Illustration, Detection & Prevention of Sleep Deprivation Anomaly in Mobile Ad Hoc Networks

ADNAN NADEEM*, KAMRAN AHSAN*, AND MUHAMMAD SARIM*

RECEIVED ON 27.03.2015 ACCEPTED ON 11.05.2016

ABSTRACT

MANETs (Mobile Ad Hoc Networks) have applications in various walks of life from rescue operations to battle fieled operations, personal and commercial. However, routing operations in MANETs are still vulnerable to anomalies and DoS (Denial of Service) attacks such as sleep deprivation. In SD (Sleep Deprivation) attack malicious node exploits the vulnerability in the route discovery function of the reactive routing protocol for example AODV (Ad Hoc On-Demand Distance Vector). In this paper, we first illustrate the SD anomaly in MANETs and then propose a SD detection and prevention algorithm which efficiently deals with this attack. We assess the performance of our proposed approach through simulation, evaluating its successfulness using different network scenarios.

Key Words: Anomaly Detection and Prevention, Mobile Ad Hoc Network, Security, Sleep Deprivation.

1. INTRODUCTION

MANETs routing protocols can be divided mainly in two types proactive and reactive routing protocols. Reactive routing protocols are most commonly used by the research community to analyze and asses various security vulnerabilities in MANETs. MANETs operations at MAC and network layer are vulnerable to various attacks [1]. SD is a severe DoS attack that can bring the entire network down. It exploits the vulnerabilities of the route discovery procedure of the routing protocol to force victim nodes to power consuming sleep mode. In this paper, we extend our previous initial work on DoS detection [2] by including in dept analysis of SD anomaly its detection and prevention in this paper. We first illustrate different ways of launching this attack and then propose an SD Detection Algorithm which efficiently deals with SD attack. We implement and assess the performance of our proposed approach through simulation. The rest of the paper is arranged as follows: we illustrate the SD attacks in MANETs in detail in section 2. In section 3 we include a brief literature review of the related work. After it, we propose the point detection algorithm that deals with SD attack in Section 4. Then, we present the implementation of the proposed algorithm and present the results which evaluate its successfulness using different network scenarios in section 5. Finally, we summarize the work and highlight the future research in section 6.

2. ILLUSTRATION OF SLEEP DEPRIVATION ANOMALY

SD is a major threat for MANETs. In this attack the attacker forces node to process un necessary packets to cause

* Department of Computer Science, Federal Urdu University of Arts, Science & Technology, Karachi.

congestion in the network and drains the batteries of the nodes. We use AODV as an example to describe in detail the ways this anomaly can be introduced in the network and to illustrate weaknesses in some previously proposed protection mechanism. Fig. 1 shows a snapshot of the network where circles represent nodes and the links between the nodes are shown by dotted lines. We assume node (v_6 :intruder) launches an SD attack through flooding bogus RREQ packets as follows:

Intruder V_6 broadcasts this RREQ (Route Request) with a TTL (Transistor Transistor Logic) value of one (assume TTL_START=1**).** Fig. 2 shows the network after this broadcast. Nodes v_2, v_1, v_5 and v_9 will receive the RREQ (solid arrow line shows the RREQ flow). They will check their routing table entries for route to the destination node v_{25} for this RREQ.

- Because nodes v_2, v_1, v_5 and v_9 do not have the route for node v_{25}, they will also broadcast the RREQ initiated by intruder.
- Nodes which will receive RREQs from v_2, v_1, v_5 and v_9 will first check if they have not processed these requests then further broadcast this request.
- This process will continue because no nodes know the route for this node.

Fig. 3 shows the state of the network under sleep deprivation attack. As can be seen from the diagram after three broadcasts this network is flooded with malicious RREQs. Then this will have cumulative effects and whole network gets flooded. node will drain their batteries.

3. RELATED WORK

Some researchers have focussed on the detection of this denial of service attack in different MANET scenarios. For example, Ping and Zhang [3] considered a RREQ flooding attack in MANETs. They proposed a RREQ flooding prevention mechanism based on neighbor's supervision that maintains a priority queue of the incoming RREQs. This mechanism reduces the priority of RREQs generated

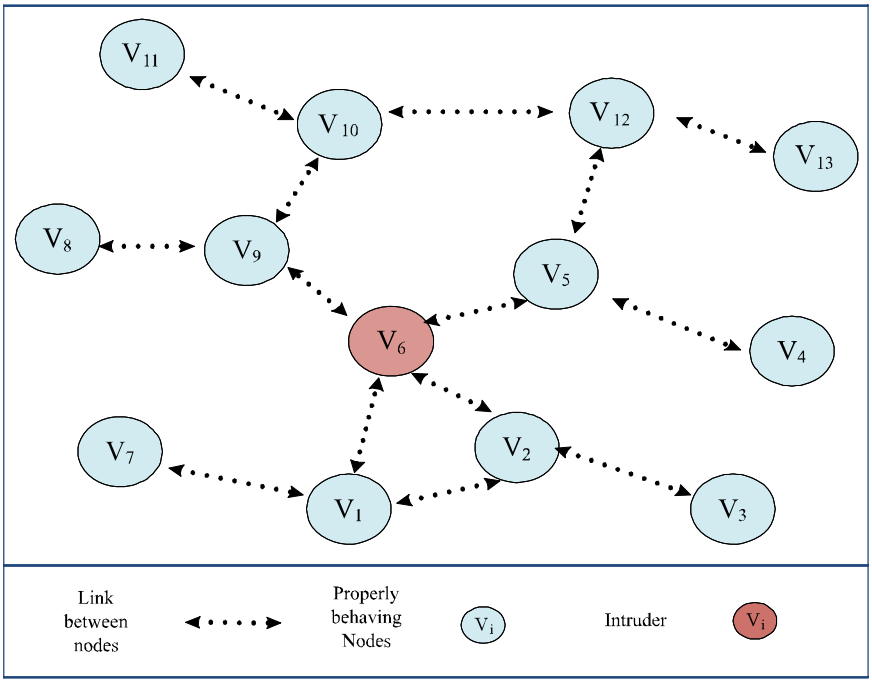

FIG. 1. SNAPSHOT OF THE NETWORK WITHOUT ANY ATTACK [5]

by a specific node if a higher rate of incoming queries from that particular node is observed. Recently authors in [4] have proposed to use the session based history table and limiting the flooding value of the AODV to detect and mitigate the effect of flooding attach in MANETs. They propose to keep record of the average no of RREQ packets

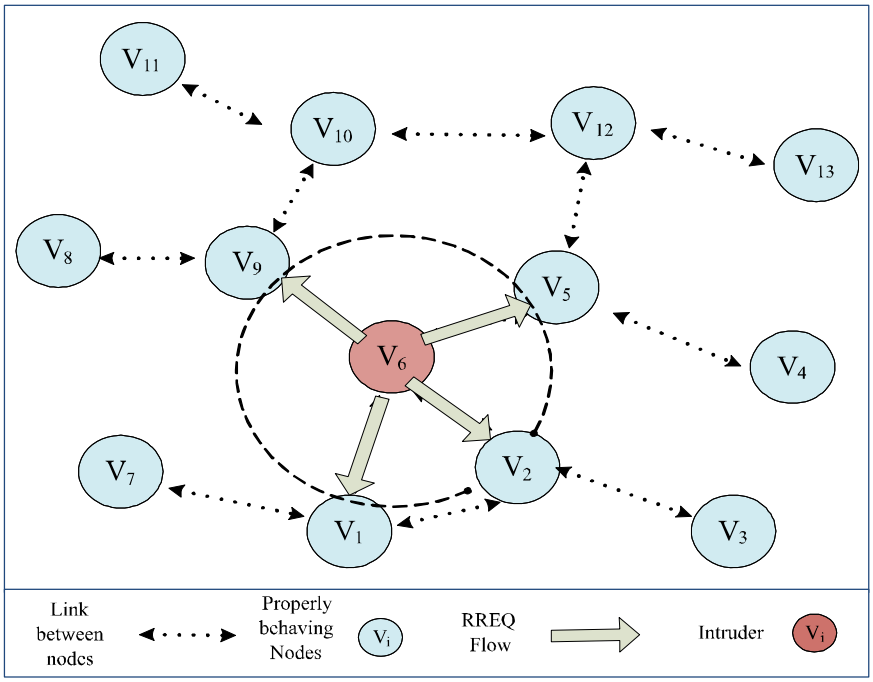

FIG. 2. SNAPSHOT OF NETWORK AFTER INTRUDER GENERATE MRREQ [5]

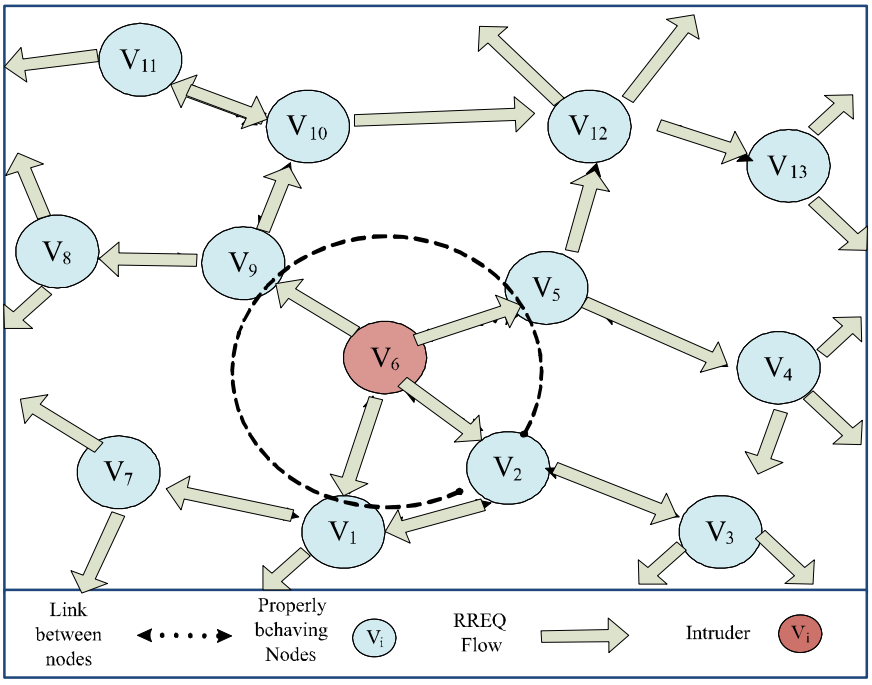

FIG. 3. SNAPSHOT OF THE NETWORK AFTER INTRUDER GENERATE MRREQ [5]

sent and compare it with the discard limit to detect flooding. In [6] authors describe ways through which attacker can drain the batteries of wireless devices such as PDAs and notepads in a mobile computing environment. In an experiment they measure the battery life of notepads and PDAs under this attack and concluded that this attack drains their batteries more quickly and shortened the battery life drastically. Then they propose the power secure architecture with the aim to defend against these attacks by guarantying a minimum battery life even when the device is under attack. The architecture employs two features in a system energy signature monitoring and multilayer authentication. In [7] authors have performed an investigation on the impact of malicious flooding on the QoS (Quality of Service) of MANET though analyzing the throughput in a simulation based study. In another example, Yu and Ray [8] have described SD attacks through two types of injecting traffic attack in ad hoc network as query flooding and injecting data packets. They investigated query flooding and injecting data packets attacks from attacker's point of view and theoretically analyzed the probability of cases where attacker can successfully launch these attacks without being detected. Then assuming nodes can authenticate each other through public key, they propose query flooding attack detection using neighbor monitoring mechanism.

4. SLEEP DEPRIVATION DETECTION AND PREVENTION

Model Assumptions: We note that anomaly detection requires data from normal activities, to build a training profile. We can find such resources in fixed networks for example in [9], but data resources reflecting normal activities of MANETs applications are not available. Therefore, we assume that the initial behaviour of the network is free from anomalies. To illustrate the implementation of the detection algorithm, we also assume that the MANET is organized in clusters. We assume a clustered MANET organization. We select the most capable node in terms of its processing abilities and lowest mobility ratio as CH (Cluster Head) and the others nodes become CN (Cluster Nodes). The only CH is assumed to perform the processing required by the algorithm. We assume threshold based cryptography mechanism such as [10-11] can be used to protect communication between CH and CNs.

4.1 Overview of SD Detection Algorithm

The CH aggregates RREQ information from CNs in the network after each TI (Time Interval) during both training and testing phase. For collecting RREQs information the CH broadcast a request packet limited broadcast to make sure this broadcast does not cause congestion to the network and avoid duplicate request processing. The CH then applies training module of algorithm for *N* TI. An ITP (Initial Training Profile) is the output of training module. ITP reflects the normal (expected) behaviour of the network operations in the network. Once training is complete, the CH performs testing after each TI. Testing module consists of a number of tasks; the first is the detection of intrusion in the network. It updates the ITP in case no intrusion is detected. This update is important in dynamic networks such as MANETs in order to adapt the changes in the network behaviour with the passage of time. If intrusion is detected in the network then the CH performs a second task of the testing module: that is the identification of the intruding nodes. It maintains a TSW (Test Sliding Window) to optimise the success rate of identification, The TSW is a moving window of specific number of TI; for example with a TSW of size five TIs, SD Detection Algorithm only considers the latest five TIs for SD attack detection and prevention. One of the main reasons of maintaining TSW is that we notice that a single detection of any node as an intruder is not sufficient in MANETs as it might leads to wrongly accusing and then punishing the properly behaving node. Therefore, we introduce the concept of a TSW where *d* detections of the *same node* are required in *p* TI to confirm the node as an intruder. The value of *d* is always less than or equal to

p. CH BL (Blacklist) the node once the detection threshold is reached. It then isolates the node by informing all CNs. A further reason of maintaining the TSW is that our algorithm looks for detecting persistent intrusion that can actually deny the services or cause SD of the nodes in the network. Therefore, SD Detection Algorithm does not react to the intrusion if the number of detection in a TSW is less than *d,* in other words if a node introduce a very low volume of attack for a very short time or repeat this activity later. Because SD Detection Algorithm estimates the attacks is not harmful and tolerable for the network.

Sleep Deprivation Detection Algorithm: This section explains the SD Detection Algorithm training and testing module algorithms.

(1) **Training Module:** The main goal of the training module is to gather RREQ information and generate an initial training profile reflecting the normal expected behaviour of the nodes in the network.

$X^i_k = \{X_1, X_2, X_3, \ldots, X_M\}$ is a random variables set showing the number of RREQs received by all CNs in the ith time interval. Where random variables X_1 to X_M subscripts represents the category of the number of RREQs received in a TI and these subscripts are denoted through k = 1 to M . Therefore, X_1 is the random variable representing the lowest and X_M represents the highest category of the number of RREQs received by any CN in a TI. The probability distribution of X^i_k is calculated for the time interval i. Then this process is repeated for the N time intervals. We then calculate the mean $\overline{X^i_k}$ of the probability distributions $P(X^i_k)$ for each of N time intervals, which is then save as an training profile.

(2) **Testing Module:** The SD Detection Algorithm Testing module consists of several tasks including intrusion detection, intruder identification, Accusation Packet handling and intruding node isolation. In intrusion detection, the CH first monitors the network for one TI i and calculates the probability distribution of X^i_k for k = 1 to M. The CH employs the chi-square test to identify any intrusion after monitoring in a single TI. This test tell us that how well the observed model fits with the expected.

Equation (1) is the specific form of the test applied to SD Detection Algorithm, in which X^i_k the observed is and is the expected value of the k^{th} variable from ITP for TI i. Chi-computed is calculated through Equation (1).

$$\chi^2 = \sum_{k=1}^{M} \frac{\left(X^i_k - \overline{X^i_k}\right)^2}{\overline{X^i_k}} \qquad (1)$$

The CH performs hypothesis testing by setting the null hypothesis H_o (*Ho*: Observed distribution fits the expected) and alternative hypothesis H_a (*Ha*: Observed distribution does not fits the expected). The P-value is calculated at selected DoF (Degree of Freedom) and give level of significance (α). In this paper we have selected the standard value of α=5%. The DoF is the number of classes of X^i_k (i.e. the number of groups in which the frequency of RREQ is divided), *M* (maximum value of k). In this paper the value of M and k is determined at run time. If the calculated chi-computed value is larger than the critical value then we reject the null hypothesis H_o, and assume intrusion in the TI.

We then perform the intruder-identification using variable control chart. We use control chart using standard deviation *ó* to identify the intruding node because of its

very low computational overhead. We calculate the *ó* of the number of RREQs generated by all nodes, then set the CL (Control Line), UCL (Upper Control Limit) and LCL (Lower Control Limit). We choose 3*ó* limits because literature suggests that for a normal distribution 99.7% of the observation lies within $\pm 3\sigma$ limits and also from some initial simulations we learn that this limit of $\pm 3\sigma$ keeps the false identification rate to its minimum value. We conclude node V_i to be an intruder if it generates higher RREQs than the UCL. If any node V_i is detected more than *d* times in a test sliding window of *p* intervals then the CH BL the node and send all CNs an AP (Accusation Packet). CH sends AP using limited broadcast with a very low TTL value i.e. TTL=2.

CN first avoids processing a duplicate AP by checking the broadcast id and source address of the packet. All CNs maintains a BLT (Blacklist Table) which contains the entries of current BL nodes in the network. CN checks it is BLT and the CN will ignore and drop the AP to prevent unnecessary network traffic in case the node is already blacklisted. Otherwise, the accused node will be blacklisted by CN. At the end, all nodes isolate the intruder from the network. We update the training profile in case of no intrusion using an EWMA (Exponentially Weighted Moving Average) as given in Equation (2):

$$\overline{X}^{i}_{\left(q,k_1^M\right)} = \beta X^{i}_{\left(q,k_1^M\right)} + \left(1 - \beta * \overline{X}^{i}_{\left(q,k_1^M\right)}\right) \quad (2)$$

Where $\overline{X}^{i}_{\left(q,k_1^M\right)}$ and $X^{i}_{\left(q,k_1^M\right)}$ represents the expected and observed value for update period number *q* respectively. The value of *q* starts at one at the start of the simulation and is incremented for each TI when no intrusion in the MANET is detected. *K* represents the random variable from 1 to *M* and $\beta=2/(q-1)$ is the weighting factor. In EWMA the degree of weight decrease is expressed through a constant smoothing factor ®. The updated expected profile model reflects the current behaviour of the network.

(3) Evaluation of Proposed Algorithm: We use GloMoSim version 2.03 to build the simulation environment. Table 1 contains simulation parameters for all scenarios.

Evaluation of SD Detection Algorithm: In this section, we present the results of the algorithm using a combination of chi square and control chart. We then consider a sliding window and consider detections across multiple TIs. Initially we set the size of TI =30s during the training phase assessment we found statistical inconsistencies in collected audit data in various TIs that

TABLE 1. SIMULATION PARAMETERS

Number of nodes	25	49	64
Terrain dimensions	400*400 m	560*560 m	640*640 m
No of intruders	1 or 2		
Node placement	Grid with grid unit=10 metres		
Time interval TI	100 seconds		
Simulation time	Training + Testing =500+2000=2500 seconds		
Routing protocol	AODV		
MAC protocol	IEEE 802.11		
Pause time	Varies from 10-60 seconds		
Mean speed	Varies from 0-20 m/s		

leads to an undertrained training profile. Therefore, we try training with increase size of TI. We suspect same problem in testing and to avoid that we set the TI=100 second. In simulations for SD Detection Algorithm the CH applies the training module for N=5 TI in order to reduce the chances of an undertrained profile, and then applies the testing module for 20 TI each of 100 seconds. Considering our initial experiments with the combination of d and p values and the analysis of results from [12], we choose to illustrate the results with size of TSW (p)=5 and Detection_to_Accuse (d)=3 i.e. three detection of a nodes is required in five TIs to declare the node intruder. To keep low false positive, we perform 20 runs with each scenario with normal traffic and then with intruders picked randomly from the nodes cause DoS by sleep deprivation.

Figs. 4-6 shows the SR (Success Rate) and FA (False Alarm) rate of the algorithm verses nodes' mean speed in 25, 49 and 64 node networks respectively. When there is no intrusion in the network the FA rate is zero in all three scenarios. In general, graphs in Figs. 4-5 shows good performance of SD Detection Algorithm in terms of high SR and very low FA rate against DoS attacks. The graph in Fig. 4 shows that mobility has a least effects on the performance in a smaller network. However, in larger networks (Fig. 5) the success rate drops slightly when nodes are moving with high mean speed. This is mainly because of two reasons. First is due to the routing protocol (AODV) performance degrades in terms of route discovery procedure where expanding ring broadcasts produced high network overhead.

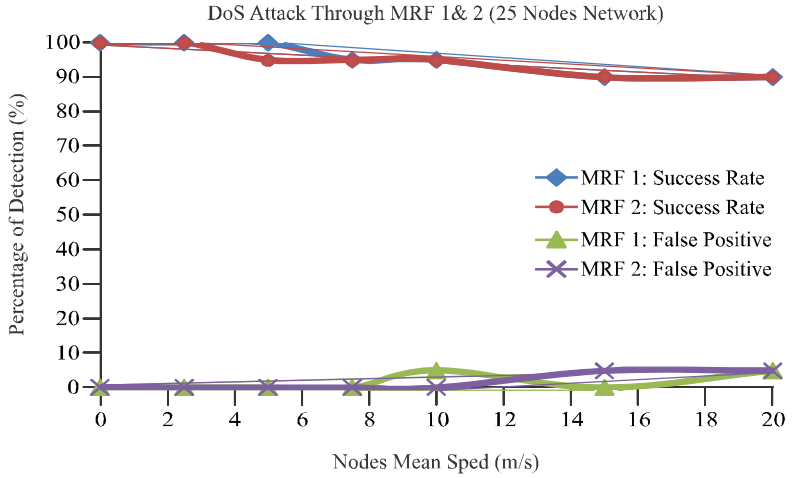

FIG. 4. SUCCESS AND FALSE ALARM RATE VERSUS NODE MEAN SPEED

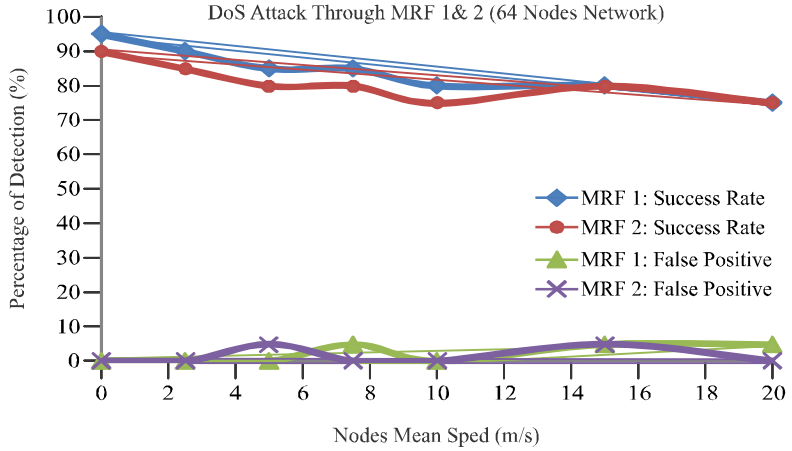

FIG. 5. SUCCESS AND FALSE ALARM RATE VERSUS NODE MEAN SPEED

Therefore, some of the RREQ information send by CNs to CH is lost thus partially reducing the accuracy of SD Detection Algorithm. Secondly we use only one CH the algorithm and in a network of 64 nodes with high nodes mean speed it affects the audit data collection process, therefore we have implemented improve clustering scheme based on virtual.

clusters. It can be seen from the graphs that the gap between success and false positive rate of our approach is wide and the minimum value of the difference between success and false positive rate at certain mean speed of the nodes is 70% that shows the effectiveness of our approach. The time taken by any protection mechanism to detect and prevent attack is another essential parameter.

Since we choose three detections in a TSW of size of 5 TIs to accuse a node, any accusation takes a minimum of 300 seconds. In situations where detection and prevention time is critical the network administrator can re set the size of TI by reducing it and the algorithm will adapt accordingly. In Fig. 6 show the effectiveness of our algorithm in terms of the mean time, it takes to successfully isolate the intruding node & prevent attack in all three scenarios. We see that in general intruders are isolated almost as soon as the algorithm allows.

Effects on Network Performance: We observe the performance impact of the algorithm on the network by monitoring control packets & data packets during simulations. Control packets include routing packets and algorithm packets consists of request packet (packet sent from CH to CNs for audit data request), response packet (packet sent from CNs to CH contains audit data information) and accusation packet (sent from CH to CNs to inform about the intruder). The size of the request packet is 10 bytes, response packet size is 56 bytes and accusation packet consists 18 bytes. We estimate the control packet overhead by calculating the ratio of the number of control packets to the number of data packets delivered to their destination during the simulations of all three scenarios of 25, 49 and 64 nodes network.

Fig. 7 shows overhead of control packet with increasing mean node speed in 25 nodes network. The graph displays the average control packet overhead in three cases, (a) no DoS attack in the network (b) DoS attack with no

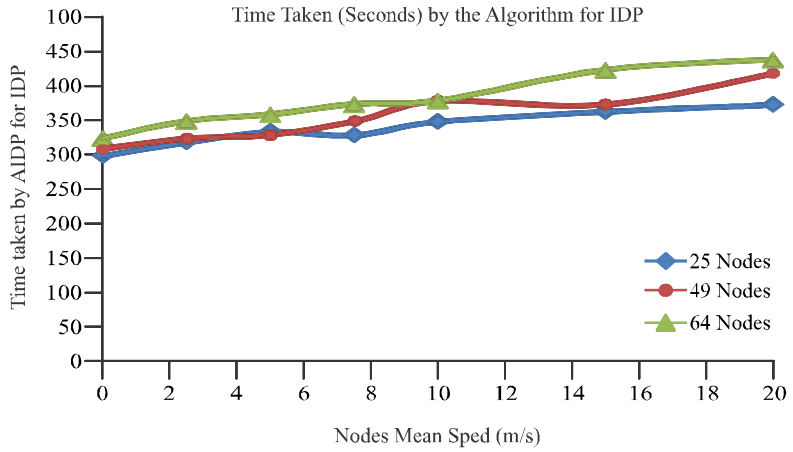

FIG. 6. TIME TAKEN BY SD DETECTION ALGORITHM

protection, and (c) DoS attacks with SD Detection Algorithm in place to protect the network. The algorithm reduces the control packet overhead & conversely improves the network throughput in case of DoS attack. However, the control packet overhead is not as low as that of a network when there is no intrusion because our algorithm also requires certain number of control packets for intrusion detection and prevention in the network. In addition, SD Detection Algorithm takes certain amount of time for detecting, identifying, and isolating intruder which cause slight degradation in network performance before intrusion is detected in the network hence the control packet overhead in graphs in Fig. 7, is slightly higher as compared to when there is no intrusion in the network.

When we compare our algorithm performance with the method propose by Yu et. al. [10] our detection rate is slightly better and in contrast with their strategy our algorithm reduces the false alarm rate to a maximum of 5%. Our algorithm manage to reduce 40% of the control packet overhead caused due to the malicious RREQ, which is not consider in [10].

5. CONCLUSION

MANET routing protocols are vulnerable to DoS attacks, such as sleep deprivation. In this paper, we have focused on protecting MANETs from DoS attacks. We have first illustrated how DoS attacks can be launch in MANETs. We then test the use of control chart only to protect against these attacks. We find out that this method based on static threshold similar to the one proposed is not suitable in MANETs. This is because it does not adapt the dynamic operations of MANETs. We then proposed a SD detection algorithm. It employs intrusion detection, which first use chi-square test to check the overall behavior of the network and indicate intrusion in the network. Although chi-square test is been widely used as an anomaly detector in fixed networks, we have demonstrated that by making reasonable adjustment and including adaptability this test can also detect intrusion in MANETs. Algorithm then use control chart for intruder node identification. Finally, our approach isolates the intruding nodes from the network. Additionally, results shows that our approach decreases the control packet overhead and consequently improve the performance of the network affected by nodes causing DoS attack in the network.

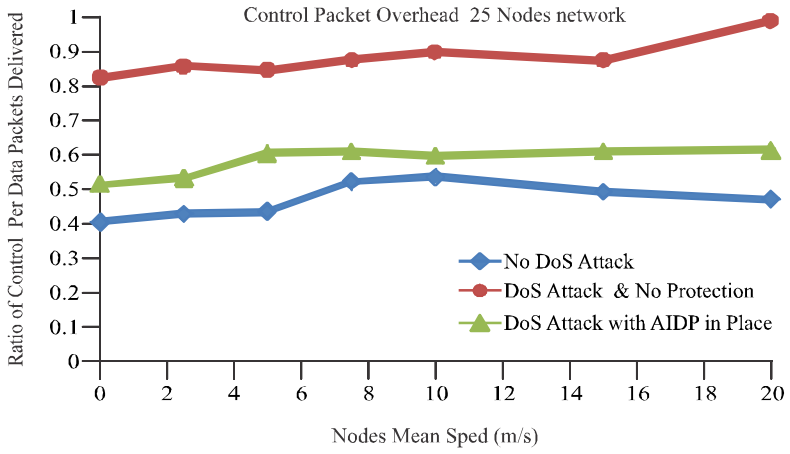

FIG. 7. CONTROL PACKET OVERHEAD VS NODE MEAN SPEED (M/S)

ACKNOWLEDGEMENT

Authors like to express our gratitude and sincere appreciation to the Reviewers/Experts, Mehran University Journal of Engineering & Technology, for their time and valuable feedback. This has certainly improves the contribution in this paper.

REFERENCES

[1] Nadeem, A., and Howarth, M., "A Survey of MANET Intrusion Detection & Prevention Approaches for Network Layer Attacks", IEEE Journal of Communication Survey and Tutorial, Volume 15, No. 4, pp 2027-2045, 2013, Canada.

[2] Nadeem, A., and Howarth, M., "Adaptive Intrusion Detection & Prevention of DoS attack in MANETs", ACM 5th International Conference on Wireless Communication & Mobile Computing, pp. 926 -930, Leipzig Germany, June, 2009.

[3] Yi. P., Dai, Z., and Zhang, S., "Resisting Flooding Attack in Ad Hoc Networks", IEEE Internal Conference on Information Technology Coding & Computing, pp. 657-662, LV, USA, April 2005.

[4] Choube, C., and Murli, M., "Detection of Route Request Flooding Attack in MANET Using Session Based History Table", International Journal of Innovative Science, Engineering & Technology, Volume 2, No. 4, pp. 348-352, Indian, April, 2015

[5] Nadeem, A., "An Intrusion Detection and Prevention Mechanism for Mobile Ad Hoc Network", Ph.D. Thesis, Centre for Communication System Research, University of Surrey, UK, January, 2011.

[6] Martin, T., Hsiao, M., Dong, H., and Krishnaswami, J., "Denial-of-Service Attacks on Battery Powered Mobile Computers", Proceedings of IEEE 2nd International Conference on Pervasive Computing & Communications, Florida, USA, March, 2004.

[7] Verma, S.S., Patel, R.B., and Lenka, S.K., "Investigating Variable Time Flood Request Impact Over QOS in MANET", Elsevier 3rd International Conference on Recent Trends in Computing, India, 2015.

[8] Yu, W., and Ray, K., "Defense Against Injecting Traffic Attack in Cooperative Ad Hoc Networks", Proceedings of IEEE Conference on GLOBECOM, Louis, MO, December, 2005.

[9] "KDD Data Set Used" 3rd International Knowledge Discovery and Data Mining Tool Competition (Available: URL:http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 1999.

[10] Deng, H., and Agrawal, D.P., "TIDS: Threshold and Identity Based Security Scheme for Wireless Ad Hoc Networks", Journal of Ad Hoc Networks, Volume 2, No 3, 291-307, Netherland, 2004.

[11] Dahshan, H., and Irvine, J., "A Trust based Threshold Cryptography Key Management for Mobile Ad Hoc Networks", IEEE Conference on Vehicular Technology, pp. 1-9, USA, September 2009.

[12] Gonzalez-Duque, O.F., Hadjiantonis, A.M., Pavlou, G., and Howarth, M., "Adaptive Misbehaviour Detection and Isolation in Wireless Ad Hoc networks Using Policies", IFIP/IEEE International Symposium on Integrated Network Management, pp 242-255, NY, USA, 2009.