QAIDS: Quantitative and Agent based Intrusion Detection System

Mojtaba Karami
Department of Electrical and Computer Engineering
Islamic Azad University Zanjan Branch, Zanjan, Iran
E-mail: mjt.karami@gmail.com

Marjan Kuchaki Rafsanjani (Corresponding author)
Department of Computer Science, Shahid Bahonar University of Kerman
Kerman, Iran
E-mail: kuchaki@mail.uk.ac.ir

Amir Hosein Fathi Navid
Department of Electrical and Computer Engineering
Islamic Azad University Hamedan Branch, Hamedan, Iran
E-mail: A.fathi.navid@gmail.com

Yaeghoob Yavari
Department of Electrical and Computer Engineering
Islamic Azad University Hamedan Branch, Hamedan, Iran
E-mail: yavari64@gmail.com

Abstract

Intrusion Detection Systems (IDSs) for Mobile Ad hoc Networks (MANETs) are necessary when we deploy MANETs in reality. In this paper, we focus on the protection of MANET routing protocols. Therefore, we present a new intrusion detection architecture based on quantitative, agents, and clusters that is suitable for multi-hop mobile ad hoc networks. It detects nodes misbehavior and anomalies in packet forwarding such as dropping, modifying or delaying of packets by intermediate nodes. Simulation results are given to validate the efficiency of our IDS architecture in detecting intrusions and reducing false positive rate.

Keywords: Mobile Ad hoc Networks (MANETs), Intrusion Detection System (IDS), Anomaly detection, Misbehaving nodes, DSR routing protocol

1. Introduction

The mobile ad hoc network (MANET) is an autonomous system of mobile nodes connected by wireless link to form a network. It does not rely on predefined infrastructure to keep the network connected, therefore it is also known as infrastructure-less networks. In MANET, each node is equipped with a wireless transmitter and receiver that communicate with other nodes in the vicinity of its radio communication range and those which are beyond the range can communicate using the concept of multi hop communication in which other node relay the packets (Yi et al., 2005). In MANETs, the network topology may change rapidly and unpredictably. To deals with this, nodes exchange information about network topology. So the functioning of the ad hoc network depends on the trust and cooperation between nodes.

The mobile ad hoc network have many individual characteristics such as open medium, dynamic network topology, cooperative algorithms, lack of centralized monitoring and management point, bandwidth constrained, variable link capacity, limited energy and limited physical security. Due to these features, mobile ad hoc networks are particularly vulnerable to various types of attacks. Various intrusion detection methods are developed for detecting the intrusion in the wired networks. Due to the mobility and resource constraints of nodes, the intrusion detection methods of wired network cannot be used for MANETs (Yi et al., 2005).

Intrusion detection can be defined as a process of monitoring activities in a system which can be a computer or a network. The mechanism that performs this task is called an Intrusion Detection System (IDS) (Sahu, Shandilya, 2010)(Rafsanjani et al., 2008). IDS works under following assumptions: A) User and program behaviors are observable also, B) Normal and abnormal behaviors must have distinct activities (Rafsanjani et al., 2008).

Intrusion detection can be classified based on audit data as either host-based or network-based. A network-based IDS captures and analyzes packets from network traffic while a host-based IDS uses operating system or application logs in its analysis. Based on detection techniques, IDS can also be classified into three categories as follows: *A)Misuse-based detection, B)Anomaly-based detection and, C)Specification-based detection* (Sahu, Shandilya, 2007)(Anantvalee, Wu, 2007)(Mandala et al., 2010).

In misuse based intrusion detection, also called signature based detection, a predefined pattern or signature is used to match an attack. In anomaly detection, a normal profile of user is kept in the system and then the captured profile is compared. When IDS found any behavior that deviated from the normal behavior is detected as an attack. In Specification based intrusion detection, the system monitors current behavior of systems according to specifications that describe desired functionality for security critical entities. A mismatch is reported as an attack.

Hence in this paper, we proposed a new architecture based on anomaly and agents. Also, we discuss how to distinct the intrusion and abnormal behaviors from expected and normal behavior. So, we have proposed a quantitative and statistical based algorithm for identifying misbehavior nodes. Our method is a behavioral anomaly based system which makes it dynamic, scalable, configurable and robust. Finally, we verify our method by running simulations with mobile nodes using Dynamic Source Routing (DSR) protocol as the routing protocol.

The rest of this paper is organized as follows. Section 2 reviews the state of the art scenario. Section 3 describes our proposed IDS. Section 4 presents the simulation study and performance analysis. Finally, in section 5 concludes the paper and discusses some future work.

2. State of the Art Scenario

The security difference between wired infrastructure networks and mobile ad hoc networks motivated researchers to model an IDS that can handle the new security challenges such as securing routing protocols (Mitrokotsa et al., 2006)(Otrok et al., 2007)(Mishra et al., 2004). We only list here some of the existent research work that is related to our approach.

Zhang and Lee (Zhang et al., 2003) have proposed that the intrusion detection and response system in MANETs should be both distributed and cooperative. They proposed an intrusion detection approach for anomaly detection in routing updates, on the both MAC and in the application layer.

Kachirski and Guha (Kachirski, Guha, 2002) proposed a multi-sensor intrusion detection system based on mobile agent technology and clustering approach. In this system each mobile agent performing a certain functionality: monitoring, decision-making or initiating a response. The results of each node are aggregated in cluster points in order to limit the packet monitoring task in a few nodes and minimize the IDS-related processing time by each node.

Huang and Lee (Huang, Lee, 2003) presented a cluster-based cooperative intrusion detection system similar to Kachirski and Guha's IDS. In this approach, an IDS is not only able to detect an intrusion, but also to identify the source of the attack, if the identified attack occurs within one hop, whenever possible, through statistical anomaly detection.

Albers et al.(Albers et al., 2002) suggested an architecture for local detection based on mobile agents. In this proposed architecture, the intrusion detection agent running at each node for local concern which can be extended for global concern by cooperating with other local agents. Two types of data are exchanged among local agents: security data (to obtain complementary information from collaborating nodes) and intrusion alerts (to inform other local detected intrusion).

Sterne et al. (Sterne et al., 2005) proposed a dynamic intrusion detection hierarchy that is potentially scalable to large networks with using clustering. This method is similar with Kachirski and Guha (Kachirski, Guha, 2002), but it can be structured in more than two levels. Thus, nodes on first level are cluster-heads and nodes on the second level are *leaf nodes*. In this model, every node has the task of monitoring, logging, analyzing, properly responding to intrusions detection if there is enough evidence, and alert or report to cluster-heads. The cluster-heads, in addition, must also perform:1) data fusion/integration and data filtering, 2) computations of intrusion, and 3) security management.

Sun et al.(Sun et al., 2003) have proposed an anomaly-based two-level non-overlapping Zone-Based Intrusion Detection System (ZBIDS). In this architecture, inter-zone (gateway) nodes are responsible for collecting and aggregating the reports and alerts from intra-zone nodes. Inter-zone node in neighboring zone can collaborate to perform the intrusion detection in the wide area and to attempt to increase the detection rate and reduce the false positive alarm.

3. Our Proposed Approach

3.1 Network Model and Assumptions

The entire network is divided into multiple clusters. Nodes may be partitioned into clusters with one cluster-head for each cluster. Each cluster-head node is aware of its cluster information. The authenticity of a node is mostly determined by the nodes that are in same cluster. A cluster may split into sub-clusters and multiple clusters may merge into a super cluster. Data packets may traverse between different clusters. Thus, there are two types of data flows in terms of clusters: inbound data and outbound data. Thus, the process of detecting misbehaving nodes can be divided into two steps. First, it needs to determine whether a cluster has a misbehaving node. Should a misbehaving node exist in the cluster then the second step is to locate it. A node will be dismissed from network if it is misbehaving. Also, we have considered a hierarchical structure inside every cluster. The cluster-head is in the first level, the gateway nodes are in the second level and finally the leaf nodes or ordinary ones are in the third level of this hierarchy.

We assume that transmission is always atomic in terms of packets - a packet is either transmitted completely, or not at all. Finally, we define a bucket term for a specific count of packets (Kumar, 2009).

3.2Threat Model

Since routing protocols are the cornerstone of MANETs, in this paper we will focus on the detection of attacks targeted at MANET routing protocols, more specifically on detecting some of the most important active attacks: nodes misbehaving and anomalies in packet forwarding, such as intermediate nodes dropping or modifying packets. We use Dynamic Source Routing (DSR) protocol as the exemplary routing protocol to model the behavior of the routing anomalies attacks.

3.2.1 DSR

The key feature of DSR (Das et al., 2001)(Broch et al., 1998) is the use of *source routing*. In source routing, when a node originates a data packet, it puts in the *header* of the packet, all the hops that the packet needs to traverse to get to the destination. These routes are stored in a *route cache*. When a node needs a new route to a destination, it initiates the *route discovery* process by sending a route request (RREQ) message. The route request is broadcast by the originator and contains the address of the originator and the destination. Each node receives a RREQ, rebroadcasts it, unless it is the destination or it has a fresh route to the destination in its route cache. Such a node replies to the RREQ with a route reply (RREP) packet that is routed back to the original source. RREQ and RREP packets are also source routed. The RREQ builds up the path traversed so far. The RREP route itself back to the source by traversing this path backwards. The route carried back by the RREP packet is cached at the source for future use.

If any link on a source route is broken, the source node is notified using a route error (RERR) packet. The source removes any route using this link from its cache. A new route discovery process must be initiated by the source, if this route is still needed.

3.3 Quantitative and Agent based IDS (QAIDS)

In this subsection, we detail our proposed intrusion detection system - QAIDS. From the system aspect, we attach an IDS agent to each mobile node. These IDS agents run independently and monitor local activities to detect abnormal behaviors. We choose to implement a quantitative anomaly-based detection algorithm because it is expected that more types of attacks will be launched against MANETs in the future.

3.3.1 QAIDS Framework

We adopt a cluster-based intrusion detection framework because of the following considerations:

- Since the nature of MANET is dynamic and the mobility of nodes, the purview of every node is limited and the information and instances gathered are not enough. By clustering the nodes and determining a cluster-head node, it will be possible to do the task of information gathering in a vast domain and with a high accuracy and also to manage the alarms more efficiency.
- By organizing the nodes in a hierarchical structure and appointing the responsibilities to the nodes in

harmony with the role of every node, it will be possible to prevent the production of extra load and parallel actions. In this way the consumption of resources (like energy,…) will be reduced.

- The cluster-head nodes are suitable chokepoints to be placed the main tasks of an IDS (as router or gateway in wired networks).

The process of intrusion detection is composed of two major steps: first, detection of misbehaving nodes, second their punishment and ostracism from the network. The QAIDS is composed of 4 modules: data collection, detection engine, voting, and intrusion response modules (figure 1).

- *Data Collection Module*

The main duty of the data collection module is to supervise the behaviors of the nodes for collecting related security data. Data collection module supervises the intermediate nodes' packets forwarding quality. We use the Data Transmission Quality (*DTQ*) function proposed by Tao Li et al (Li et al., 2008) to measure a node's communication quality.

In this function, for every packet that node receives, it replies with an Ack packet to inform the sender that the packet is delivered safely or not. By investigating the received Ack packets, the transmitter nodes obtain information about the transmission quality of intermediate nodes which are located in the cluster's communication path segment. In other words, the senders get information on misbehaving and well behaving nodes. We use this function to measure the quality of intermediate nodes' behaviors in the process of packets forwarding. Each node has a DTQ table in which it keeps the quality of intermediate nodes' behaviors in the process of packet forwarding.

In order to decrease the amount of communication overhead due to Ack and statistical packets, we have used the concept of bucket. As it was mentioned, every bucket is composed of some specified packets. Instead of sending an Ack or statistical packets for every data packet, these packets are sent for each bucket. We will have two kinds of buckets: first the long-term or historical bucket which has the statistic information on the quality of packet forwarding of the last N packets; and second the short-term or the data bucket that includes the statistic data on the recently sent M packets. We have defined M and N in a way that N could be dividable into M, so N packets are divided into M short-term buckets. Each short-term bucket has the statistic related to the sent N/M packets. We call this statistic, the STability of the nodal Behavior (STB).

At the end of each short-term bucket, the data collected by the data collection module of the gateway node or the destination node is sent to the data collection module of the cluster-head node. The cluster-head node based on this statistic, updates the DTQ amount of all nodes in the communication path segment of its cluster in its DTQ table. The cluster-head also sends this statistic data to the nodes which are in the communication path segment. The data collection module of these nodes updates its DTQ table, based on the received data and the amount of its D and E then it sends this table to the detection engine.

The DTQ is defined as a function of STB(), probability of error in the channel(*P()*), and the energy needed to transmit data (*E*) as follows.

$$DTQ = K \times \frac{D \times STB()}{E \times P()} \tag{1}$$

The STB() function is defined as:

$$STB = \left[\frac{\Sigma(d_i, u_i)}{\Sigma(d_i, u_i)}\right]^{\alpha} = \left(\frac{\sum_{j=0}^{M} \left(\frac{d_j}{u_j}\right)}{\sum_{i=0}^{N'} \left(\frac{d_i}{u_i}\right)}\right)^{\alpha} \tag{2}$$

Where d_i and u_i represent the bytes successfully transmitted and the bytes attempted to be transmitted, respectively, when sending the past *ith* short-term bucket.

But since in our proposed IDS, the sending of a packet is in the atomic form and the size of a packet is constant, we will have STB as (Kumar, 2009):

$$STB() = \left(\frac{TotalACKedpacketsforthelast\ \frac{N}{M}packets}{TotalACKedpackets\ forthelastNpackets}\right)^{\alpha} \tag{3}$$

Finally, we have the *DTQ* function as follows:

$$DTQ = k \times \frac{D}{F} \times \frac{1}{PQ} \times \left(\frac{TotalACKedpacketsforthelast\frac{N}{M}packets}{TotalACKedpacketsforthelastNpackets} \right)^{\alpha} \quad (4)$$

Where k> 0 and α >1.

- *Detection Engine*

The task work of the detection engine is to detect the misbehaving nodes. The detection of misbehaving nodes requires an exact definition of the term misbehaving. It also requires specifying an appropriate threshold between normal and abnormal acts. In this article, we have defined the misbehaving nodes as the nodes which had abnormality in the process of packets forwarding. Since we have used cluster structure with a cluster-head for each of them, each cluster-head determines a threshold for its cluster as follow (equation (5)). Cluster-head updates this threshold based on the amounts of DTQ of all its nodes at the end of each short-term bucket then it sends this threshold to all its nodes.

$$Th = \tau \times \frac{1}{|N_{DTQ}|} \sum_{i \in N_{DTQ}} Q_i \quad (5)$$

where 0<τ<1.

In this equation, N_{DTQ} shows all listed nodes in the DTQ table of the cluster-head node, $|N_{DTQ}|$ is the number of nodes, Q_i is the DTQ value for node *i*.

If the detection engine finds one or some values of DTQ in the table that are less than the threshold, then it realizes that there may be one or some misbehaving nodes in its cluster. So, it sends to the voting module a vote request about the suspect nodes. According to the result of voting for the suspect node's authenticity, a decision is taken. To prevent a misbehaving node, starting false voting requests continuously, a node may only start another voting request after $2^{(N/M)-1}$ times.

- *Voting Module*

The operation of voting module depends on its node type. If a node is a leaf or gateway one, it only sends the alarms (vote-request) and reports generated by detection engine to the cluster-head node. While if the node is a cluster-head node, then the voting module handles the received alarms and reports from leaf and gateway nodes. Also, voting module of the cluster-head node allows the voting or prevents it by aggregation and correlation of the received alarms and reports along with its own information, and also by considering the privilege of the suspect and the node requesting for voting. If the cluster-head node agrees with voting, then it announces the process of voting by broadcasting a packet called Vote Request Packet to all its cluster nodes.

When the voting module of each leaf or gateway node receives the vote request packet, votes for or vetoes the suspect node according to the results announced by the detection engine and send result to the voting module of its cluster-head node. In the process of voting, simply accounting the positive or negative votes is not fair, because the values of each node's DTQ are not the same throughout a cluster. The values with recent timestamp are more important than the values which are older. We have considered this fact by considering the *w* variant as the time weight. Let's assume that *k* nodes participate in the process of voting for or against the authenticity of the node *m*. The voting module of the cluster-head node calculates the result of voting according to bellow.

$$V_m = \frac{k}{N} \times \sum_{i=1}^{k} w_{im} Q_{im} v_{im} \quad (6)$$

In this equation, Q_{im} shows the DTQ value of the node *m* in the DTQ table of node *I* and *N* shows the total number of nodes which are located within the cluster. w_{im} shows the time weight of Q_{im}. v_{im}=1, if node *i* votes for node *m*, and v_{im}=-1, if node *i* votes against node *m*.

At the end of voting process, voting module of the cluster-head node sends the result of voting (V_m) to its intrusion response module and voting modules of all its cluster nodes.

- *Intrusion Response Module*

According to the results of voting, the *m* node is a well-behaving one and is acquitted or it is a misbehaving one and

should be punished. If $V_m<<0$, then intrusion response module finds that the node *m* is misbehaving and adds *m* to its blacklist. When a node is added to the blacklist, the intrusion response module deletes it from its DTQ and routing tables. Thereafter, the intrusion response module prevents the node from cooperating with the blacklisted node. In other words, not only no packet is sent by the misbehaving node but it is also prevented from forwarding its packets by other nodes. Besides, they deny the access of the misbehaving node to their routing and DTQ tables. In this way the misbehaving node is punished and ostracized from the network. Also, the intrusion response module of the cluster-head node decreases the DTQ amount of all nodes if there exists a misbehaving node in its communication path segment or an attack happens in its cluster.

If $V_m>>0$, then the intrusion response module realizes that the node *m* is a well behaving one, and those nodes that have voted against it, should update the amount of their DTQ. Let's assume that there are *K* nodes in the communication path segment, then the nodes that have voted against *m*, update the amount of their DTQ as follow:

$$Q_m = \frac{1}{K}\sum_{i=1}^{K} Q_i \tag{7}$$

In this equation, Q_i stands for the amount of the DTQ related to the node *i* in the communication path segment inside the cluster. This equation decreases the difference among the voting nodes and prevents from repeated requests for voting in near future.

If the value of $V_m \cong 0$, then the process of voting is repeated in the clusters level. In this situation, the cluster-head node carries out the process of the voting in a vast area by sending vote request to the cluster-head nodes of the downstream and upstream clusters (if there is any).

There is a module called pathrater inside the intrusion response module that prevents the routes that have misbehaving nodes from being selected. Since we are using the DSR protocol as the routing protocol and according to what was mentioned before, this is a source routing protocol. So the route discovery operation is done by the source node and all discovered routes are stored in its route cache and then a route is selected among them. Based on the following equation and according to the nodes of every route, the pathrater evaluates the rating of all cached routes and then selects the best route.

$$R= \frac{1}{|R|}\sum_{i=1}^{|R|} w_i Q_i \tag{8}$$

In this equation, |R| stands for the length of the route or the number of the nodes making the route R, Q_i shows the DTQ amount of the node *i* on the route R in the DTQ table of the source node and w_i stands for the time weight of Q_i. According to this equation, if the route is short and the related nodes are well behaving, the route will have the highest rating.

3.4 Example of QAIDS Operation

In this section, by giving an example, we elaborate on the quality of QAIDS operation. Based on the cluster structure, we consider two communication scenarios, the intra-cluster scenario and the inter-cluster scenario. In the first scenario, both the receiver and the sender belong to the same cluster.

In the second scenario, the sender and receiver belong to two different clusters (figure 2). In intra-cluster communication, the module for collecting the information of the destination node sends the statistic of Ack packets of the nodes in the communication path segment to its cluster-head node at the end of every short-term bucket. In inter-cluster communication, the gateway nodes play the main role. In this communication, the gateway node sends the statistic of the nodes located along the communication path segment inside its cluster to its cluster-head node, at the end of every short-term bucket. Also prevents modified packets from being forwarded, and finally, the destination node sends this statistic to its cluster-head node at the destination cluster. As an example in figure 2, let's assume that the node S in the cluster C_1 is determined to send some packets to the node D in the cluster C_2. The communication paths are S→A→M→G_1→G_2→B→D. now consider the following scenario:

- The node S sends a short-term bucket (on the assumption of 5 packets) to the node A.
- The node A forwards these packets to the node M.
- The node M is a misbehaving node and modifies some packets (take 2) and sends them to the gateway node G_1.

- The node G_1 makes sure about the safe or unsafe delivery of the packets, by investigating the received packets at the end of the short-term bucket (by using checksum techniques). Then it sends this statistic to the cluster-head node C_1, and prevents the modified packets from being forwarded to the gateway node G_2, so that it can save the resources like energy and memory of the intermediate nodes and the bandwidth of the network.
- By receiving this information, the cluster-head node C_1 is informed about the misbehaving of at least one of the nodes in the communication path segment inside its cluster. Then it decreases the DTQ value of all these nodes (A and M).
- In C_2 cluster, the gateway node G_2 forwards the packets to the node B, and the node B forwards them to D (the final destination). The node D by investigating the received packets at the end of the short-term bucket sends the statistic related to them to the cluster-head node C_2. By receiving this statistic, the node C_2 is informed about the misbehaving or well behaving of its cluster nodes.

4. Simulation Results and Analysis

GloMoSim 2.03 simulator is used to simulate our model. We conducted our experiment using Dynamic Source Routing (DSR) protocol as the routing protocol.

4.1 Simulation Environment

The channel capacity of mobile hosts is set to 2 Mbps and the transmission range is set to 250 meters. A free space propagation model with a threshold cutoff is used as the channel model. We use the Distributed Coordination Function (DCF) of IEEE 802.11 for wireless LANs as the MAC layer protocol. In the simulation, mobile nodes move in a 2000 * 2000 meter quadrangular region. The mobility model is the random waypoint model. The minimal speed is 5 m/s, and the maximal speed is 15 m/s. Various source-destination pairs are selected randomly to generate Constant Bit Rate (CBR) traffic as the background traffic. The size of all data packets is set to 512 bytes. The duration of each simulation was 1800 seconds.

4.2 Simulated Attacks

In this article, we have simulated three flooding, black-hole and denial-of-Service (DoS) attacks to evaluate the functionality of our QAIDS.

- Flooding attack (Hongsong et al., 2007): In this attack, the misbehaving node pumps a great deal of useless and garbage packets to the network. In this way it corrodes the resources of the network (like bandwidth and energy).
- Black-hole attack (Anjum, Mouchtaris, 2007): In this attack, a misbehaving node uses the routing protocol to advertise itself as having the shortest path to the node whose packets it wants to intercept. The attacker will then receive the traffic destined for other nodes, then can drop or modify the packets.
- Denial of Service (DoS) Attack (Rong, Çayırcı, 2009): In this attack, the misbehaving node prevents other nodes from cooperating, by depleting the resources of the nodes and the network (like energy and bandwidth). When the resources of the nodes (specially the energy) are depleted and reach a specific threshold, the node prevents itself from cooperating with other nodes so that it can increase its life-time.

4.3 Performance Metrics

- Detection Ratio: It is defined as the percentage of IDS capability in detecting the misbehaving nodes. And is resulted from dividing the accurate detections into all detections.
- False Positive Ratio: It is defined as the percentage of decisions in which well behaving nodes are flagged as misbehaving ones inaccurately.

4.4 Simulation Results

In the first simulation, we have considered the relation between the detection rate and number of the nodes. As it is showed in figure 3, most of the attacks have been detected successfully. In detecting three attacks, back-hole, DoS and the flooding attack, QAIDS have had a good stability, and its detection rate has been over 85%. But, detection rate of flooding and DoS attacks has decreased in high density networks. This was predictable, because in high densities, the amount of the traffic is high and it is not possible to distinguish between the network's normal status and it's under attack situation easily. So it is not possible to distinguish between the normal and under flooding attack situation of the network. Also, in DoS attack, because of high rate of the generated packets, some packets may not reach the destination due to congestion or link destruction.

The result of the simulation between false positive and the number of nodes is shown in figure 4. In its worst status, the false positive rate has not been over 14%, this is a good rate. In networks with high densities, the false

positive rate is high and this is so natural. As an example, in networks with high densities, the amount of the generated traffic is high. So it is not possible to identify whether the high rate of traffic is due to flooding attack or then normal situation of the network. In this condition, the sending nodes whose packets have been dropped because of reasons other than attack, have to resend them. So the amount of DTQ function of these nodes decreases and they are considered as misbehaving nodes inaccurately.

Next, we have investigated the effect of percentage of misbehaving nodes, short-term and long-term bucket size as parameters on the performance of QAIDS. The number of nodes in the remainder simulations is 50.

In figure 5, we have shown the result of the simulation between the detection rate and the percentage of misbehaving nodes. As it is seen, most of the misbehaving nodes have been detected successfully. But as it was predicted, in networks with high percentages of misbehaving nodes, the detection rate decreases. This happens due to different reasons and the most important of all, is related to the process of voting. Because, when the percentage of misbehaving nodes is high, the accuracy of voting result will be reduced.

The result of the simulation of the relation between false positive rate and the percentage of the misbehaving nodes is shown in figure 6. As the result of the simulation shows, when at first the percentage of the misbehaving nodes is low, the false positive rate is high. This was predictable, because in normal situation, some packets may not reach the destination due to reasons like congestion or link destruction. This situation is mistaken for misbehaving of the node. But when the percentage of the misbehaving nodes is high, the false positive rate again increases; because as we mentioned in the previous paragraph, under this situation the exactness of the voting process decreases.

In the next simulation, we discussed the relation between detection rate and short-term bucket size. As it shows in figure 7, the best result is related to small-size buckets of 5. Because, nodes update the *DTQ* table and send the reports and request for voting at the end of every short-term bucket, this operation is done more times and in short time intervals in small short-term buckets. So, in small-size buckets the rate of detection is very high.

Figure 8 shows the result of simulation between false positive and short-term bucket size. As you see, the rate of false positive is high at the first and is very close in the size of 10 and 15 and the rate of false positive is in its best status. In bigger buckets there is neither improvement nor better false positive, it has even got worse. These results were predictable in the equation 4. Because, in small buckets, the recent behavior of the node greatly affects the node's DTQ, and the node reacts fast. As an example, if the size of the short-term bucket is 5 packets, and if two of these 5 packets do not reach the destination due to congestion or destruction of the link, the node reacts and sends alarm pronto. Because, 40% of the sent packets have been dropped during the recent short-term bucket. In bigger buckets, some packets, for example, three packets may be dropped because of misbehaving. But as far as this amount is ignorable in comparison with length of bigger-size buckets (e.g 20), in bigger buckets, the rate of false positive is high.

The result of simulation between detection rate and long-term bucket size is shown in figure 9. As the figure shows, the detection rate at the first is low. When long-term bucket size is small, the purview of each node about the history of other nodes' behavior is decreases. If the long-term bucket size is small and is close to the short-term bucket size, then the behavior of the node in the process of forwarding the recent short-term bucket will be similar to each other in comparison with long-term bucket size and it will not yield accurate and exact information.

The result of the simulation of the relation between long-term bucket size and false positive rate is shown in figure 10. As it was predictable, the rate of the false positive is high in small long-term buckets; because in smaller buckets, the purview of each node decreases about the behavior history of other nodes. Due to the nodes' good behaviors and especially ill behaviors affect their DTQ's in the process of forwarding the recent *M* packets.

In very large long-term buckets, the rate of the false positive is very high. This fact is also predictable, because the topological situation of the network is changing due to the mobility of the nodes. The nodes may have been in a situation that there has been utmost link among them. So the rate of the lost packets is low due to disconnection of the links. But because of mobility of the nodes, they may have been in a new situation that they have had the least links among nodes due to long distance of the nodes or some obstacles. In this way the rate of the lost packets due to disconnection of the links will be high. Because of this, considering the nodes' very long-term behavior history in the process of rating the nodes will increase the false positive rate.

As we observed, a change in the parameters like short-term and long-term bucket sizes will highly affect the performance of QAIDS. According to the recent 4 simulations, the best performance of QAIDS is acquired according to the detection and false positive rate, if the short-term bucket size is considered 10 and long-term

one 30 packets. This status is shown in the first 4 simulations.

5. Conclusion and Future Work

This paper presents and evaluates a quantitative anomaly-based detection algorithm used in QAIDS. Using the routing attacks as the threat model and DSR protocol as routing protocol, we have carried out simulation and demonstrated its effectiveness. One of our future works is to develop and present new mechanisms to aggregate and fusion security-related information to achieve good false positive ratios. We also plan to investigate more attack scenarios and other routing protocols in MANETs, not only at the routing layer, but also at other layers.

References

Albers P., Camp O., et al. (2002). 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)*, 1-12.

Anantvalee T. and Wu J. (2007). A Survey on Intrusion Detection in Mobile Ad Hoc Networks, *Book Series Wireless Network Security, Springer*, ISBN: 978-0-387-28040-0 , 170-196.

Anjum F. and Mouchtaris P. (2007). The Handbook of Security for Wireless Ad Hoc Networks (Chapter 1), CRC Press LLC.

Broch J., Johnson D. and Maltz D. (1998). The dynamic source routing protocol for mobile ad hoc networks, http://www.ietf.org/internet-drafts/ draft-ietf-manet-dsr-01.txt, IETF Internet Draft (work in progress).

Das S. R., C. E. Perkins C. E. and Royer E. M. (2001). Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks, Mobile Ad Hoc Networking Working Group, Internet Draft, February.

Hongsong C., Zhongchuan F., Chengyao W., Zhenzhou J. and Mingzeng H. (2007). Using Network Processor to Establish Security Agent for AODV Routing Protocol, *Journal of Computing and Information Technology* - CIT 15, 61–70.

Huang Y., and Lee W., (2003). A Cooperative Intrusion Detection System for Ad Hoc Networks, *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, 135-147.

Kachirski O. and Guha R. (2002). Intrusion Detection Using Mobile agents in wireless Ad hoc Networks, *Proceedings of the IEEE workshop on Knowledge Media Networking*, 153-158.

Kumar K. (2009). Intrusion Detection in Mobile Ad hoc Networks, Master's Thesis, The University of Toledo.

Li T., Song M. and Alam M. (2008). Compromised sensor node detection: A quantitative approach, *Proceedings of the IEEE International Conference on Distributed Computing Systems*, 352–357.

Mandala S., Ngadi M. A. and Abdullah A. H. (2010). A Survey on MANET Intrusion Detection, *International Journal of Computer Science and Security*, 2 (1).

Mishra A., Nadkarni K. and Patcha A. (2004). Intrusion Detection in Wireless Ad Hoc Networks, *IEEE Wireless Communications*, IEEE press, 48-60.

Mitrokotsa A., Mavropodi R. and Douligeris C. (2006). Intrusion Detection of Packet Dropping Attacks in Mobile Ad Hoc Networks, *Proceedings of the International Conference on Intelligent Systems And Computing: Theory And Applications*, Ayia Napa, Cyprus 111-118.

Otrok H., Debbabi M., Assi C. and Bhattacharya P. (2007). A Cooperative Approach for Analyzing Intrusions in Mobile Ad hoc Networks, *Proceedings of the 27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07).*

Rafsanjani M. K., Movaghar A. and Koroupi F. (2008). Investigating Intrusion Detection Systems in MANET and Comparing IDSs for Detecting Misbehaving Nodes, *Proceedings of the World Academy of Science, Engineering and Technology,* 44.

Sahu S. and Shandilya S. K. (2007). A Comprehensive Survey on Intrusion Detection in MANET, *Proceedings of the International Journal of Information Technology and Knowledge Management*, 2(2), 305-310.

Rong C. and Çayırcı E. (2009). Security Attacks in Ad Hoc, Sensor and Mesh Networks, *in Book Security in Wireless Ad Hoc and Sensor Networks (Chapter 8)*, CRC Press LLC.

Sterne D., Balasubramanyam P. and et al. (2005). A General Cooperative Intrusion Detection Architecture for MANETs, *Proceedings of the 3rd IEEE International Workshop on Information Assurance (IWIA'05)*, 57-70.

Sun B., Wu K, and Pooch U. W. (2003). Alert Aggregation in Mobile Ad Hoc Networks, *Proceedings of the 2003 ACM Workshop on Wireless Security (WiSe'03) in conjunction with the 9th Annual International Conference on Mobile Computing and Networking (MobiCom'03)*, 69-78.

Yi P., Dai Z., Zhang S. and Zhong Y. (2005). A New Routing Attack in Mobile Ad Hoc Networks. *Proceedings of the International Journal of Information Technology*, 11(2).

Zhang y., Lee W. and Huang Y. (2003). Intrusion Detection Techniques for Mobile Wireless Networks, *Wireless Networks*, 9, 545-556.

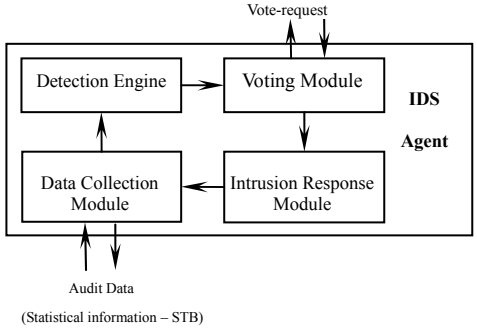

Figure 1. Structure of an IDS agent

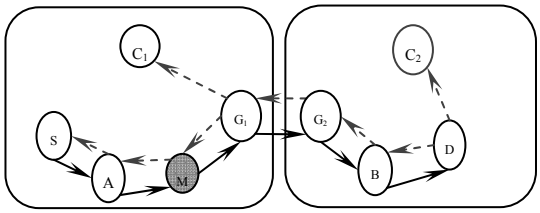

Figure 2. Inter-cluster communication scenario

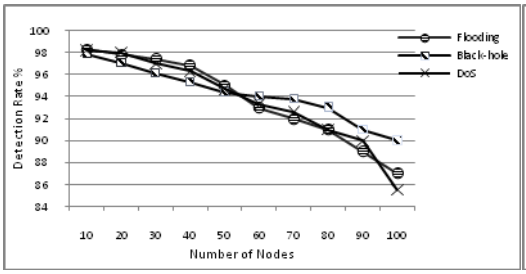

Figure 3. Detection rate vs. number of nodes

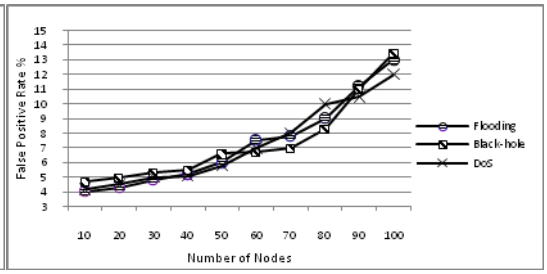

Figure 4. False positive rate vs. number of nodes

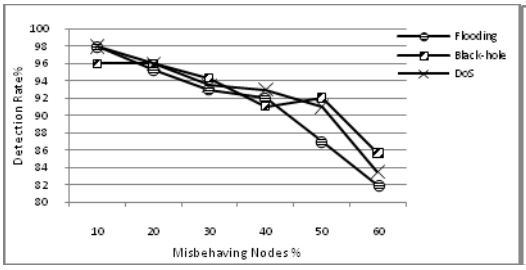

Figure 5. Detection rate vs. percentage of misbehaving nodes

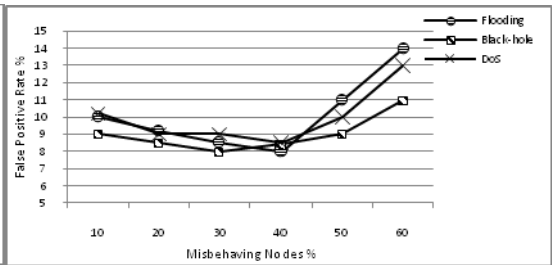

Figure 6. False positive rate vs. percentage of misbehaving nodes

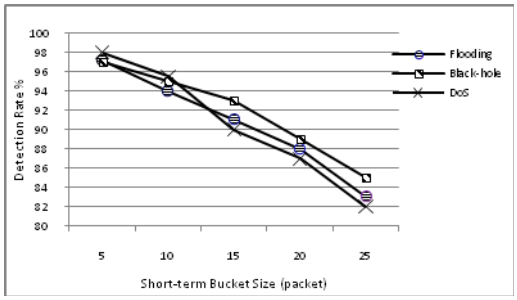

Figure 7. Detection rate vs. Short-term bucket size

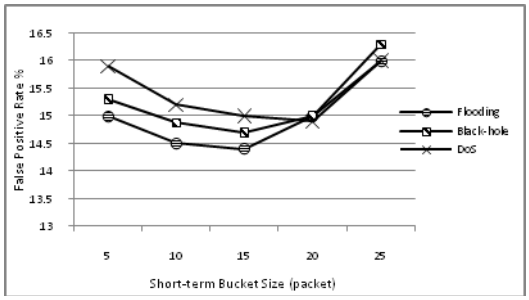

Figure 8. False positive rate vs. Short-term bucket size

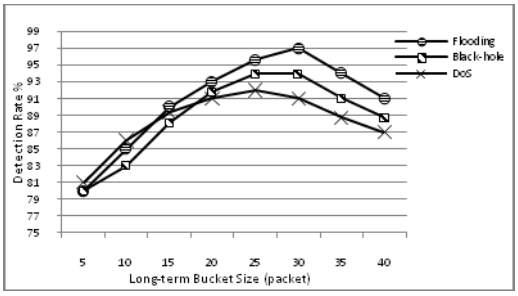

Figure 9. Detection rate vs. Long-term bucket size

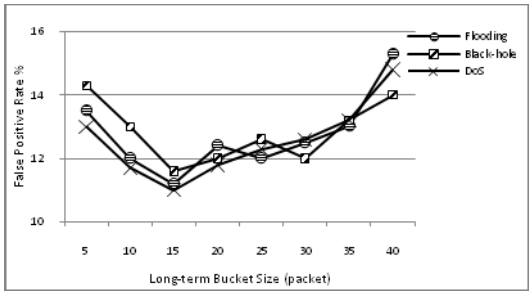

Figure 10. False positive rate vs. Long-term bucket size

 ISSN 1913-8989 E-ISSN 1913-8997