,* May 2002, Proceedings 29-30, pages: 204-215.

[19] Ghalin, C., "Secure Ad Hoc Networking," *A Master's Thesis in Computing Science,* <e-mail: claes@cs.umu.se. > Retrieved on 5 February, 2005.

[20] Basu, P.; Khan, N.; Little, T.D.C. "A Mobility based Metric for Clustering in Mobile Ad hoc Networks," In *International Conference on Distributed Computing Systems,* April 2001, pages: 413-418.

[21] Lin, C.R.; Gerla, M. "Adaptive Clustering for Mobile Wireless Networks," In *IEEE Journal on Selected Areas in Communications,* Sept. 1997, Volume 15, Issue 7, pages: 1265-1275.

[22] Birk, T.; Liss L.; Schuster, A.; Wolff, R. "A Local Algorithm for Ad Hoc Majority Voting via Charge Fusion," In *18th International Symposium on Distributed Computing (DISC),* October 2004, pages: 275-289

[23] Aad, I., Hubaux, J. P.; Knightly, E. "Denial of Service Resilience in Ad Hoc Networks," In *Proceedings of ACM MobiCom 2004*, September 2004, pages: 202-215

[24] Zytrax.com, info. 802.11 MAC (Media Access Control) http://www.zytrax.com/tech/wireless/802_mac.htm, Retrieved on 7 February, 2005.

[25] Samar, P.; Pearlman, M.R.; Haas, Z.J. "Independent Zone Routing: An Adaptive Hybrid Routing Framework for Ad hoc Wireless Networks," In *IEEE/ACM Transactions on Networking,* Aug. 2004, Volume 12, Issue 4, pages: 595-608.

[26] McDonald, A.B.; Znati, T.F. "A Mobility-based Framework for Adaptive Clustering in Wireless Ad hoc Networks," In *IEEE Journal on Selected Areas in Communications,* Aug. 1999, Volume 17, Issue 8, pages: 1466-1487.