Intrusion Detection Technique in Mobile Adhoc Network based on Quantitative Approach

Saroj Hirnwal
Professor
Balaji College of Engineering Technology, Jaipur (Rajasthan)

Kirti Chauhan
M.Tech Scholar
Balaji College of Engineering Technology, Jaipur (Rajasthan)

Amit Gupta
Asst. Professor
Balaji College of Engineering Technology, Jaipur (Rajasthan)

ABSTRACT
In this paper ,we study the types of attacks in intrusion detection system in Mobile Ad Hoc network(MANET).Mobile Adhoc Networks are a relatively new and rapidly evolving area of interests. One such field concerns mobile adhoc networks (MANETs) in which mobile nodes organize themselves in a network without the help of any predefined infrastructure. Securing MANETs is an important part of deploying and utilizing them, since they are often used in critical applications where data and communications integrity in important. Many solution for intrusion detection in wireless environments have been developed but these solution may not always be sufficient, as ad-hoc networks have their own vulnerabilities that cannot be addressed by these solutions. In this paper traditional security algorithms coupled with intrusion detection mechanism. Here we using a quantitative method to detect intrusion in MANETS with mobile nodes. Our method is a behavioral anomaly based system, which makes it dynamic, scalable, configurable and robust. For simulating our mobile nodes use AODV (Adhoc on demand distance Vector)routing. It is observed that the malicious node detection rate is very good and false positive detection rate is slow.

Keywords
MANET, Intrusion Detection System, Behavioral Anomaly Based System.

1. INTRODUCTION
A mobile ad hoc network (MANET) is a collection of mobile computers or devices that cooperatively communicate with each other without any pre-established infrastructures such as a centralized access point. Computing nodes in an ad hoc network act as routers to deliver messages between nodes that are not within their wireless communication range. Because of this unique capability, mobile ad hoc networks are envisioned in many critical applications (e.g., in battlefields). Therefore, these critical ad hoc networks should be sufficiently protected to achieve confidentiality, integrity, and availability. Manet have dynamic and cooperative nature so here challenges in securing these networks. In wired networks which have a higher level of security for gateways and routers, ad hoc networks have the characteristics such as dynamically changing topology, weak physical protection of nodes, the absence of centralized administration, and highly dependence on inherent node cooperation. Topology always changing, and these networks do not have a well-defined boundary, and thus, network-based access control mechanisms like firewalls are not directly applicable. In addition, there is no centralized administration, making bootstrapping of crypto systems very difficult. It is extremely easy for a malicious node to bring down the whole network. As a result, ad hoc networks are vulnerable to various attacks including eavesdropping, spoofing, modification of packets and distributed denial-of-service (DDoS) attacks.

Intrusion detection involves the runtime gathering of data from system operation, and the subsequent analysis of the data; the data can be audit logs generated by an operating system or packets "sniffed" from a network. We limit our focus to intrusion detection based on behavior, we think it is a more efficient, lightweight and easily scalable solution to Intrusion Detection in MANETs. Intrusion Detection Systems based on behavior can be broadly classified into these categories: anomaly detection, signature or misuse detection, and specification based detection. In signature-based intrusion detection [5][10], the data is matched against known attack characteristics. In anomaly detection find out the normal behavior of systems, usually established through automated training, are compared with the actual activity of the system to flag any significant deviation. In specification-based detection [7][8], the correct behaviors of critical objects are manually abstracted and crafted as security specifications, which are compared with the actual behavior of the objects. This paper describes intrusion detection for mobile ad hoc networks. we employ Behavioral-based techniques to monitor the ad hoc on-demand distance vector (AODV) routing protocol, a widely adopted ad hoc routing protocol. AODV is a reactive and stateless routing protocol that establishes routes only as desired by the source node. AODV is vulnerable to various kinds of attacks. This paper analyzes some of the vulnerabilities, specifically discussing attacks against AODV that manipulate the routing messages. We propose a solution based on the behavioral based intrusion detection technique to detect attacks on AODV. Briefly, our approach involves the use of simulator.

2. RELATED WORK
A number of IDS techniques have been proposed in the research literature. A number of trust building and cluster-based voting schemes have been proposed to enable the sharing and vetting of messages, and data, generated and gathered by IDS systems. Zhang and Lee describe a distributed and collaborative anomaly detection-based IDS for ad hoc networks [2, 3]. AODV routing behavior and distributed network monitors for detecting run-time violation of the specifications [4]. Pirzada and McDonald present a method for building confidence measures of route trustworthiness without a central trust authority. The authors also present a concise summary of previous work in the area of establishing trust in ad hoc networks [5]. Michiardi and Molva assign a value to the "reputation" of a node and use this information to identify misbehaving nodes and cooperate only with nodes with trusted reputations [6]. Albers and Camp

couple a trust-based mechanism with a mobile agent based intrusion detection system, but do not discuss the security implications or overhead needed to secure the network and individual nodes from the mobile agents themselves [7].Gateway nodes in neighboring zones can then further collaborate to perform intrusion detection tasks in a wider area and to attempt to reduce false positive alarms [8].

2.1 Network Simulator

Simulation is the imitation of some real thing, state of affairs, or process. The act of simulating something requires capturing the essential characteristics, activities and rules of a selected physical or abstract system. Simulations are often used to model natural, machine or human systems in order to gain insight into their functioning. In addition, it is a very important mechanism to understanding interactions between various systems, parts of which may be difficult to recreate or control in the real world. In technology, simulations are used widely for testing, performance optimization and measurement, safety engineering, training and education. Simulation is specifically necessary in the context of our research, since it involves wireless networks. It is almost impossible to reproduce the wireless propagation environment, difficult to use real radio-wave based transmission to test our concept, and non-trivial to use real wireless sensor/devices to present our research idea. Use of a simulator solves all these challenges. Simulators for communication networks can provide near accurate reproductions of most features in the environment, like noise, probability of loss or alteration of data. They often allow user implementations of protocols for transmission, propagation, reception or other communication aspects to work with their"ether". Thus, using simulators allows us to concentrate on the research idea instead of physical implementation details. In this paper for show the simulation process we use one kind of simulator Known as OMNet++. Short overview of OMNet++ describe below.

2.2 Overview of Simulator OMNet++

OMNeT++ is a component-based, modular and open-architecture discrete event simulation framework. The most common use of OMNeT++ is for simulation of computer networks, but it is also used for queuing network simulations and other areas as well. its primary application area is the simulation of communication networks, but because of its generic and flexible architecture, is successfully used in other areas like the simulation of complex IT systems, queuing networks or hardware architectures as well OMNeT++ provides a component architecture for models. Components (modules) are programmed in C++, and then assembled into larger components and models using a high-level language (NED). Reusability of models comes for free. OMNeT++ has extensive GUI support, and due to its modular architecture, the simulation kernel (and models) can be embedded easily into your application. OMNet++ is basically a collection of software tools and libraries which you can use to build your own simulation models. The simulation framework provides the following.

2.3 Components of OMNet++

- simulation kernel library
- compiler for the NED topology description language
- OMNeT++ IDE based on the Eclipse platform
- GUI for simulation execution, links into simulation executable (Tkenv)
- command-line user interface for simulation execution (Cmdenv)
- utilities (make file creation tool, etc.)
- documentation, sample simulations, etc.

2.4 Research Approach

The solution to our research challenge is presented here. It is based on the quantitative intrusion detection techniques in [9], and is applied to a MANET containing mobile nodes. The main questions were earlier classified under two areas. They are:

1. Detection of intrusion

2. Choice of simulator

   (a) Availability of mobility models

   (b) Availability of routing protocol implementation

2.5 Detection of Intrusion

The first level of moving toward a secure adhoc network consists of identification of nodes within the network that display unexpected behavior, or, in other words, may have turned malicious. Identifying malicious nodes consists of two steps. The first is the recognition of nodes that displaying malicious behavior, and the second is to ascertain whether that classification is correct.

A. Recognition

Detecting malicious nodes entails defining the term malicious. The scope of the current research allows definition of malicious nodes as those that have aberrations in data exchange patterns. Dr.Alam, Tao Li et al, in [9], proposes a method in which nodes are expected to acknowledge every message it receives. Every node measures the number of acknowledgments it has received from the neighbor nodes; it has tried to transmit to. In other words, each node records the throughput of every neighbor node it has attempted to communicate with. This value is a measure of near-term behavior. This behavior measured over a period of time determines the historical quality of behavior of the neighbor node. This statistic is the stability of the nodal behavior, and will henceforth be referred to as"STB ()". "Data transmission quality" (referred to as DTQ from now on) is defined as a function of STB (), probability of error in the channel(P()), and the energy needed to transmit data (E).

$$DTQ = \frac{K*(D*STB())}{E*P()}$$

D = Power needed for transmitting the total data attempted to be sent.

E = Energy needed to send 1 byte of data, and k is a constant.

The current research paper is limited to non-cluster based networks. Also, in the current research, transmission is always atomic in terms of packets - a packet is either transmitted completely, or not at all. We rely on measuring the effectiveness of transmission from a node to another. Each node calculates and maintains DTQ for each of its neighbor nodes. When DTQ value falls below a set threshold, the neighbor is signals as a malicious node.

B. Confirmation

The next step in the identification of such nodes is to ascertain whether a reading made by one node is correct. This is decided based on a group consensus approach every node in the network is sent a request to accept/reject this decision. Nodes receiving such a request can vote for or vote by referring to its own DTQ readings for the node in question. The vote initiating node then draws a consensus based on these replies. If more votes have been received approving of malicious behavior, the node is added into a Black-list that allows all nodes to refrain from further communication with this node.

2.6 Detection of malicious nodes

A. Recognition of malicious nodes

Recognition of a node displaying malicious behavior is a continuous process followed by each node. The process of malicious node recognition is detailed by the flowchart:

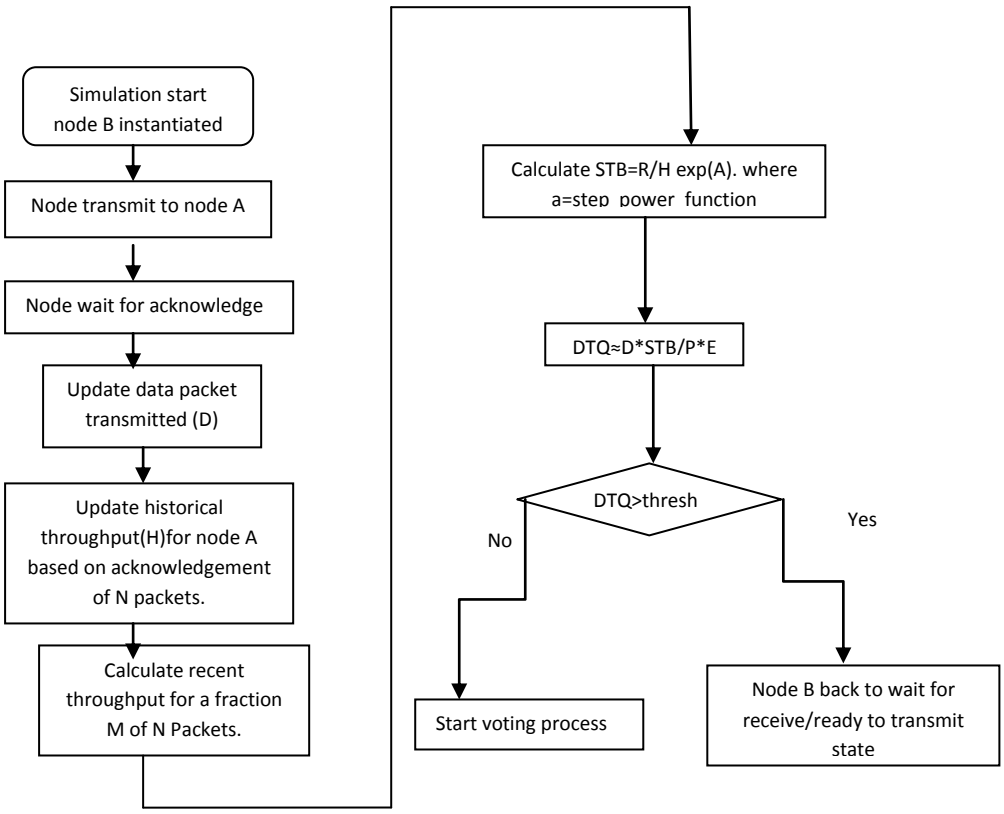

Fig 1: Malicious Node Recognition Flowchart

B. Confirmation of detection

The next step is that of collectively deciding whether a node whose behavior is erratic is actually a malicious one. For example, node A has detected that node B's DTQ has fallen below a threshold. Node A now wants consensus on its suspicion, and triggers a vote by sending a broadcast request for the same. When MANET nodes receive such a request, they check the DTQ values for node B in their tables, and reply with a positive or negative vote. These votes are aggregated at node A to decide node B's status. Voting process diagram given below.

3. VOTING DETAILS

Voting details include: Vote Arrival, Vote request timeout, Who vote, Process after vote decision.

Vote Arrival: A vote-initiating node count the number of votes in receives. It does not register more than one votes from the same neighbor. Once it has received votes from all of its neighbors, it decides for or against the voted-upon node.

Vote Request Timeout: The situation where some data packets may be lost and some node decide not to vote. In such cases, the vote-initiator cannot wait. The vote request time out solves this dilemma, and is set as soon as the vote-request is sent out. At the end of this time-out period, the vote request initiator aggregates all the votes it has received, and makes a decision based on the counts. All votes received after this timeout are useless.

Who vote: All nodes that receive a vote-request attempt to vote. However, if the number of messages they receive from the vote-initiator is not sufficient for them to decide, they refrain from voting. Process after vote decision: Two process are using: On Blacklisting, On being acquitted.

In on blacklisting process the number of nodes is in blacklisted never used for further communication. On being acquitted, if a node is acquitted after the vote decision, all nodes treat it as usual. now matter is that if any node is declared acquitted and have low DTQ value can take participate in vote request process but this process always shifted into bucket to bucket and give chance to every node.

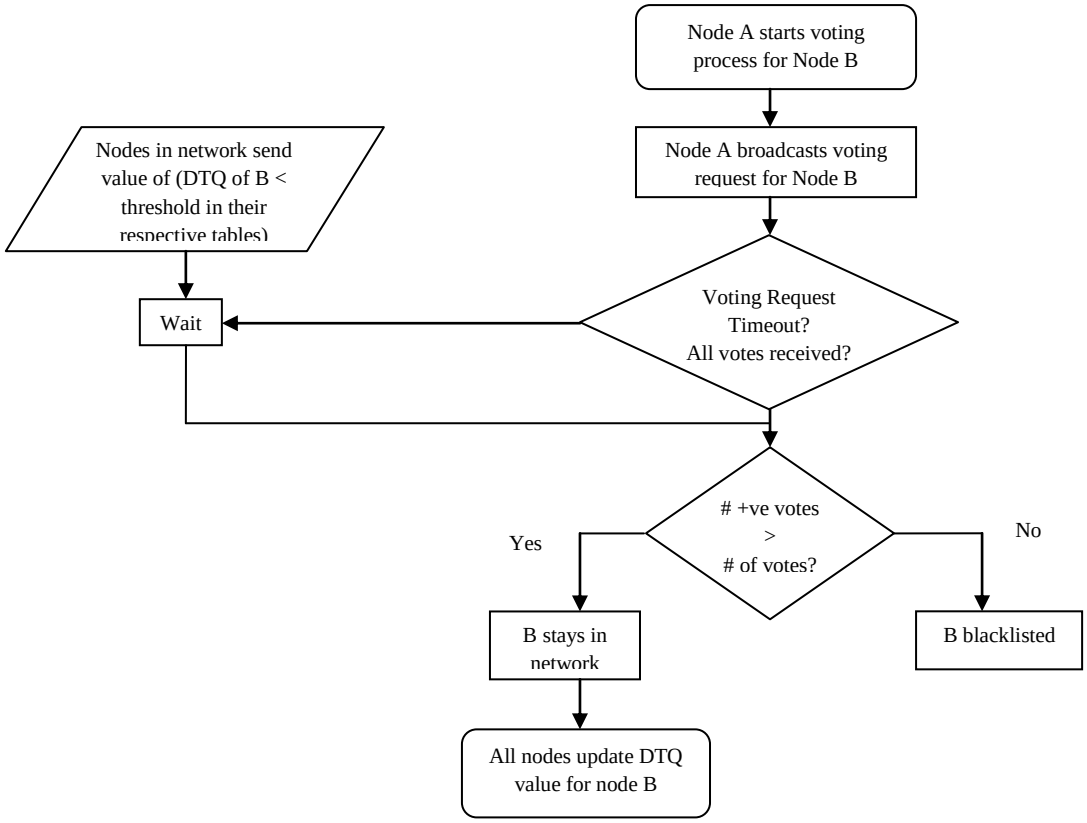

Fig 2: Malicious Node Detection Flowchart

4. ACKNOWLEDGEMENT FOR MESSAGES

Every node sends an acknowledgment of message receipt as soon as a it receives a data message. The sender waits for acknowledgment for some time.

A. Acknowledgement Arrival

If the acknowledgment arrives on time, the statistics for the acknowledgment sender are updated. If this is the end of a bucket, the DTQ is calculated anew, and a comparison for DTQ versus threshold is made. If necessary, a vote-request is scheduled.

B. Acknowledgement Timeout

The ACK-timeout is the time a sender A waits for an acknowledgment from the intended recipient, node B .If the acknowledgment does not arrive on time and if this is the end of a block, then, again, the DTQ is recalculated and the process of comparison repeats. Also, if the end of a block (bucket) is reached, the sender no longer accepts any more acknowledgments for this block of sent data, i.e. the DTQ for this block of data is final.

4.1 Implementation Details

Implementation details part include a discussion of the simulator packages used, modified or extended. The module defines AODV module, Physical layer module and MAC layer module.

A. Physical Layer Module

The Physical layer implements the physical layer of each host. It enables on-the-fly creation and deletion of connections among hosts. These connections are enabled via the creation of gates and they allow exchange of messages among the hosts. The check to update connections happens with every movement of a host, and if it is close enough (with respect to the transmission power) of some neighbor, a"gate" is created connecting their physical layers. Sometimes, these gates can be a simplex connection too, i.e. one node may be able to send to, but cannot receive from the other. Similar to creating gates is the process of stripping off the connection with a node which is no longer in range.

B. Medium Access Control (MAC) Layer Module

The MAC layer implemented is a very simple, non-standard one. Its main functions are to let outgoing messages pass through, and queue incoming messages on a M/M/1 queue On message arrival, the MAC module checks a flag to see if the higher level is busy. If busy, the message is queued is possible, else dropped if the buffer is full. When the higher level signals readiness, the the MAC module picks the first message in the buffer, sends it upward and schedule to itself an end of service message that will trigger a new pick from the buffer or set the busy-flag as free. This level filters out only those packets that are intended for this node by watching MAC Address of packets.

5. SIMULATION RESULT

A. Changes in mobility features based on Varying Speed

This section aims to measure the functioning of our IDS when changing features of mobility like speed and acceleration.

Table 1. Settings used for varying mobility feature tests

Number of nodes = 10
Simulation run time = 800 seconds
Mobility update Interval = 1 second
Malicious node count = 4
Malicious node Id = nodes 5,6,8,9
Acknowledgment timeout = 30 s
Initial speed = 5 m/s
History Count bucket = 20
Number of buckets = 2, i.e block bucket count = 10

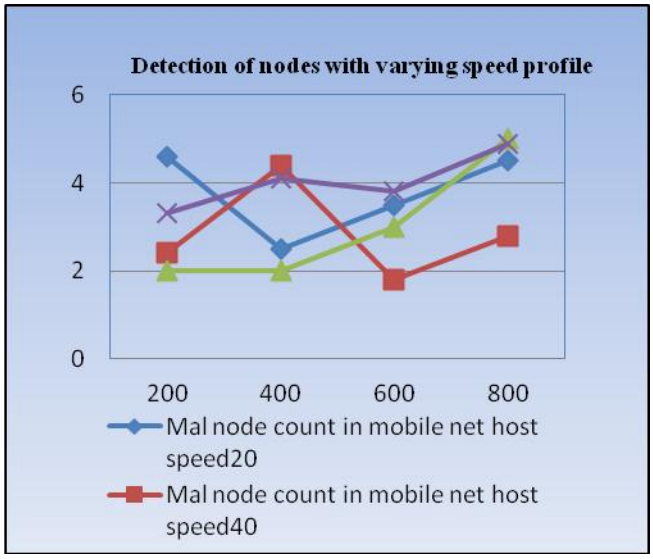

Fig 3: Graph showing the varying speed

Discussion

All malicious nodes are successfully detected and there is possibility of false positives. False positives can be explained by analyzing the sent/received/acknowledged message counts for various nodes using the output files. False positives occur due to one of the following reasons: There is two node say node A and B. A does not receive messages for an extended period from a node B. The sending node B evaluates the absence of acknowledgments from A as malicious behavior, even though A is a legal node. We have positively identified nodes based on their transmission characteristics, and can identify innocent nodes that have turned malicious after establishment of the network.

B. Result based on varying acceleration

Acceleration increases the speed at which nodes travel. This change in acceleration is applied once every"mobility interval update" seconds.

Where no of nodes 18
Simulation Run Time 1800 sec

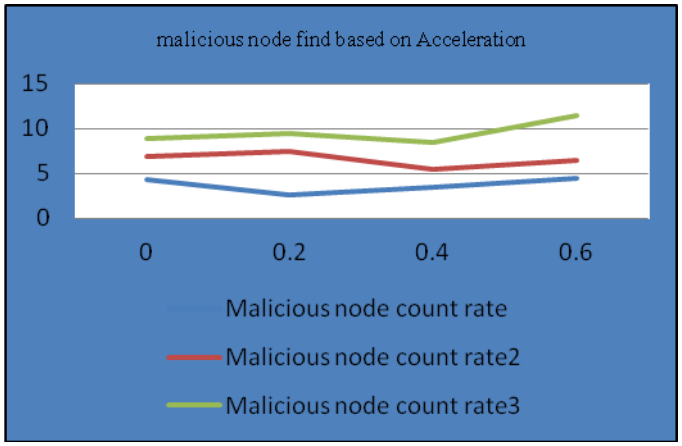

Fig 4: Graph showing the varying accleration

Discussion

All the malicious nodes are always recognized. It is noted that as the acceleration increases, the percentage of false positives increases, as seen in the figure depicting network behavior with varying acceleration. This is because as acceleration increases, the velocity of the nodes increases periodically (mobility update interval based). Proportionally, as discussed in previous section, connection updates happen; nodes create/sever connections often. This leading to an increase in lost, untransmitted messages and un-reachable nodes.

6. CONCLUSION

Here after developed a simulation model the conclusion of paper is to determine a method to identify malicious or compromised nodes in a MANET with mobile nodes based on behavioral attributes. We proposed to use a system in which aberrations of normal behavior are defined Quantitatively by observing data exchange activity. We then selected OMNet++ as the simulator of choice to create an environment dubbing real-life mobile nodes. Where there are mobile nodes, forwarding of data to the correct recipient cannot be done without the use of a routing algorithm. We used an implementation of AODV protocol to perform this function for us. The last phase involved measurement of all the data with various kinds of simulation runs. We scale out varying speed based and varying acceleration based malicious node detection. In this kind of techniques we easily identified misbehavior node and accurately simulate MANET mobile nodes.

7. REFERENCES

[1] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. Computer Networks, 38(4):393–427, 2002.

[2] Patrick Albers, Olivier Camp, Jean-Marc Percher1, Bernard Jouga, Ludovic Me, and Ricardo Puttini. Security in ad hoc networks: a general intrusion detection architecture enhancing trust based approaches. 2005.

[3] A.Rajaram and S.Palaniswami. A trust based cross layer security protocol for ad hoc networks. International Journal of Computer Science And Information Security, 6(1), 2009.

[4] F. Baker. An outsider's view of manet. February 2002. http://tools.ietf.org/html/draftbaker- manet-review-00.

[5] Joo B. D. Cabrera, Raman K. Mehra, and Carlos Gutirrez. Ensemble methods for anomaly detection and distributed intrusion detection in mobile ad-hoc networks.nternational Conference on Mobile Computing and Networks, 9(1), January 2008.

[6] Tracy Camp, Jeff Boleng, and Vanessa Davies. Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications. IEEE International Conference on Distributed Computing Systems, 2(5):483–502, September 2002.

[7] Zhijiang Chang, Georgi Gaydadjiev, and Stamatis Vassiliadis. Routing Protocols for Mobile Ad-hoc Networks: Current Development and Evaluation. Military Communications Conference, MILCOM, 1, 2002.

[8] Tao Li, Min Song, and Mansoor Alam. Compromized sensor node detection: A quantitative approach. IEEE International Conference on Distributed Computing Systems, pages 352–357, 2008.

[9] Dong Seong Kim, Khaja Mohammad Shazzad, and Jong Sou Park. A framework of survivability model for wireless sensor network. IEEE Proceedings of the First International Conference on Availability, Reliability and Security, February 2006.

[10] Amitabh Mishra, Ketan Nadkarni, and Animesh Patcha. Intrusion detection in wireless ad hoc networks. IEEE wireless communications, February 2004.

[11] A.B. Malany, V.R.S. Dhulipala, RM.Chandrasekaran, "Throughput and Delay Comparison of MANET Routing Protocols" Intl. Journal Open Problems Comp. Math., Vol. 2, No. 3, Sep 2009.

[12] D.O. Jörg, "Performance Comparison of MANET Routing Protocols In Different Network Sizes" Comp. Science Project, Institute of Comp. Science and Networks and Distributed Sys, University of Berne, Switzerland, 2003.

[13] A. Shrestha, F. Tekiner, "Investigation of MANET routing protocols for mobility and scalability" International Conference on Parallel and Distributed Computing, Applications and Technologies, Higashi Hiroshima, 2009.

[14] N. Qasim, F. Said, H. Aghvami, "Mobile Ad-Hoc Networks Simulations Using Routing Protocols for Performance Comparisons", Proceedings of the World Congress on Engineering 2008, Vol 1, WCE 2008, UK, July, 2008.

[15] V.N. Talooki, K. Ziarati "Performance comparison of routing protocols for Mobile Ad-Hoc Networks", Asia-Pacific conf. on Comm., APCC‟06.