Improvement of Sybil Attack Detection in Vehicular Ad-Hoc Networks using Cross-layer and Fuzzy Logic

MohammadReza Karimi[1], Rasool Sadeghi[2*]
1-Department of Computer Engineering, Dolatabad Branch, Islamic Azad University, Isfahan, Iran.
E-mail: Mohamadrezakarimi64@gmail.com
2-Department of Electrical Engineering, Dolatabad Branch, Islamic Azad University, Isfahan, Iran.
Email: r.sadeghi@iauda.ac.ir (Corresponding author)

Received: July 2020 Revised: October 2020 Accepted: September 2020

ABSTRACT:
Nowadays Vehicular Ad-Hoc Networks (VANETs) are very popular and significantly used, due to their unique abilities to improve road safety. As a consequence, the security of these networks is of great importance and it has become one of the central topics in scientific and research fields such as information exchange. Sybil attack is one of the challenges for Ad-Hoc networks security. In this paper, a cross-layer approach and fuzzy logic method are used to detect the Sybil attacks. The proposed fuzzy logic method has four inputs form different OSI layers: entry time to the network, a number of neighbors, buffer size and signal to noise ratio. These inputs are imported to several membership functions of the fuzzy logic methods and the simulation results indicate that the proposed solution provides a robust technique in Sybil attack detection.

KEYWORDS: VANET, Sybil Attacks, Cross Layer, Fuzzy Logic.

1. INTRODUCTION

Vehicular Ad-Hoc Network (VANET) is an infrastructure-less, self-configuring network of mobile devices connected wirelessly [1]. The VANET can be used for smart transportation systems. In VANETs, there is a new generation of networks that have significantly transformed the transportation system. These networks enable wireless communications between vehicles, which in turn enables many useful applications. Among important topics in VANET, there are issues related to security and misbehavior detection. Misbehavior in communication protocols and routing causes the network to fail, so networks should be defended against them by using misbehavior detection methods. It is obvious that a lack of security can cause problems for the whole system or even can render it inoperative for some time [2].

Related papers have noted numerous security threats for VANETs. Sybil attack is one of these threats, in which a malicious vehicle claims many fake identities [3]. Sybil attacks are considered as a serious security threat for VANET and sensor networks. One method of detecting security attacks including Sybil attacks is the use of detection systems that employ cross-layer and fuzzy approaches as well as node location analysis to verify that the claimed location matches the estimated location.

The VANET consists of vehicles and Roadside Units (RSUs) which use short-range wireless communications to communicate with each other. The VANETs not only can simplify decision making for drivers, but they also have the potential to improve highways' safety, (through using the collected information about catastrophic incidents and traffic data and warning the drivers) [3]. The main goal of the VANETs is to provide safety and comfort for the passengers [4]. In these networks, each vehicle is equipped with the technology necessary to communicate with other vehicles or with road infrastructure. Consider the sudden braking during an accident. As soon as the drivers see the accident happening, they suddenly brake. In this situation, any hesitation by derivers behind will cause a multiple-vehicle collision, which causes significant damages and injuries. The VANETs can help drivers and to some degree avoid such incidents. Such a mechanism works as follows: When the driver that sees the accident uses the brakes to stop the vehicle, the VANET sends an alarm for the vehicles behind. The vehicles that receive the alarm reduce their speed to possibly prevent a multiple-vehicle collision [3], [4].

Paper type: Research paper
DOI: https://doi.org/10.29252/mjee.15.1.9
How to cite this paper: M. R Karimi, R. Sadeghi**,** "**Improvement of Sybil Attack Detection in Vehicular Ad-Hoc Networks Using Cross-layer and Fuzzy Logic**"**,** *Majlesi Journal of Electrical Engineering*, Vol. 15, No. 1, pp. 9-18, 2021.

Various papers have noted numerous security threats for VANETs. Sybil attack is one of these threats, in which a malicious vehicle claims many fake identities [3]. Sybil attacks are considered as a serious security threat for vehicular and sensor networks. These attacks disrupt the potential applications of VANETs by creating the illusion of a high-density traffic jam. Traditionally, in Ad-Hoc and sensor networks three types of defenses against Sybil attack have been introduced, which include radio source test, identity registration and location analysis. In the first method, it is assumed that a radio could not be transmitted or received in more than one channel. This method could not be used in vehicular networks, since a greedy driver can easily acquire many cheap radios. Mere identity registration also could not prevent Sybil attacks, since a malicious node may acquire several identities by using non-technical methods like theft. The location analysis method, analyzes the location of every node in the network and ensures that each physical node is limited to a single identity.

In this paper, by using the cross-layer fuzzy approach and node's location analysis method, we intend to verify that the claimed location matches the estimated location, so that roadside units be able to determine the identities including the identity of the Sybil vehicle. Our design is based on the analysis of the locations of vehicles, and the cross-layer approach. By sending a packet to the vehicles claimed location through a directional antenna, we detect the presence of a vehicle in that location. If the vehicle in the claimed location could receive the packet and transmit a valid response, the authenticity of the claimed location would be confirmed. So this way, instead of sending the packet to all the vehicles each time, we send the packet only when there is a possibility for a Sybil attack. In other words, this method decreases the broadcast transmission overhead.

The remainder of this paper is structured as follows: In the next section, the past research in Sybil attack detection is reviewed. In section three, there is an overview of the problem and the proposed solution. In section four the results are presented and finally, in section five there is the conclusion for this research.

2. RELATED WORKS

There is a lot of research work relating to the topic of VANETs. Considering the topic of this research we review the parts related to malicious node detection and Sybil attack detection. First we will review several methods for detecting misbehaving nodes in VANETs and then review the methods for detecting Sybil attack in VANETs.

2.1. Attacker Node Detection

Marti et al. [5] constantly eavesdrop the communication channel to detect misbehavior of nodes. Whenever a node transmits a packet, the watchdog node observes the next node that receives the packet, and if after a specific period, the next node would not send the packet, the misbehavior parameter for the node is increased. Whenever this parameter exceeds some threshold, that specific node is identified as a misbehaving node and this fact is reported to the source node. In some situations, such as a collision in the channel, collision in packets receipt, nodes that change their transmission power or nodes that wrongly mark other nodes as misbehaving, could not detect misbehavior correctly. Moreover, the partial removal of packets, and advanced cases of misbehavior which are formed under the cooperation between misbehaving nodes, are among the cases which are hard to detect by this method. The Pathrater process chooses the most reliable path based on the data related to path reliability and misbehaving nodes in the path. In the Dynamic Source Routing (DSR) method, there are various routes from source to destination, and based on the protocol the shortest one is selected. When there exists a misbehaving node in the network, the Pathrater module would choose a reliable path instead of a short path. By applying modifications to DSR, a rather reliable path can be selected, even in the presence of misbehaving nodes in all paths. The Pathrater module associates a reliability parameter to every node and the reliability of the path is calculated based on the reliability parameters of consisting nodes. The reliability parameter of nodes should be set in specific time periods so that the nodes that are incorrectly identified as misbehaving, be able to transmit and route again.

In the method proposed by Tseng et al. [6], It is assumed that all nodes observe the whole network and when a node exits the scope of a neighbor node, it enters the scope of another neighbor node. The overall process is that a finite state machine is designed based on the specifications of the AODV routing protocol, which supervises the routing operations, especially the route discovery operation. For each node that is monitored, a transmission table is maintained and each request packet with the corresponding response is considered as a request-response flow. If a supervising node needs information about nodes that are out of its scope, it gets help from its neighbors.

In another research [7] like the proposed method by [6], AODV request and response packets are considered as Request-Response flows and are maintained in transmission tables. Every flow is uniquely identified by the request ID and the source and destination addresses. Every request-response flow has several branches and uses a session tree to maintain these branches. Whenever a supervisor receives an AODV packet, it searches its tree for the packet before

the received packet, and then compares these two packets. If it does not find the packet in its own tree, it gets help from the neighbors. If they do not have any information about this packet too, the packet is treated as a bogus packet.

The authors of paper [8] have presented a method to detect packet drop attacks. In this method, every node counts its number of received and transmitted packets and periodically sends this information to a special coordinator node. This algorithm is easily executed in all nodes, but on the other hand, the bandwidth usage would be a significant overhead for the network, due to the continuous reporting. The coordinator node analyzes the received reports and does the detection job by comparing the received reports from all nodes to each other. In the method proposed by [9], two parameters are used to detect some of these misbehaviors: 1) the percentage of changes in the number of indices of the routing table, 2) the percentage change in a number of steps. The first parameter indicates the number of added or removed indices in a specific time period, and the second parameter indicates the changes in total steps of all indices in a specific time period. In this architecture, some penetration detection methods are employed and the results from each method is compared to others, and in order to increase the detection accuracy, the opinions from neighbor nodes are used too. When mobility of nodes and the changes in routing table increase, the accuracy of misbehavior detection is diminished.

In paper [10], authors present a new approach for detecting malicious nodes in VANETs. In their proposed design, the Detection of Malicious Nodes (DMN) is achieved by improving the Detection of Malicious Vehicles (DMV) algorithm, such that by improving critical parts of the DVM algorithm, the detection of malicious nodes and the performance of the network are increased. The location of vehicle is one of the most important pieces of information in a vehicular network. By identifying the neighbor vehicles with the help of side radars, and verifying the declared specifications, we can reach local safety. The malicious node detection algorithm begins by showing the anomalous behavior of detaching a node from the network. By using three factors of the appropriate verifier, distance and time in the DMV algorithm, we can improve the performance of the DMN algorithm.

In the next section, the research work related to the detection of Sybil attacks in VANETs is reviewed.

2.2. Detection of Sybil Attacks

The first method used for the detection of Sybil attacks was introduced by Douceur, and according to it, there is no practical approach to prevent the attack, and the trust certificate is the only scheme that can completely mitigate the Sybil attack. This method suffers from several problems like scalability, initialization, attack surface and failure. Moreover, this method is based upon the assumption that each node has an identity, which is hard to implement in large-scale networks. In another approach that detection and analysis of the behavior of nodes are used in tandem, the nodes that are freely and independently moving in different directions are considered as legitimate nodes and the nodes that show identical movements are considered as Sybil nodes. By using this method, one can detect the suspicious behavior of nodes, but verifying that the suspicious node is really a Sybil node, remains an open research question.

In another mechanism called “Radio Source Test” [13], it is assumed that nodes in a general network do not have the ability to transfer on more than one channel simultaneously. When a node wants to know if it is itself a victim of a Sybil attack, it dedicates a unique channel to each of its neighbors, and asks them to broadcast an ACK message on these channels, at a specific time. Then it randomly tunes its receiver to one of these channels and waits for an ACK. If it would not receive an ACK, it suspects the associated node. This method works this way, because the malicious node could not simultaneously send ACK messages for all of its fake identities on multiple channels. Authentication methods usually need a lot of memory space to store necessary identity information (shared cryptographic keys, IDs, etc.), and also need complicated processing.

In another method called “Key Random Predistribution” [13], each node randomly selects k keys from an m key store, so that every two nodes have one key in common. Then the ID for each node is combined with the IDs of keys that it has selected, and consequently unique IDs are created. This way each node can be authenticated by verifying some or all of the keys it claims to possess. Drawbacks for this method include high space requirements for storing identity information, and also if an attacker can penetrate the authentication mechanism, then the general integrity of the security mechanism is compromised.

Zhang et al. use “Asymmetric Cryptography” to prevent Sybil attacks [14]. This method uses one important characteristic of the Merkle hash tree, so that each leaf node can be verified provided that its parent has a predefined value. This characteristic is also used in another method, in which each node in the network can analyze and verify the identity of other nodes in the network [15]. This method can only be used in small wireless networks which is a weak point.

Demirbas et al. suggest the utilization of the “Received Signal Strength Indicator (RSSI)” method to detect Sybil attacks [16]. In this method, upon receiving a message, a node calculates the RSSI for that message. Then it associated this calculated RSSI with

the sender's ID (which is contained in the message) and stores it in a search table. If later it receives another message with the same RSSI but with a different sender's ID, it declares a Sybil attack incident. Although using RSSI is not a suitable solution, however it is inherently unstable and unreliable. In addition, network nodes can easily change their transmission strength and deceive the receiver. Also, since RSSI is a function of the transmission strength, different strengths lead to different RSSIs.

In another research, Quercia et al. propose a distributed method [17] for detecting Sybil attacks in networks with mobile nodes. They use a social network idea. In this method, every node collects and maintains two collections which include information about nodes. One collection is for the friends' network and another one is for the enemies' network. The first collection includes trustworthy nodes and the second one includes suspicious ones.

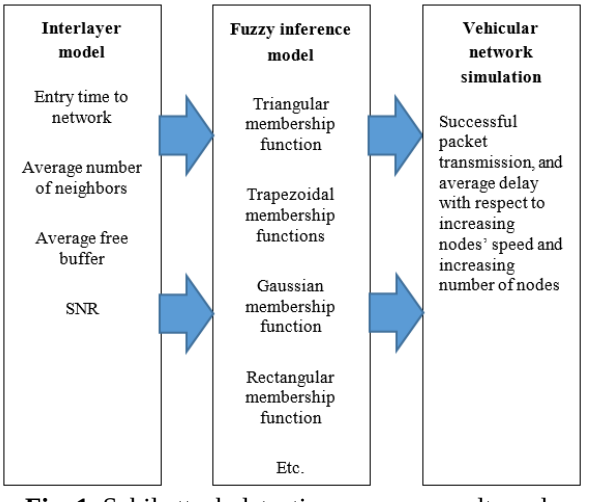


Fig. 1. Sybil attack detection process, results and evaluation.

3. THEORETICAL FOUNDATIONS AND PROPOSED METHOD

In this section, the proposed method of this paper, for detection of Sybil attacks in vehicular Ad-Hoc networks is presented, which is based upon a combination of the fuzzy logic and the cross-layer approach. The overall framework for the proposed method is shown in. As shown in Fig. 1, using the cross-layer approach, first parameters that help detecting Sybil attacks are extracted, and then these parameters are fed into the fuzzy system as input. The output of the fuzzy system is applied to the vehicular network simulator and in that simulator, the parameters related to Sybil attacks are evaluated.

3.1. Network Topology

In the network topology shown in Fig. 2, a destination node is at the center and other nodes are randomly distributed around it. VANETs have issues related to data transmission security and reliability. So is it inevitable to use software (like MATLAB) as the least expensive way to find out about their possible issues in real operational conditions. Nodes are configured with different types of communications. Vehicles move within a specific range of the network. Data exchange between nodes is done using UDP and CBR traffic. The communication channel is wireless and the communications is of radio type. Different routing protocols have been proposed for data transfer in the network, however, in our simulation, we have created a hybrid routing protocol for heterogeneous vehicular networks.

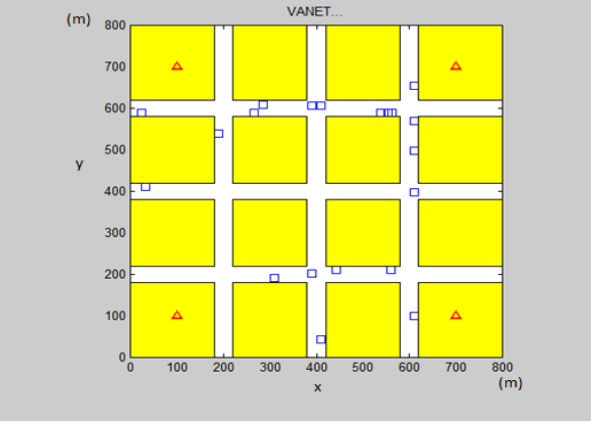


Fig. 2. Initial network topology.

3. 2. Proposed Cross-layer Model

In the proposed cross-layer model, a combination of width and length is used simultaneously. Data from various layers are processed in the fuzzy system and the presence of an attack is detected. Parameters used for decision making are entry time to network, average buffer, signal to noise ratio (SNR), the average number of neighbors. These four inputs are chosen based on various research papers and the analyses are done to improve the performance [2], [18- 19].

1. **Entry time to network**: The less time it takes for a node to enter the system, the more is the probability of insecurity, and consequently the more the possibility of a Sybil attack. This fact is shown in equation 1, in which p is the probability of the Sybil attack and t is the entry time to the network [20]-[21].

$$p \propto {}^{1}/_{t} \quad (1)$$

2. **Average number of neighbors**: The more neighbor nodes a specific node has

interactions with, the less probable that node would be a Sybil node. The degree of each node in the network graph indicates the number of neighbors for that node. Thus the average number of neighbors for each node is calculated by equation (2) [28], in which D_i indicates the degree of that node in the network graph, $\bar{D}_{path}$ is the average degree of nodes in the path, hop_i is the ith step and n is the number of nodes in the network [22].

$$\bar{D}_{path} = \frac{\sum_{i=1}^{n} D_i}{\sum_{i=1}^{n-1} hop_i} \quad (2)$$

3. **Average remaining buffer**: each node has an amount of buffer or free space, that is used when transmitting a packet, and as the amount of packet transmission increases, it is more likely that the buffer size is used up. On the other hand, as the probability of a full buffer increases, energy consumption increases and the number of incoming packets from the sink decreases, which has a direct effect on detecting the attack. So one of the parameters that can be helpful in this situation is the amount of free buffer in the system. In equation (3), the average free buffer is an input to the fuzzy system [18], [22]:

$$\bar{B}_{path} = \frac{\sum_{i=1}^{n} B_i}{\sum_{i=1}^{n-1} hop_i} \quad (3)$$

In the above equation, B_i indicates the buffer for the ith node, $\bar{B}_{path}$ is the average node buffer along the path, and hop_i is the ith step.

4. **Signal to noise ratio**: if $\mathbf{L_{packet}/k}$ is the number of encrypted blocks for transmitting L bits (k is the length of each block), Some commercial transmitters do not show the $\mathbf{E_b/N_0}$ ratio, but the receiver's signal strength display can help by calculating SNR. In fact, this quantity shows the strength of noise compared to signal strength in a system. SNR can be calculated by equation (4) in which E_b is the signal energy, N_0 is spectral density of noise, R and B_N are indicators of signal strength [18], [22]:

$$SNR(d) = \frac{E_b}{No}\frac{R}{B_N} \quad (4)$$

After parameters are extracted from the cross-layer model, each of these parameters are applied to the fuzzy inference system as an input. The fuzzy inference system has four fuzzy inputs and each input has three membership functions. Constants in the standard set of each input are calculated by using equations (1) to (4). Next the basics of the fuzzy system is introduced and it is decided if a Sybil attack has occurred.

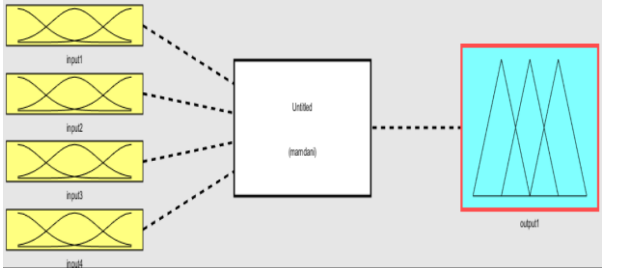
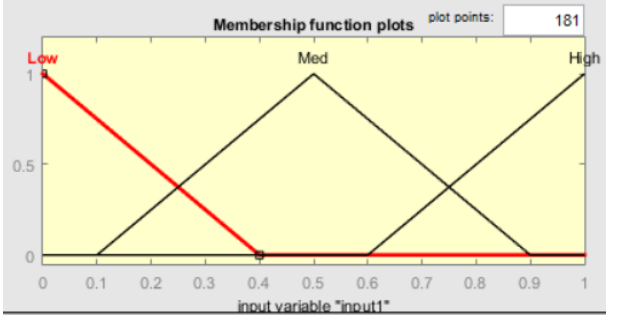
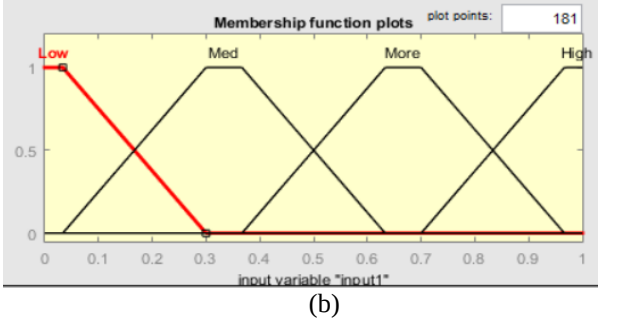


Fig. 3. Fuzzy system.

In Fig. 3, four fuzzy inputs and one fuzzy output is indicated. The first input is the entry time to the network which indicates if the node's energy level is low, medium or high. Its membership function is shown in equation (5) and section (a) of Fig. 4. x is the input value and a, b and c are constants which are defined in the membership function.



(a)



(b)

Fig. 4. Membership function in fuzzy interface: (a)Triangular, (b) trapezoidal.

Section (b) of Fig. 4 shows the trapezoidal membership function which has four constant values a, b, c and d. By specifying these values, all the membership functions for the time input are determined. For simplicity, for every membership function only the constants a, b, c and d are shown which are calculated according to membership equation (6) and values in equation (7).

$$f(x;a,b,c)=\begin{cases}\text{Low}:\max\left(\min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right) & [a,b,c]=\\ \text{Med}:\max\left(\min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right) & [a,b,c]=\\ \text{High}:\max\left(\min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right) & [a,b,c]=\end{cases} \quad (5)$$

$$f(x;a,b,c,d)=\begin{cases}0, & x\le a\\ \frac{x-a}{b-a}, & a\le x\le b\\ 1, & b\le x\le c\\ \frac{d-x}{d-c}, & c\le x\le d\\ 0, & d\le x\end{cases} \quad (6)$$

$$[a\ b\ c\ d]=[0.35\ 0.4833\ 0.5167\ 0.65] \quad (7)$$

Second input indicates the average number of neighbors and shows if the number of neighbors is low, medium or high. Its membership function is depicted in equation (5) and section (a) of Fig. 4. Constants are for the four levels case (section (b) of Fig. 4) which conforms to membership equation (6) and values in equation (7).

The third input indicates the free buffer space in a node which is defined in three levels of weak, medium and strong. Its membership function is shown in equation (5) and section (a) of Fig. 4. Constants are for the four levels case (section (b) of Fig. 4) which conforms to membership equation (6) and values in equation (7).

The fourth input indicates the SNR which is also defined in three levels of low, medium and high. Its membership function is shown in equation (5) and section (a) of Fig. 4. Constants are for the four levels case (section (b) of Fig. 4) which conforms to membership equation (6) and values in equation (7).

For input membership functions, triangular and trapezoidal functions are used, and for each input the equation is indicated. The output of the fuzzy system also uses triangular and trapezoidal functions. The goal is to decrease computation costs. As values of output approach to one, the probability of an attack increases.

4. RESULTS

In this section, we evaluate the proposed method and analyze the results using simulation. In this paper the simulation of VANETs is done for different scenarios. Designed scenarios are in a virtual area while applying real characteristics of VANETs. In these simulations, effect of parameters like a number of vehicles, their speed and the communication scope is evaluated, and it is compared to the results from the PPBMA method. The process of transforming a string of characters to a shorter string with a constant length is called hashing, which is used to speed up searching and verifying the integrity of transferred data. A distributed hash table is just like a simple distributed table, except for having the ability to intercept a large amount of data across multiple nodes. These tables help avoid problems in the system when adding or removing nodes to the system. An Ad-Hoc vehicular network without security considerations is totally vulnerable to attacks like bogus warning message broadcasting and genuine warning message busting. So security is one of the most critical issues in implementing such networks.

Using MATLAB 2014, the simulation scenario is implemented as shown in Fig. 5. In this figure, big grey square regions show urban areas and the white regions between them indicate streets and vehicle pathways. Small squares in streets and paths show vehicular nodes.

The small triangles in urban areas indicate roadside units. The simulation area is 800 x 800 square meters and the number of vehicles is 30. Two random momentary images of the running simulation at times t_m and t_n is shown in Fig. 6, which shows nodes in motion. The set of values used for speed in this simulation are as follows:

Speeds = [20 40 60 80] m/s

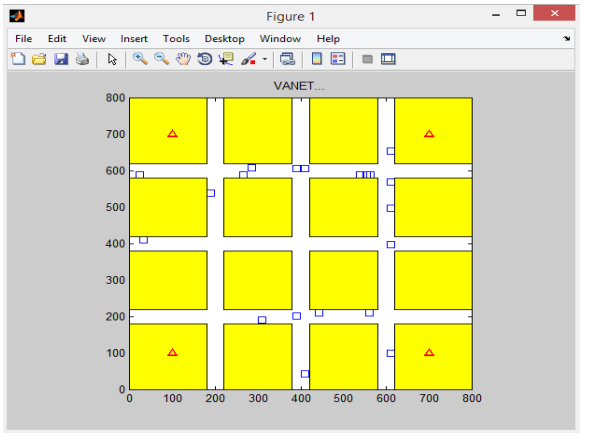


Fig. 5. Simulation topology in MATLAB.

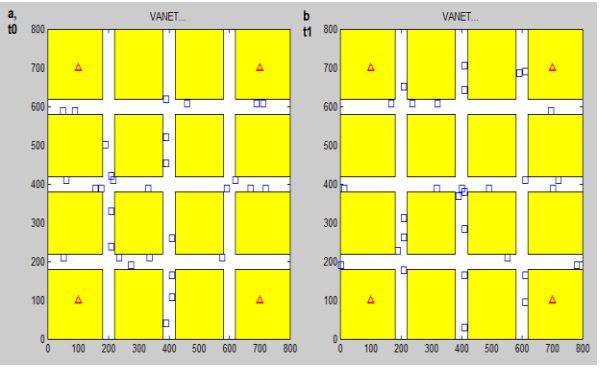


Fig. 6. Two random momentary images of the simulated scenario at times t_n and t_m.

This simulation ran for 20 minutes in which all the nodes had specific motion characteristics and directed trajectory which could be changed arbitrarily. Results like average delay, successful transfer rate and etc. are evaluated while considering changes in speed, a number of nodes and the area of communication.

The simulation defined above is run in an outdoor space. There are no height differences between nodes. Simulation time is the same for all scenarios and is equal to 2500ms. It is worth mentioning that simulation is done in a way that can output different statistics. Also using the software environment, it is possible to see a 2D animation of nodes in motion.

The number of nodes is a randomly distributed variable which can be easily increased or decreased.

The results of the proposed method are shown in two formats. As shown in section 3, the fuzzy system is defined in two formats, one for three-membership-function case (A) and the other for the four-membership-function case (B). Here, the results for these methods are shown in two formats for method A and method B.

4.1. Variation in the Number of Vehicles

Fig. 7 shows the average delay with respect to the number of vehicles. As shown, as the number of nodes increases, the average delay for nodes in the network also increases.

As shown, in the PPBMA method, as the number of nodes increases, the average delay increases exponentially. On the other hand, although in the proposed method, the increased number of nodes also causes the delay to increase, but the order of growth is much less than the PPBMA method, which results in a better performance in large scale networks. As indicated in the figure above as the number of nodes increases, the average delay increases accordingly, since the size of packets and the processing workload grow.

To improve security, the PPBMA method uses hash tables, while the proposed method uses fuzzy functions which is a linear system that offers high computational speed and low computational workload.

Another parameter that is analyzed, is the packet transfer success rate. Fig. 8 shows the packet transfer success rate with respect to variations in the number of nodes. As expected as the number of nodes increases the packet transfer success rate decreases.

In most networks as the number of nodes grows, the number of successful delivered packets decreases. The first reason is congestion due to the increased number of nodes and the second reason is that our network is entirely time dependent. As time passes, it is possible that while transmitting packets, one node goes out of another node's view along the path. So the network's topology, which is dependent on the node's speed, will entirely change.

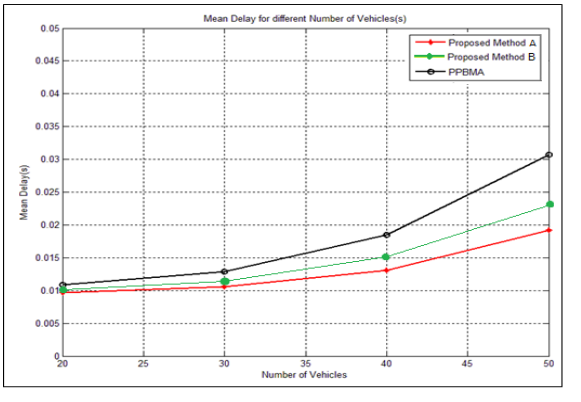


Fig. 7. The average delay for different number of vehicles.

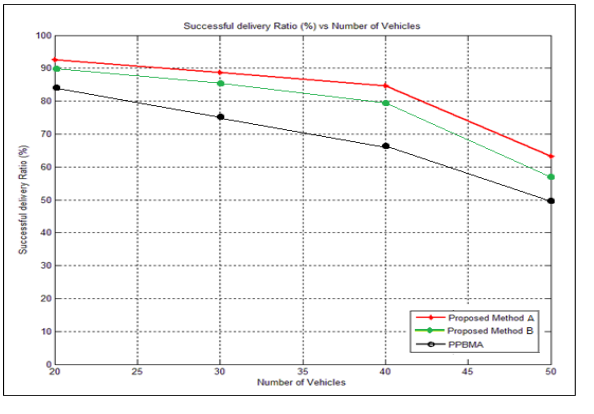


Fig. 8. Successful packet transmission rate for different number of vehicles.

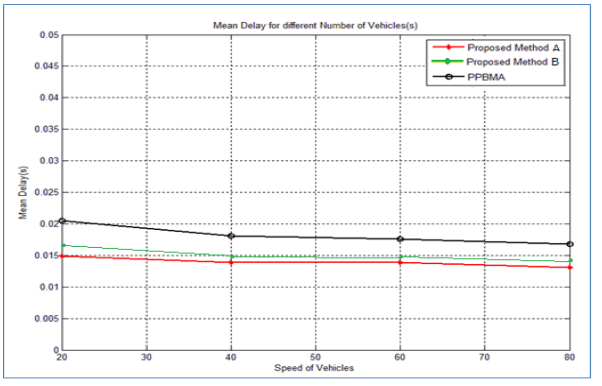


Fig. 9. Average delay versus movement speed.

4.2. Change in the Motion of vehicles

Fig. 9 shows the changes in the average delay variation of the speeds. As expected, as the speed of nodes in the network increases, the average delay decreases. So the performance of the proposed method improves as nodes move faster. As the speed of network nodes increases, the average delay is quite constant. As shown in figures relating to the proposed method and PPBMA method, the proposed method has less delay.

Our system is absolutely topology dependent, and in Ad-Hoc networks, since the speed of movement is high, topology is time-dependent. For example, two nodes exchange data with each other and when they arrive at an intersection, one of them goes east and another one goes west.

In this situation, nodes go out of each other's view, so the transmitting packet would become a failed packet. The higher the speed, the more the probability of a failed packet.

In the proposed method, we do not put in a parameter for speed, the number of nodes is fixed and the number of packets is decreased, which means delay has decreased. Packets that arrive faster have less delay and thus average delay decreases.

Fig. 10 shows the changes in successful packet transmission rate with respect to time, for various speeds of 20, 40, 60 and 80 Km/hr. The simulation time is 2500ms. As shown, as the speed of movement increases, packet transmission success decreases. The more the speed increases, the more successful packet transmission rate decreases.

4.3. Communication Ranges

Fig. 11 shows the changes in average delay when the communication range varies. When the communication range increases, the average delay also increases. The results from this figure seem totally possible in real conditions, in which as range increases, delay increases too. As shown, in networks with short communication ranges, both methods are similar, but as range increases, average delay increases. As shown and mentioned, in environments with short communication range, two methods behave similarly, but as number of nodes (vehicles) increases in VANETs, the proposed method performs better and shows more convergent behavior compared to the PPBMA method.

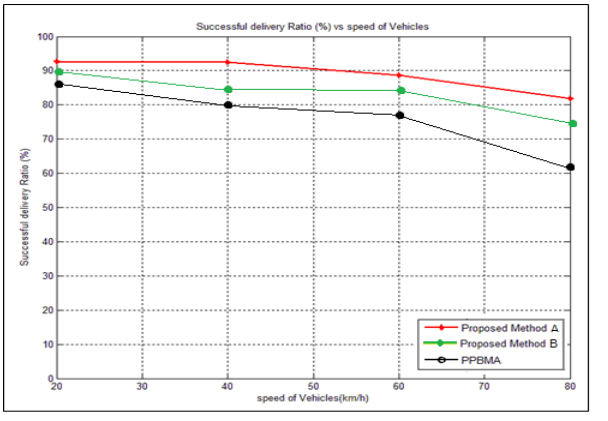


Fig. 10. Successful packet transmission rate for various movement speeds.

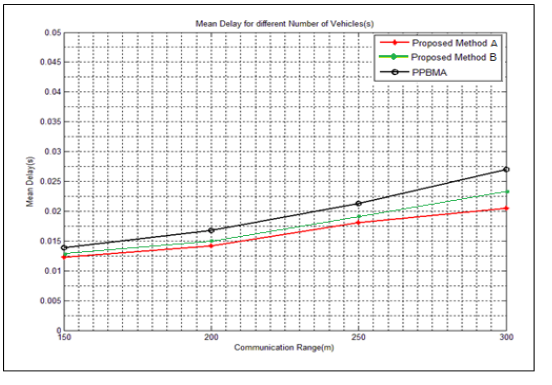


Fig. 11. Average delay versus communication range.

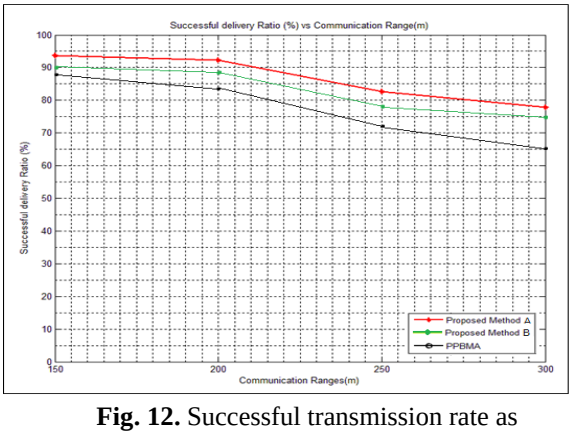


Fig. 12. Successful transmission rate as communication range increases.

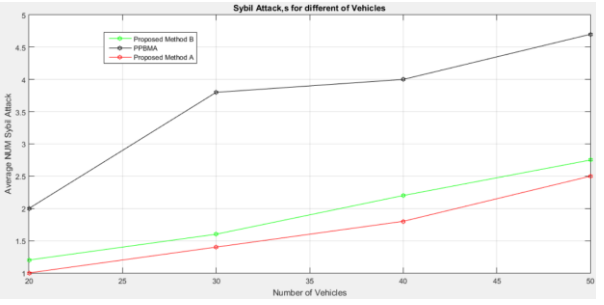


Fig. 13. Number of Sybil attacks for different number of vehicles.

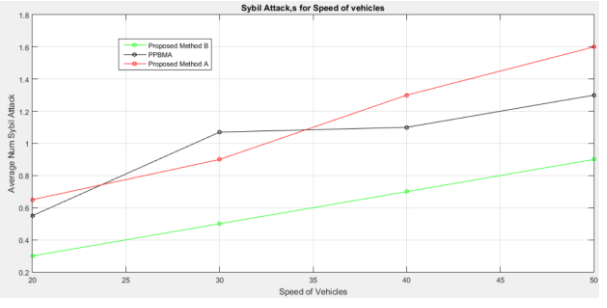


Fig. 14. Number of Sybil nodes as vehicle speed increases

Fig. 12 shows changes in successful packet transmission rate with respect to increasing communication range. As expected, as communication range increases, successful transmission rate decreases which can be due to increasing distances, increasing the probability of going out of scope while transmitting and etc.

The range of communication is the range in which two nodes can see each other. The Communication range makes for two conflicting cases. In the first case, for example when a node sees another node from 100-meter distance, is also visible from 150 meters, because the communication range is 200 meters. So when two nodes are getting away from each other, they can still see each other, and consequently the number of successfully delivered packets increases. With changes in speed, nodes do not exit each other's communication range. In the second case, as the communication range gets larger, more nodes can see each other leading to congestion. Since routing is identical in both methods and is not changed, this decreases successful packets.

In other words, as the number of nodes increases, there would be more routes for communication, and thus more packets are exchanged in the network. Fig. 13 shows the number of Sybil attacks/nodes with respect to changes in the number of nodes/vehicles. As shown, only one of the proposed methods shows less nodes compared to the reference method, but as the number of vehicles increases, the number of Sybil attacks increases too.

Fig. 14 shows this situation for changes in the vehicle's speed. In this method, the reference method performs better than other methods. These numbers show the average number of attacks in various iterations of simulation. In this case, the number of Sybil attack increases too, as speed increases. In high speed, our system is not optimal, which means it runs slowly and faces some issues, especially in case A. But in the case B, because there is more decision power, again system runs slowly, but our fitness function performs better, compared to the other cases in the proposed method.

5. CONCLUSION

In this paper, the location of the node is verified using the cross-layer approach and fuzzy logic. The claimed location is verified against the estimated location. Our proposed design is based on verifying the locations of vehicles and also the cross-layer approach. By sending a packet to the claimed location of the vehicle, using a directional antenna, the presence of a vehicle in the claimed location is detected, and if the vehicle at the claimed location was able to receive the packet and send a valid response, the claimed location is verified. So this way, instead of sending packets to all vehicles all the time, we only send packets when there is a probability for a Sybil attack. In other words, this issue decreases the broadcast overhead.

REFERENCES

[1] N. Patel, P. Modi, N. l Patel, and P. Modi, **"Detecting Sybil attack using AODV in MANET,"** *Int. J. Adv. Eng. Res. Dev.* Vol. 1, No. 5, pp. 1–5, 2014.

[2] T. Krag and S. Büettrich, **"Wireless mesh networking,"** *posted Wirel. DevCenter Jan*, Vol. 22, pp. 1–9, 2004.

[3] J.-P. Hubaux, J. Luo, and M. Raya, **"The security of vehicular networks.,"** *in Workshop on Wireless Security*, 2005, pp. 31–32.

[4] B. Parno and A. Perrig, **"Challenges in securing vehicular networks,"** *in Workshop on hot topics in networks (HotNets-IV)*, No. 4, pp. 1–6, 2005.

[5] S. Marti, T. J. Giuli, K. Lai, and M. Baker**, "Mitigating routing misbehavior in mobile ad hoc networks,"** *Proc. 6th Annu. Int. Conf. Mob. Comput. Netw.*- MobiCom ' pp. 255–265, 2000.

[6] C.-Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt, **"A specification-based intrusion detection system for AODV,"** *Proc. 1st ACM Work. Secur. ad hoc Sens. networks - (SASN '03),* pp. 125–134, 2003.

[7] Y. Chin and H. Tseng, **"Distributed Intrusion Detection Models for Mobile Ad Hoc Networks,"** *Ph.D. Thesis*, pp. 1–133, 2006.

[8] F. Anjum and R. Talpade, **"LiPaD: lightweight packet drop detection for ad hoc networks,"** *Veh.*

Technol. Conf. 2004. VTC2004-Fall. 2004 IEEE 60th, Vol. 2, No. C, p. 1233–1237, 2004.

[9] B. Sun, K. Wu, and U. W. Pooch, **"Routing anomaly detection in mobile ad hoc networks,"** *in Computer Communications and Networks, 2003. ICCCN 2003. Proceedings. The 12th International Conference on*, pp. 25–31, 2003.

[10] U. Khan, S. Agrawal, and S. Silakari, **"Detection of Malicious Nodes (DMN) in vehicular ad-hoc networks,"** *Procedia Comput. Sci.*, Vol. 46, No. Icict 2014, pp. 965–972, 2015.

[11] J. R. Douceur, **"The sybil attack,"** *in International workshop on peer-to-peer systems*, pp. 251–260, 2002.

[12] C. Piro, C. Shields, and B. N. Levine, **"Detecting the sybil attack in mobile ad hoc networks,"** *in Securecomm and Workshops*, pp. 1–11, 2006.

[13] J. Newsome, E. Shi, D. Song, and A. Perrig, **"The sybil attack in sensor networks,"** *Proc. third Int. Symp. Inf. Process. Sens. networks - IPSN'04*, pp. 259, 2004.

[14] Q. Zhang, P. Wang, D. S. Reeves, and P. Ning, **"Defending against Sybil Attacks in Sensor Networks,"** *25th IEEE Int. Conf. Distrib. Comput. Syst. Work.*, pp. 185–191, 2005.

[15] Y. Zhang, W. Liu, W. Lou, Y. Fang, X. Ren, and H. Yu, **"Location-based compromise-tolerant security mechanisms for wireless sensor networks,"** *IEEE J. Sel. areas Commun.*, Vol. 24, No. 2, pp. 247–260, 2006.

[16] Y. Song, M. Demirbas, and Y. Song, **"RSSI-based Scheme for Sybil Attack Detection in Wireless Sensor Networks,"** *Proc. 2006 Int. Symp. World Wireless, Mob. Multimed. Networks*, pp. 564–570, 2006.

[17] D. Quercia and S. Hailes, "**Sybil attacks against mobile users: Friends and foes to the rescue,"** *Proc. - IEEE INFOCOM*, 2010.

[18] K. Rabieh, M. M. E. A. Mahmoud, T. N. Guo, and M. Younis, **"Cross-layer scheme for detecting large-scale colluding Sybil attack in VANETs,"** *IEEE Int. Conf. Commun.*, pp. 7298–7303, 2015.

[19] C. M. Chen, Y. L. Chen, and H. C. Lin, **"An efficient network intrusion detection,"** *Comput. Commun.*, Vol. 33, No. 4, pp. 477–484, 2010.

[20] D. M. da C. e Castro and M. Oliveira, **"Thwarting the Sybil Attack in Wireless Ad Hoc Networks,"** *Gsd.Inesc-Id.Pt*, pp. 1–28.

[21] D. Monica, **"Thwarting the sybil attack in wireless ad hoc networks,"** *Inst. Super. Tec.*, 2009.

[22] P. Tyagi and D. Dembla, **"Performance analysis and implementation of proposed mechanism for detection and prevention of security attacks in routing protocols of vehicular ad-hoc network (VANET),"** *Egypt. Informatics J.*, Vol. 18, No. 2, pp. 133–139, 2017.