Journal of mathematics and computer Science 15 (2015) 146 - 153

Contents list available at JMCS

Journal of Mathematics and Computer Science

Journal Homepage: www.tjmcs.com

A Hybrid Mechanism to Detection and Isolation Sinkhole Node for MANETs DSR Protocol

Azam Davahli[1,*], Hamed Rezvani Hamedani[2], Molood Noghrehabadi[3], Media Aminian[4]

[1]Young Researcher and Elite Club, Islamic Azad University, Qom Branch, Qom, Iran.
[*] (corresponding author) *azam.davahli@yahoo.com*
[2] Islamic Azad University, Qom Branch, Qom, Iran.
rezvani.eng61@gmail.com
[3]Department of Computer, Islamic Azad University, Dezful Branch, Dezful, Iran.
molood_47@yahoo.com
[4]Computer Department, Islamic Azad University, Science and Research Branch, Kerman, Iran
media.aminian@yahoo.com

Article history:
Received November 2014
Accepted February 2015
Available online February 2015

Abstract

A mobile ad hoc network is a group of wireless mobile nodes with a self-organizing system. A malicious node exhibits some unusual behavior, like sending bogus RREQs to generate sinkhole attacks. In this paper, we have presented a hybrid mechanism with three phases that are following: sinkhole indicator, detection and isolation. First, our approach finds the path where the sinkhole node is located on it. Then, the node that detects the existence of the sinkhole node will probe the other nodes to detect if any of them fail to perform the forwarding function. The proposed algorithm has lower overhead and higher time detection than other described algorithms in this paper.

Keywords: Ad-hoc Networks, MAODV protocol, Multicast Routing Protocols.

1. Introduction

A mobile ad hoc network is a group of wireless mobile nodes with a self-organizing system. In this network, mobile nodes communicate with each other via wireless links with no fixed infrastructure or centralized administration such as base station or access points, and cooperate by forwarding packets for each other. Ad-hoc networks are used in many cases because their implementation is quick and easy and also they are independent of prefabricated structures [1].

Independent Intrusion detection in MANETs is challenging for a number of reasons [2, 3]. IDS is a protective system that can detect disorders occurring on the network. The procedure goes as intrusion detection can report and control occurred disorders through steps including collecting data, seeking ports, controlling computers, and finally hacking. So, intrusion detection can report control intrusion sabotage that composed of phases collecting data, probing port, gaining computer's control and finally hacking [4]. MANET networks change their topologies dynamically due to node mobility, so they are susceptible to malicious manipulation. A malicious node [5] exhibits some unusual behavior, like sending bogus RREQs to generate sinkhole attacks. Several algorithms have been proposed to detect these malicious nodes in MANETs using the evidence of malicious behaviors. In this paper, we have presented a hybrid mechanism that uses two different techniques, in which we focus mainly on the low overhead and low detection time. The mechanism is an improvement to [6] and [7]. We show that distributing inherent of probing technique can decreases the detection time. We also, introduce a black list that maintain by nodes. By this black list, malicious nodes cannot participate in routing process any more. However, our algorithm uses collaborative efforts of nodes in the neighborhood to detect a malicious node, and it is dependent on DSR routing protocol.

2. Related Works

The method that has presented in [8] acts on the DSR protocol and only employs first-hand information. The watchdog locating at each node keeps a faulty-list. All the neighbor nodes detected as malicious are put into the faulty-list. Before deciding whether to forward the packet for a neighbor, the node checks its faulty-list. If the neighbor lies in its faulty-list, the node denies the request.

There are two algorithms in [9,10] that can be used in different cliques (clusters)[11]. Each clique (cluster) will come up with information about the malicious nodes, if any. It can then use this information for isolating these malicious nodes from itself. Moreover, this information may be sent to other cliques (clusters), so that they can also isolate the malicious nodes from themselves.

Albers et al. [12] proposed a distributed and collaborative architecture of IDS by using mobile agents. A Local Intrusion Detection System (LIDS) is implemented on every node for local concern, which can be extended for global concern by cooperating with other LIDS. Two types of data are exchanged among LIDS: security data (to obtain complementary information from collaborating nodes) and intrusion alerts (to inform others of locally detected intrusion). In order to analyze the possible intrusion, data must be obtained from what the LIDS detects on, along with additional information from other nodes. Other LIDS might be run on different operating systems or use data from different activities such as system, application, or network activities; therefore, the format of this raw data might be different, which makes it hard for LIDS to analyze. However, such difficulties can be solved by using Simple Network Management Protocol (SNMP) data located in Management Information Base (MIBs) as an audit data source. Such a data source not only eliminates those difficulties, but also reduces the increase in using additional resources to collect audit data if an SNMP agent is already run on each node. For the methodology of detection, Local IDS Agent can use either anomaly or misuse detection. However, the combination of two mechanisms will offer the better model. Once the local intrusion is detected, the LIDS initiate a response and informs the other nodes in the network. Upon receiving an alert, the LIDS can protect itself against the intrusion.

3. Sinkhole Attack

Sinkhole node is a node that alters the routing messages and gathers the traffic of the network. In DSR protocol when a node decides to send a packet to some other node, it firstly obtains a source route by searching its cache of routes previously learned. If it does not find a route, the node creates a Route Request (RREQ) and sends it to its neighbors. The RREQ contains the initiator id and target id of the Route discovery. This message also consists of a sequence number that determined by the sender of RREQ. Each node in network that receives RREQ, it searches its route cache, if it does not find a route to

the destination, appends its Id to the RREQ and sends it to the next hop, else it returns a Route Reply message(RREP) to the sender. A node learns the source routes from received RREQs and RREPs [13]. Figure1 illustrates a route discovery.

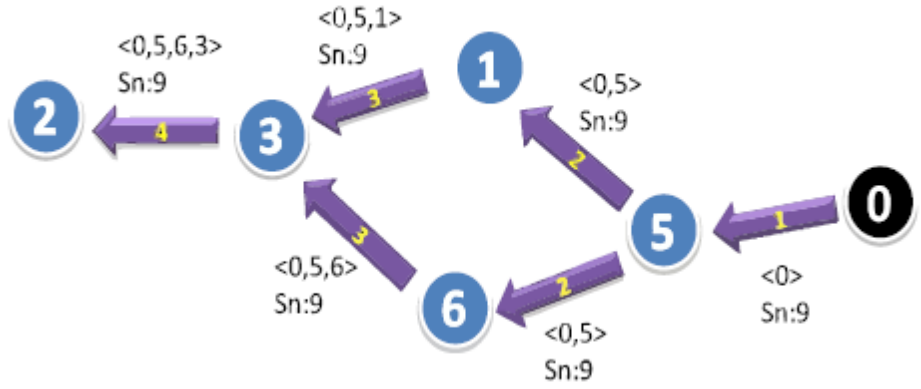

Figure 1. An example of propagation of RREQ. Source node: 0, sequence number: 9.

Sequence Number (SN) in the RREQ refers to the freshness of the RREQ. When a node generates an RREQ, it increases the SN by one. Every time, a node receives an RREQ, it compares the SN of RREQ to previous sequence number. If it has processed this RREQ, it will discard it. Else the node appends its id to the RREQ and re-broadcasts it. When a sinkhole node receives an RREQ, it makes changes in the message. The sinkhole node appends its id and its target to the RREQ and on the other hand it changes the SN into a higher number than the current SN. Then it propagates the bogus RREQ in the network. Because of this high sequence number, each node that receives this RREQ discards the other RREQs with smaller sequence number and stores the route in the bogus RREQ in its route cache. After that, each node that wants to send a packet uses this bogus route. Consequently, the packets will concentrate in the sinkhole node. Therefore, a sinkhole attack increases network overhead, decreases the network's lifetime by boosting energy consumption, and finally destroys the network [9]. Figure2 shows the propagation of a bogus RREQ in the network. In this figure, node 2 receives the RREQ and appends its id to the RREQ after the source node ID in the RREQ and puts the SN equal to a high number like 700.

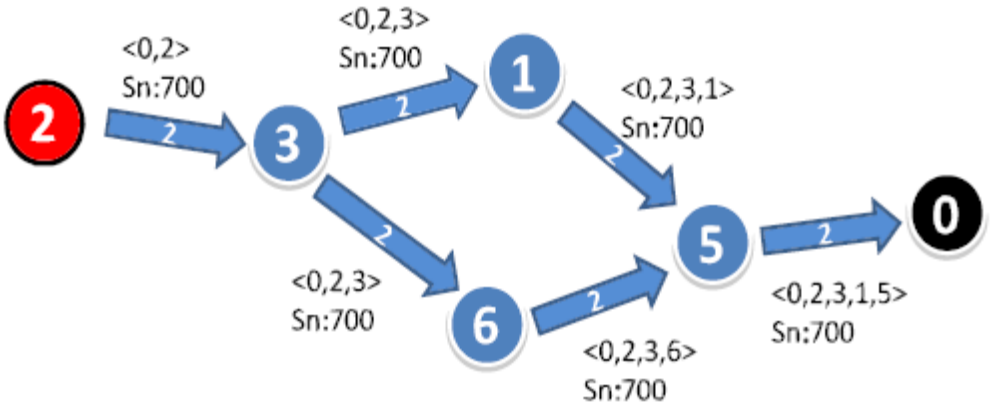

Figure 2. An example of propagation of a bogus RREQ. Sinkhole node: 2 , target node: 0.

4. Main Algorithms

4.1. Probing technique

Probing technique has presented in[6].This method is a distributed detecting technique to detect the malicious packet dropping attack in MANETs, in which each node monitors the other node's behavior. The detecting algorithm probes all the paths to the destination. For detecting the malicious node, a detecting node sends a probe message to all nodes on a given path. When a node receives this message, send back an acknowledgment message to the sender. Therefore, if the sender dos not receive the acknowledgement from a given node, it assumes that the node is a malicious node. There are at least two ways of probing. One way is to probe from the farthest node to the nearest. The other way is to probe from the nearest node to the farthest. In the first way, detecting node starts to send the probe message from the farthest node in the path to the nearest node (figure3). This way is better when the malicious node is located far from the detecting node and the malicious node can be detected by minimum probe messages and if there is no malicious node on the path by one probe message, we can prove the goodness of all the immediate nodes.

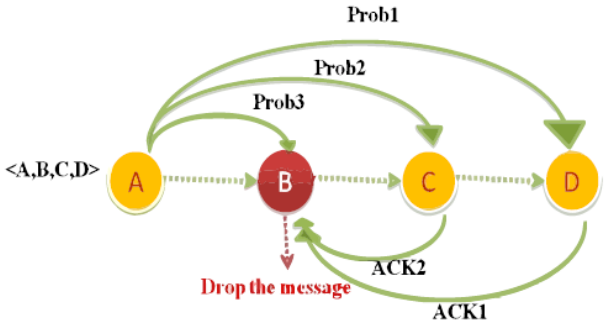

Figure 3. An example of far to near probing. Malicious node: B.

In the second way, detecting node starts to send the probe message from the nearest node in the path to the farthest node (figure4). When we are using this way to detecting the malicious node, if the malicious node is located far from the detecting node, it will be detected by the maximum probe messages. Consequently, the overhead will be increased. The disadvantage of this technique is that the malicious nodes will not be isolated after detecting and they can participate in forwarding data packets processes again.

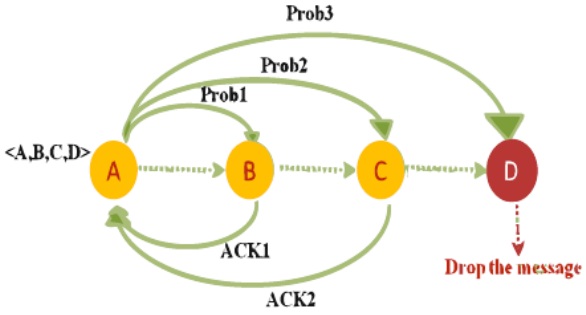

Figure 4. An example of near to far probing. Malicious node: D.

4.2. A Cooperative-Sinkhole detection method

This technique uses two phases for finding sinkhole nodes in the network. In the first phase, nodes cooperate to recognize the existence of a sinkhole node in the network. When a sinkhole node is detected in the network, the detection phase will start. The sinkhole detection algorithm is initiated by broadcasting

a ”sinkhole alarm packet” (SAP). Then, the sinkhole detection algorithm tries to detect and isolate a sinkhole node by broadcasting a ”sinkhole detection packet” (SDP) and ”sinkhole node packet” (SNP)[7]. The disadvantage of this technique is that the proposed algorithm broadcasts SAP, SDP, and SNP, causing increased network overhead and the overhead increases the energy consumption in the nodes. But in other hand, it has high detection rate.

5. Proposed Algorithm

The proposed sinkhole detection algorithm is composed of three phases: detection sinkhole indicator, detection sinkhole node and isolation the malicious node. When a sinkhole indicator is detected, the sinkhole detection algorithm is initiated by broadcasting”probe packets” on the sinkhole path. When the sinkhole detection algorithm recognizes the sinkhole node, tries to isolate the sinkhole node by broadcasting a ’warning packet’. Each node receives the warning message, it adds the id in the warning message in its black list to prevent the sinkhole participates in route discovery.

5.1. Sinkhole Indicator

When a node receives an RREQ, if it finds its Id in the RREQ’s source route or if the sequence number of the RREQ is smaller than the previous sequence number of the source of the RREQ; it discards the RREQ. If a node receives an RREQ whose source id is equal to the id of the receiving node, it checks the sequence number. If the sequence number of the RREQ is greater than the current sequence number of the node, then the node recognizes the existence of sinkhole and this RREQ is from the malicious node. Hence, we can conclude that there is a sinkhole node in the route path of the RREQ and that the nodes in the route path are sinkhole candidates [8]. Figure5 has presented the sinkhole indicator algorithm.

```
After receiving the RREQ, checks the packet:
if sender's ID is in the black list then
    Drop the packet;
end
if finds it's ID in this packet then
    if bogus SN > current SN then
        Starts the detection algorithm;
    else
        Drop the packet;
    end
else
    append its ID to the packet and rebroadcast
    it.
end
```

Figure 5. Sinkhole Indicator Algorithm.

5.2. Detection and Isolation phases

Detecting node is the node that is aware of the existence of a sinkhole node in the network. This node initiates the detection phase. Our approach uses far to near probing, that’s why detecting node sends a probe message to the last node on the sinkhole route, if it does not receive the acknowledgment from the given node, it will send the probe to the next hop. It repeats this until it receives an acknowledgment. Hence, it suspects to the previous node. To prevent a sinkhole node to participate in route discovery, detecting node generates a warning message and put the malicious node’s id and the current SN in the message. Then it broadcasts the message to the network. Figure 6 shows the detection algorithm.

```
Initiator node sends a probe msg to the last nod
in the sinkhole route
Each node receives the probe msg returns an
ACK
while initiator node receives an ACK do
    Sends probe to the next hop.;
end
Add the previous node's id to the warning msg
and rebroadcast it.;
```

Figure 6. Detection Sinkhole Indicator algorithm.

Each node receives the warning message, it modifies the stored sequence number of the source according to current sequence number in the warning message. Then, it deletes all paths that contain the sinkhole node's id from its route cache and adds this id to its black list. Hence, when a node receives a RREQ, it looks for the sender's id in its black list. If it can find the id in its black list, it will discard the RREQ. Otherwise, it processes the RREQ for routing discovery. Figure7 presents the forwarding RREQ algorithm.

```
After the receiving the RREQ, node checks its
black list
if the id exists in the black list then
    Drop the RREQ;
else
    Initiate detection sinkhole indicator
    algorithm;
end
```

Figure 7. Forwarding RREQ Algorithm.

6. Simulation

In our simulation with ns2, we investigated the performance of the proposed protocol in a network with 30 mobile nodes placed randomly within a 100*100 areas, 2Mbps bandwidth and 20m Transmission range.

Figure 8-9 compared performance of [6] and the proposed algorithm by two classical indexes: overhead and detection time. Overhead: the ratio of routing control packets to data packets. Detection time: the time between when a sinkhole node sends a fake RREQ and when the target node detects the sinkhole node.

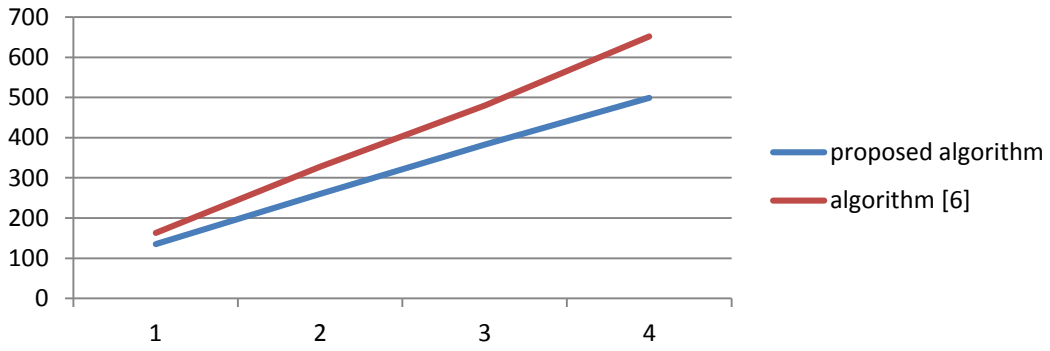

Figure 8. compared performance of [6] and the proposed algorithm base on overhead.

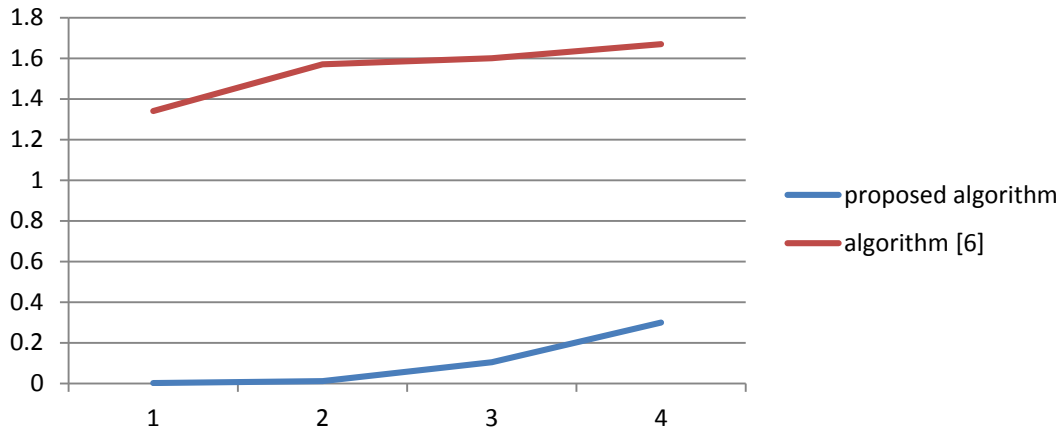

Figure 9. compared performance of [6] and the proposed algorithm base on detection time.

7. Conclusion

We proposed a method which can find sinkhole nodes. Always, sinkhole node is located near the target node in the end of the discovered route. That's why, our approach uses the far to near probing to find the sinkhole node and uses a warning message to make the other nodes aware of existence of a sinkhole node in the network and each node maintains a black list that sinkhole node's id is located in it. Because of using far to near probing, we can find the sinkhole node by minimum probe messages. This approach decreases the overhead in the network and the produced overhead in this technique is very smaller than the overhead in the cooperative-sinkhole detection method. The disadvantage is that an intelligent attacker may be able to avoid detection by forwarding all packets (including probe messages destined to the downstream nodes and acknowledgments) for a certain period of time immediately after receiving a probe message for itself. This proposed method is good because the sinkhole node only gathers the data packet in itself and does not return back an acknowledgment to the detecting node.

References

[1] A. Izadi, A. Sahab, J. Vahidi, "*A New Mechanism for Traffic Reduction the Service/Resource Discovery Protocol in Ad-Hoc Grid Network*", Journal of Mathematics and Computer Science, Vol. 6 (2013), pp. 129–138.

[2] P. Brutch, C. Ko, "*Challenges in intrusion detection for Ad Hoc networks*", Network Associates Laboratories, London, Applications and the Internet Workshops, (2003) pp. 368 – 373.

[3] K. Wrona, "*distributed security: Ad Hoc Networks and Beyond*", PAMPAS Workshop, London, (2002).

[4] M.M. Javidi, M.H. Nattaj, "*A New and Quick Method to Detect DoS Attacks by Neural Networks*", Journal of Mathematics and Computer Science, Vol. 6 (2013), pp. 85–96.

[5] M. Jain, M.P.S Bhatia, "*A Rough Set Based Approach to Classify Node Behavior in Mobile Ad Hoc Networks* ", Journal of Mathematics and Computer Science, Vol. 11 (2014), pp. 64–78.

[6] J. Mike, "*Resisting Malicious Packet Dropping in Wireless Ad Hoc Networks*", ADHOC-NOW 2856 (2003), pp.151-163.

[7] K. Gisung, "*A Cooperative-Sinkhole detection method for mobile ad hoc network*", Electronics and communication Vol. 64 (2010), pp.390-397.

[8] J.GUO, "*HEAD: A Hybrid Mechanism of enforce node cooperation in Mobile Ad Hoc Network*", Tsinghuu Science and Technology, Vol. 12 (2007).

[9] N. Marchang, R. Datta, "*Collaborative techniques for intrusion detection in mobile ad-hoc networks*", (2008) pp. 508-523.

[10] M. Nejadkheirallah, M.M. Tajari, "*Multi-hop Fuzzy Routing for Wireless Sensor Network with Mobile Sink*", Journal of Mathematics and Computer Science, Vol. 9 (2014), pp. 12–24.

[11] A. Ghorbannia Delavar, G.H. Mohebpour, "*ANR: An algorithm to recommend initial cluster centers for k-means algorithm*", Journal of Mathematics and Computer Science, Vol. 11 (2014), pp. 277–290.
[12] P. Alberts, O. Camp, "*Security in Ad Hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches*", 1st International Workshop on Wireless Information Systems, (2002) pp.1-12.
[13] D. Johnson, DA. Moltz, J. Brach, "*The dynamic source routing protocol for multi-hop wireless Ad Hoc networks*", EP C, editor, Ad Hoc Networking, Boston, (2001) pp.139-72.