Detecting black hole attack in wireless ad hoc networks based on learning automata

2 Author(s)

Soleimani, M.T. ; Dept. of Electr. & Comput. Eng., Qazvin Azad Univ., Qazvin, Iran ; Ghasemi, A.

Abstract | Authors | References | Cited By | Keywords | Metrics | Similar

Wireless ad hoc networks are vulnerable to several attacks including packet dropping. In this kind of attack, a malicious node tries to absorb network traffic and then drop them to form a denial of service (DOS) attack. Black hole attack is a sort of DOS attack. In this attack, a malicious node advertises itself as having the shortest and freshest path to the destination. Once traffic is redirected to this node, it simply drops them. In this paper, we present a novel solution to detect the black hole attack based on learning automata (LA). By using learning automata in a random environment, nodes can learn and adopt its behaviors based on the received signals from the environment. To the best of our knowledge, our work is the first one that tries to detect black hole attack by using LA. The simulation results in NS2 show that using proposed solution, attack is detected successfully.

Published in:
Computer Sciences and Convergence Information Technology (ICCIT), 2011 6th International Conference on

Date of Conference:
Nov. 29 2011-Dec. 1 2011

Page(s):
514 - 519

Conference Location :
Seogwipo

Print ISBN:
978-1-4577-0472-7

Publisher:
IEEE

INSPEC Accession Number:
13037406

Detecting Black Hole Attack in Wireless Ad Hoc Networks Based On Learning Automata

Mohammad Taqi Soleimani
Department of Electrical and Computer Engineering
Qazvin Azad University, Qazvin, Iran
soleimani@qiau.ac.ir

Abdorasoul Ghasemi
Department of Electrical and Computer Engineering
K. N. Toosi University of Technology, Tehran, Iran
arghasemi@eetd.kntu.ac.ir

***Abstract*— Wireless ad hoc networks are vulnerable to several attacks including packet dropping. In this kind of attack, a malicious node tries to absorb network traffic and then drop them to form a denial of service (DOS) attack. Black hole attack is a sort of DOS attack. In this attack, a malicious node advertises itself as having the shortest and freshest path to the destination. Once traffic is redirected to this node, it simply drops them. In this paper, we present a novel solution to detect the black hole attack based on learning automata (LA). By using learning automata in a random environment, nodes can learn and adopt its behaviors based on the received signals from the environment. To the best of our knowledge, our work is the first one that tries to detect black hole attack by using LA. The simulation results in NS2 show that using proposed solution, attack is detected successfully.**

Keywords- Wireless ad hoc network; AODV; Security; Packet dropping; Black hole attack; Learning automata

I. INTRODUCTION

Wireless ad hoc networks consist of a set of mobile nodes, to communicate using a cooperative scheme. These networks are self-organized and self-configured and have no infrastructure and centralized management. Each node of the network can communicate with all other nodes, which are placed in their communication range. If a node has packets to send, and the receiver is not in its range, it requires the cooperation of other nodes to be able to deliver its packets to the destination. Therefore, each node of network can act as a host as well as a router. Applications of such networks include establishing survivable, efficient, dynamic communication for emergency/rescue operations, disaster relief efforts, military, civilian networks and many other scenarios that require rapid deployment.

The packet dropping attack is a kind of denial of service attacks. In this type of attack, some nodes of the network can not transmit or receive their packets correctly. A packet may be dropped under various reasons. Wireless communication leads to packet dropping, if collisions occur.

Also, other reasons of packet dropping include: congestion in the medium, overflow of the transmission queue, broken links which they are cause of nodes' mobility and lacks of energy resource, errors in the packet, expiration the value of TTL (time to live). A selfish node [1, 2] wants to preserve its resources while using the services of others and consuming their resources. Therefore, the selfish node cooperates in the route discovery phase correctly, but it avoids forwarding other nodes' packets to save its resources and then drop them. Also, malicious behaviors of the adversary nodes in the network are the serious reason of packet dropping. Throughout this paper, we will deal to this type of packet dropping.

The black hole attack [1, 2] is a dangerous attack which adversary nodes try to absorb all packets on the network and then drop them to form a denial of service. The target of this attack is mainly the reactive routing scheme of the network. In the following we discuss about a reactive routing protocol and effect of this attack on it.

A. The Ad hoc On-Demand Distance Vector

The Ad hoc On-Demand Distance Vector (AODV) routing protocol [3] is developed based on the Destination Sequenced Distance Vector (DSDV) routing protocol [4] for wireless ad hoc networks. This protocol is a reactive.

Therefore, when the source node wants to establish a connection to the destination node it will initiate route discovery phase, if it has not any knowledge of that in its routing table. To find a path to the destination, it broadcasts a route request (RREQ) packet to all one-hop neighbors. Receiving a RREQ packet by intermediate node, first, it will looks up its routing table to find a path to the destination. If it has, it will be sends back a route reply (RREP) packet to the origin node. Otherwise, it will rebroadcast it to its all

immediate neighbors until it reaches to the destination node or any intermediate node that has knowledge of it. In this routing protocol, the metrics of calculating route between source and destination are based on the freshness and the length of the path. If an advertised path has fewer hops, it will be judged as a shortest path. In other words, the first received route will be as a shortest path. The value of freshness of a route will be obtained based on the value of the sequence number of RREP. Every route reply packet contains a sequence number which implies freshness of the route. The source node considers a path as the freshest among all received, if it has the greatest sequence number.

B. *Black hole attack*

Since, there is no security mechanism in AODV; this routing protocol is subject to many threats. In this routing protocol, it is supposed that every node is truthful. If a node advertises that it has the shortest and the freshest path to the destination, other nodes may trust it.

A dangerous attack in AODV is a black hole attack; which is a sort of denial of service attack. A malicious node can absorb all network packets by falsely claiming that it has the freshest and the shortest path to the destination. Once an adversary node receives a RREQ packet, it instantly sends a RREP packet back to the source node through the reverse path with the highest sequence number as the freshest path and sets hop count to one as the shortest path. Receiving RREP packet by intermediate node may lead to update its routing table with forged information and then sends back to the source. By receiving the source node, it forwards all its packets to the malicious node as the next hop. Once, the adversary node receives the packets, it simply drops all of them. Black hole attack has two major drawbacks:

- Propagating the falsely information in the network by sending a faked route reply packet.
- Dropping all received data packets instead of forwarding.

In this paper, we present a new approach based on learning automata to detect this attack. The rest of this paper is organized as follow: Related works on black hole attack detection is studied in section 2. In section 3, network model, attack model and our solution to detect the attack is described. In section 4, we present the simulation results and discuss on them. Section 5 provides our conclusion and future works about the proposed algorithms.

II. RELATED WORK

In [5], authors present a secure version of AODV to countermeasure toward the black hole attack based on the additional control packets, which are called *Further Route Request* and *Further Route Reply*. When an intermediate node sends a RREP to the source node, the source will send a further route reply to the next hop of the sender RREP node to verify whether it has really a route to the destination node or not. The node will send back a Further Route Reply which contains checking results. If the next node has no route to the intermediate node and the destination, the source node broadcasts an alarm packet to the whole network and initiates a new route request. In the case that the next node has no route to the intermediate node but it has a route to the destination, the source node forwards its packets through this node. The main drawback of this scheme is that if the intermediate node is far away from the source, the overall delay of network will be increased.

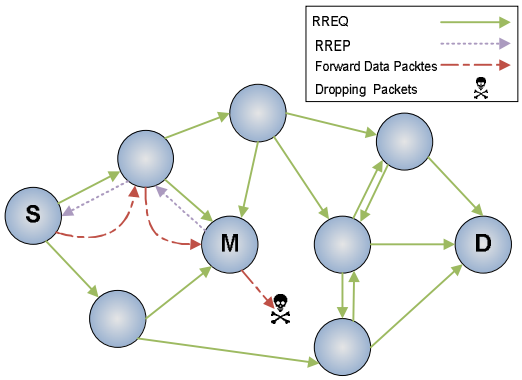

Fig. 1: The adversary node in the network responds any received RREQ by false RREP which it claims having the freshest and shortest path to the destination. When data packets are received, it simply drops them.

In [6], authors developed a neighbor-based method to detect black hole attack. Once the routing discovery phase is finished, source node requests the destination to send its neighbor set. After the source node receives the neighbor set information, it calculates the difference between them. If difference is larger than the predefined threshold, it finds that the network is exposed to black hole attack. Then, the source node authenticates the destination node using a cryptography based method and sends a control packet to correct the path.

In [7], Hollic et al. present an analytical model of the AODV route acquisition process. The proposed model can predict the probability density function (PDF) of the estimated route length. The model is extended in [8] to classify node misbehaviors. In this scheme, all destination nodes which are farther than half of the maximum hop count away from the source node will be determined as malicious node probably. The authors claimed that for distance greater than that value, number of correct routes to the destination will decrease and malicious nodes will answer the RREQ packet with a higher probability. In the other words if destination nodes are located far away from the source nodes, the false positive will be raised.

Authors of [9] proposed two solutions to detect the black hole attack. In the first solution, a path will be selected among all received routes, in terms of shared hops. From the shared

hops the source node can recognize the safe route to the destination. The main drawback of this approach is to force more delay on the network. In the second solution, each node stores the last-packet-sequence-numbers for the last packet sent to each node and the last-packet-sequence-numbers for the last packet received from each node. The received RREP contains last-packet-sequence-numbers received from the source node. According to the sequence number, the source node can detect the malicious RREP.

In [10], authors proposed a solution to identify black hole attack using anomaly detection. They used three features: the number of RREQ packets, the number of received RREP packets and the average difference between the destination sequence number in the received RREQ packet and a list in the node in each interval. The mean vector$\overline{x}^D$is calculated as $d(x) = \|x - \overline{x}^D\|$. Where D represents the training data set for N time slots and x is . When the d(x), is larger than the threshold T_h, then it will be considered as an attacker. This threshold is taken from $T_h = d(x_I), where\ I = arg_i\ max_{x_i \in D}^{d(x_i)}$. The initial mean vector is calculated in initial time $\Delta T0$, and it will be used to detect the next time interval. If ΔT is judged as normal, the corresponding data set will be used as learning data set. Otherwise, it will be used as data including attack and it will be discarded. This process will be repeated after each time interval ΔT.

Authors of [11] proposed an authentication mechanism based on the hash function, message authentication code (MAC) and pseudo random function (PRF) for detecting black hole attack. In this work, RREP packets are signed and encrypted by a sharing secret key. Each node obtains its key by selecting a random number and recursively applying pseudo random function. Also, nodes have to generate a timestamp in the RREP packets for validating the packet in the destination node. Each node receives a RREP packet; it firstly decrypts and authenticates the packet by its key. Then, it validates timestamp to ensure that it is in a reasonable time delay range. This scheme needs a time synchronization mechanism.

In [12], authors proposed a game theoretic approach for identifying the black hole attack. They used two-player non-cooperative non-zero sum game between MANET and adversary nodes. Also, authors of [13] prevented black hole attack through the IDS.

III. LEARNING AUTOMATA

In this section, we briefly introduce the basic concept of learning automata and describe how it works.

Learning automata [14] is a machine which operates in a random environment. It tries to learn adopting itself with the environment, using feedback. Each automaton has a finite set of actions and each action has a certain probability that is updated according to the feedbacks received from environment. Feedbacks are in the terms of reward and punishment. If the automaton performs an action correctly it gets reward, otherwise it will be punished.

According to the received feedbacks from environment, each automaton updates its action probabilities which finally impacts on choosing the future action. The main goal is that automata learn to select the best action out of their finite action list. Therefore, the best action is the one that maximizes probability of getting reward from the environment. Figure 2 illustrates how learning automata work.

The environment is modeled as a triple $E = \{\alpha, \beta, \gamma\}$, where $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ represents a set of inputs, $\beta = \{\beta_1, \beta_2, \dots, \beta_r\}$ represents the set of output and $\gamma = \{\gamma_1, \gamma_2, \dots, \gamma_r\}$ represents the set of penalty probability, where each member of γ is related to one input of action α_i. That environment in which β can take only binary values 0 or 1 is referred to as P-Model environment. In this kind of environment, $\beta_1 = 1$ is considered as punishment and $\beta_2 = 0$ as reward. A further generalization of the environment allows finite output set with more than two elements that take values in the interval [0,1]. Such an environment is referred to as Q-Model. Finally, when the output of environment is a contiguous random value in the interval [0,1], it is referred as S-Model.

Learning automata are categorized into fixed-structure stochastic and variable-structure stochastic. This paper uses variable-structure stochastic automata. Variable-structure stochastic automata can be defined as a quadruple (α, P, β, T) where: $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_r)$ represents the action set of the automata, $\beta = (\beta_1, \beta_2, \cdots, \beta_r)$ represents the input set, $P = (P_1, P_2, \cdots, P_r)$ represents the action probability set, and $P(n+1) = T[\alpha(n), \beta(n), P(n)]$ represents the learning algorithm. Based on the action probability set P, automaton randomly chooses an action α_i , and apply it on the environment. Automaton will update its action probability set after receiving environment's feedback. Equation (1) is for favorable responses and equation (2) is for unfavorable ones.

$$p_i(n+1) = p_i(n) + a.\big(1 - p_i(n)\big) \qquad (1)$$
$$p_j(n+1) = p_j(n) - a.p_j(n) \qquad \forall\, j \neq i$$

$$p_i(n+1) = (1-b).p_i(n) \qquad (2)$$
$$p_j(n+1) = \frac{b}{1-r} + (1-b).p_j(n) \qquad \forall\, j \neq i$$

In these two equations, a and b are reward and penalty parameters respectively. Learning algorithm is called L_{RP} (Linear Reward-Penalty) if $a = b$, for $a \ll b$, it is called L_{ReP} (Linear Reward epsilon Penalty) and for $b = 0$, it is called L_{RI} (Linear Reward Inaction). In this paper, we use L_{RP} algorithm.

In this paper, the actions set of the automata include:

- Transmitting data through the first received path.
- Transmitting data through other path, if the current path is under attack.

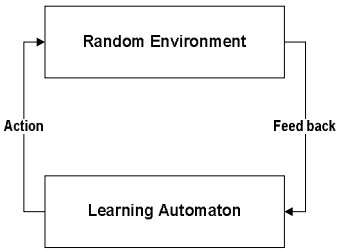

Fig. 2: The Automaton Environment Feedback Loop.

IV. DETETCTION ALGORITHM

In this section, first we describe network and attack model, then behavior of the malicious node will be studied and then by considering the behavior of adversary node the approach to detect black hole attack based on learning automata will be presented.

A. Network model

Two different kinds of nodes are randomly and uniformly distributed in the network, including normal nodes and adversary nodes. All nodes are distributed in a rectangular area. It is assumed that all links between nodes are bidirectional and all nodes in the network have the same transmission range. Hence, each node can discover all immediate neighbors which are positioned in its transmission range. Also, it is assumed that each node has no information about the entire network topology. It is supposed that nodes in the network have no mobility. It is assumed that, each node is identified by a unique identifier throughout the network.

B. Attack model

Considering that wireless ad hoc networks are deployed in an unattended environment, due to lack of physical protection, the network is susceptible toward two kind of attack model: outsider and insider [20].
In an outsider attack model, a malicious node can only access the transmission channel and it has not any secret information of network such as secret key. While in an insider attack model, once a node gets compromised, all information even secret information stored in it might be exposed to the attacker. In this paper we assume the attack model is outsider.

C. The behavior of the adversary node

Each adversary node tries to damage network by dropping data packets and propagating false information in the network. Identifying the behavior of the adversary node in the network will help us to distinguish between misbehaviors and normal behaviors of nodes.

The malicious node in the network has the following behaviors in different situations:

- A malicious node never sends a data packet to a particular destination.
- A malicious node never forwards any received data packet.

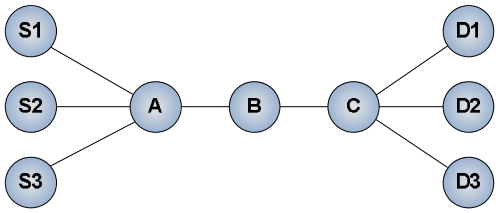

Fig. 3: nodes S1, S2 and S3 are source nodes, D1, D2 and D3 are destination nodes. Node C always sends back *RREP*. By advancing time, node B will suspect node C. While, node C is a normal node.

- A malicious node will drop received data packets..
- Since, the adversary node has no data to send, then it never initiates the route discovery phase. In the other words, it will not send any *RREQ* packet.
- Each adversary node responses to every received *RREQ* by sending a *RREP* packet and does not forward any *RREQ*.
- A malicious node always sends the *RREP* packet.

D. Algorithm Description

Every node in the network has a list of its immediate neighbors. It is assumed that all nodes are trusted and each node trusts all neighbors initially. Each node keeps a value of trust which is assigned to each of its neighbors and implies the reliability degree. The value is chosen from the interval[0,1]. The initial value of trust for each neighbors are 1. As we mentioned before, each node in the network trust all its immediate neighbors. Hence, it is supposed that all nodes have normal behavior in the network. Nodes update the value of trust associated to their neighbors by receiving feedback from the network. The corresponding value of reward therefore punish in this scheme is equal.

For each neighbor's of a node, there is a learning automaton that can calculate the degree of confidence in the neighbor. By considering the degree of confidence in any of the neighbors, the act of sending or not sending the node is selected by the automata. In fact, the probability of selecting each of the automata is arranged according to their degree of confidence .In other words; the probability of selecting action-send is proportionate with degree of confidence. Finally, regarding feedback environment, the degree of confidence in any of the neighboring nodes are updated, this process are going to identification adversary nodes. After calculating the confidence of each node, the confidence of path can be considered as minimum degree of confidence in nodes.

When a node has a packet to send and it has no knowledge of the destination node in its routing table, it starts the route discovery phase by sending a *RREQ* packet. If any node has information about the destination, it will send back a *RREP* packet. Otherwise, it forwards the received *RREQ*, until *RREQ* is received by the destination node or a node that knows how can reach to the destination. Since, the adversary node never sends or forwards a *RREQ*, each node that receives *RREQ* will

reward the sender node of that *RREQ*. Also, receiving RREP by the nodes leads to considering punishment for the sender node.

This is due to the fact, sending *RREP* is one of the actions which will be selected by adversary node when a *RREQ* packet is received.

When a data packet is received from a node, receiving node will reward sender. Since, the malicious node will never forward or send any data packet. However, sending or forwarding data packets to a node as the next hop leads to punish that node if it is not the destination node. The node has no knowledge whether other nodes will forward the packet or not.

Unfortunately, all nodes whether they are adversary or not will be punished when they send or forward the *RREP*, after a while, the value of trust for them will be less than the threshold ΔT. This scenario is illustrated in Fig. 3.

To solve this problem, each node after receiving K data packets, will sends back the ACK packet which contains number of received packets from the source. Receiving ACK from a node means that it has forwarded data packets to the destination. In result, each node in the reverse path rewards that node which has forwarded data packet to it.

If there is an adversary node in the network, receiving *RREQ*, the attacker will respond immediately by sending back a *RREP* packet which implies having the freshest and shortest path to the destination. All nodes that receive the *RREP* packet will punish the sender. Also, the attacker will drop all received data packets. In result, no ACK will be send back by the destination. Punishing by neighbor nodes will be resulted to decrease the trust value less than the thresholdΔT. In this case, the adversary node will be marked as a malicious node and an alarm will be fired by neighbors.

However, in this scheme, all intermediate nodes will be marked as an adversary node. Since, no ACK is received from the destination, all nodes in the path will suspect the next node. To address this problem, in the proposed scheme three nodes can send ACK: the destination node, the node which sends back $RREP$ and the first node in the reverse path which receives the $RREP$ (Fig. 4).

Note that, when one of these nodes sent *ACK* packet, other nodes don't send if they have sent or received that before. Also, they don't forward same received packet to save bandwidth. It is supposed that adversary nodes can't generate alarm and *ACK* packets to launch a new attack.

V. SIMULATION AND DISCUSSION

This section reports the simulation results on proposed scheme. We've used NS-2 [15] (Network Simulator) to simulate our network and AODV is used as the routing protocol. The simulation parameters are summarized in table 1. The simulation is done to evaluate the performance of network parameters. Considered metrics are as below:

- Packet Delivery Ratio: implies the packets that are sent from source node and delivered to the destination.
- Control Overhead: The overhead of routing control packets to detect the attack when the detection algorithm is applied.
- False Positive Probability: The probability that the proposed scheme detects a normal node as an adversary node.

Fig. 5 illustrates the packet delivery ratio in the presence of adversary nodes. As it is shown, data packet delivery ratio is as a function of simulation time. When there is no malicious node in the network, average packet delivery is 0.78. Some packets may be dropped due to congestion, collision. However, the ratio is decrease when network is under attack. In fact, adversary nodes try to drop all incoming data packets. Once the proposed scheme is applied, the

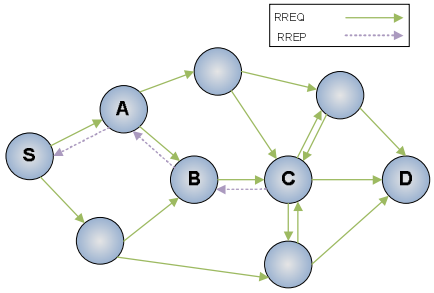

Fig. 4: Three nodes can send : Node D as the destination node, node C as the sender *RREP*, node B as the first node in the reverse path which it received the *RREP*.

Table 1: Simulation Parameters

Parameter	Value
Routing Protocol	AODV
Mac	IEEE 802.11
Terrain Area	$500m \times 500m$
Transmission Range	$100m$
Number of Nodes	50
Number of Attackers	10
Traffic Type	CBR
Packet Size	512 Kb
Rate	20pkts/s
Simulation Time	10,000 seconds
Reward, Punish value	0.01 - 0.2
Initial value of trust	1.0
ΔT	0.33

overall network throughput will be increased. In Fig. 6, the standard AODV is used as a baseline to compare with detection algorithm. As it is illustrated, the control overhead is increased. This is due to sending back *ACK* from nodes. We study and measure the false positive probability when nodes in the network detect a normal node as an adversary node. In most cases, the algorithm can detect correctly attacks with probability of 0.84. As it is shown in Fig. 7, the effect of the value of reward and punish on false positive probability is illustrated.

VI. Conclusion and Future Works

In this work we have studied the AODV routing protocol, black hole attack. We have presented a novel approach to detect the attack by considering the learning automata approach. The attack could be detected with a low false positive probability. Future work includes extending this work for MANET. Also, we would like to extend the proposed scheme for detection of all kind of packet dropping in wireless networks.

References

[1] C. Karlof and D. Wagner, "Secure Routing in Sensor Networks: Attacks and Countermeasures," in Proc. 1st IEEE International Workshop on Sensor Network Protocols and Applications, (SNPA 2003), pp. 113-127, May 2003.

[2] L. Abusalah, A. Khokhar, M. Guizani, "A survey of secure mobile Ad Hoc routing protocols," Communications Surveys & Tutorials, IEEE, vol 10, issue 4, pp. 78-93, 2008.

[3] C.E. Perkins and E. Royer, "Ad hoc on-demand distance vector routing," In Proc. of IEEE Workshop on Mobile Computing Systems and Applications, pp. 90-100, Febraury 1999.

[4] Perkins, C.E, Bhagwat,P: "Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers," Computer Communication, vol. 24, no. 4, pp.234-244, October 1994.

[5] H. Deng, W. Li; D.P. Agrawal, "Routing security in wireless ad hoc networks," IEEE Communications Magazine, vol 40, Issue 10, pp.70 - 75, Oct 2002

[6] B. Sun, Y. Guan, J. Chen and U. W.Pooch, "Detecting blackhole attack in mobile ad hoc networks," in Proc. 5th European Personal Mobile Communications Conference, pp. 490-495, Apr 2003.

[7] M. Hollick, J. Schmitt, C.Seipl and R.Steinmetz, "The ad hoc on demand distance vector protocol: an analytical model of the route acquisition process," in Proc. of Second Intl Conference on Wired/Wireless Internet Communications (WWIC'04), Frankfurt, pp. 201-212, Feb 2004.

[8] M. Hollick, J. Schmitt, C. Seipl and R.Steinmetz, "On the effect of node misbehavior in ad hoc networks," in Proc. Of IEEE Intl Conference on Communications (ICC'04), Paris, pp. 3759-3763, June 2004.

[9] Al-Shurman, M. Yoo, S. Park, "Black hole attack in Mobile Ad Hoc Networks," in Proc. ACM Southeast Regional Conference, pp. 96-97, 2004.

[10] S. Kurosawa, H. Nakayama, N. Kato, A. Jamalipour and Y. Nemoto, "Detecting black hole attack on AODV-based mobile ad hoc networks by Dynamic Learning Method," Intl Journal of Network Security, vol 5, no. 3 , pp. 338-346, Nov. 2007.

[11] J. Luo, M. Fan, and D. Ye, "Black Hole Attack Prevention Based on Authentication Mechanism", in Proc. of 11th IEEE Singapore Intl Conference on Communication Systems (ICCS 2008), Singapore, pp. 173-177, 2008

[12] E. A. Panaousis, C. Politis, "A Game Theoretic Approach for Securing AODV in Emergency Mobile Ad Hoc Networks", in Proc. of 9th Intl Workshop on Wireless Local Networks (WLN 2009), Zürich, Switzerland, pp. 985-992, 2009.

[13] Ming-Yang Su, "Prevention of selective black hole attacks on mobile ad hoc networks through intrusion detection systems", Computer Communications vol. 34, pp. 107–117, 2011.

[14] K. S. Narendra and M. A. L. Thathachar, Learning Automata, Prentice Hall, 1989.

[15] Network Simulator 2. http://isi.edu/nsnam/ns/.

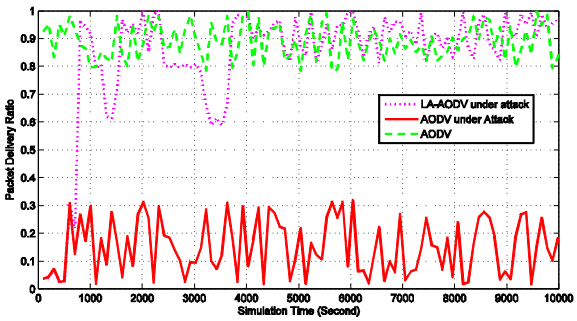

Fig. 5: Packet Delivery Ratio vs. simulation time

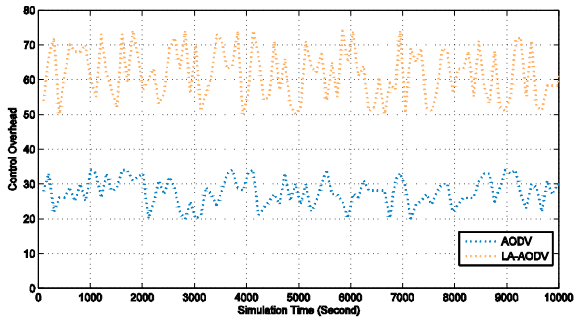

Fig. 6: Control overhead vs. simulation time

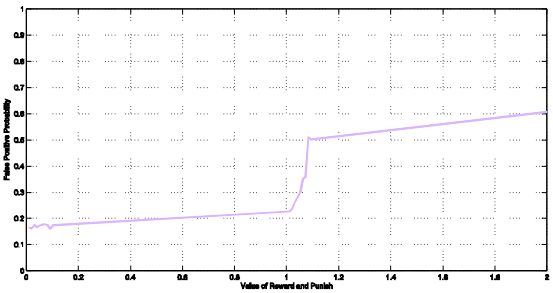

Fig. 7: False Positive Probability vs. value of reward and punish