Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 04, December 2021.

A MOBILE AGENT-BASED METHOD TO COUNTER SINKHOLE ATTACKS IN WIRELESS SENSOR NETWORKS

Hadi Khosravi[1] and Mohammad GhasemiGol[2]

(Received: 29-Sep.-2021, Revised: 22-Nov.-2021, Accepted: 25-Nov.-2021)

ABSTRACT

Wireless sensor networks (WSNs) are an applied technology widely used in various areas. According to the WSN limitations, they usually face many types of attacks. The sinkhole attack is the most popular and dangerous attack in the routing of WSNs. There are many approaches to counter sinkhole attacks in the literature. The mobile agent methods generate better results in facing sinkhole attacks and overcoming the WSN limitations. In this paper, we present a new mobile agent-based method that applies the trust value of each sensor to detect and prevent sinkhole attacks. We compute the trust values to inform the sensor nodes about their neighbors' reputations. As shown in the experiments, the proposed method generates better results in packet loss ratio. It also fixes the security flaws of previous works and reduces the agents' overhead in the network compared to previous methods.

KEYWORDS

Security, Trust management, Mobile agent, Sinkhole attacks, Wireless sensor networks.

1. INTRODUCTION

Wireless sensor networks (WSNs) consist of many multifunctional low-cost and small-size sensors. These nodes are deployed in an unattended environment with the capability of sensing, wireless communication and computing (i.e., the collection and dissemination of environmental data). WSN is a combination of sensing and embedded techniques, distributed information processing and communication mechanisms [7]. It has many applications in various areas and is usually deployed in a risky environment [3], [19]. Thus, security is a vital issue in WSNs. Due to the resource constraints of the nodes, applying security mechanisms in these types of networks is a very conservative task [13]. Sensor nodes are physically placed in an open environment and are unprotected. WSNs have many more constraints than traditional networks. Because of computing, resource constraints and the broadcast nature of the transmission medium, WSNs have different security challenges compared to traditional networks. Moreover, these networks are easily exposed to a variety of security attacks. In most security attacks in WSNs, the compromised sensor nodes insert fake information into the network [23]. Therefore, WSN protocols and algorithms must be self-organized and security mechanisms should be considered to protect the sensor nodes against various attacks [9].

A routing protocol is a software placed on the network layer and is responsible for decisions about the output route of packets that should be transmitted. In other words, it is an algorithm to find a way to transfer data. Typically, in WSNs, the destination node is called a base station. The distance between the source and the destination nodes may be far or even outside of the transmission range. Therefore, data may be transmitted to reach the sink node through multiple hops [25]. Different attacks have been designed and implemented based on essential tasks of the network layer, which endanger data packets' security.

In this paper, amongst various attacks in the network layer, the sinkhole attack has been chosen, which is one of the most widespread and destructive attacks. The sinkhole attack is a dangerous attack in WSNs that prevents reaching complete and correct information to the base station node. In this attack, a malicious node misleads the surrounding nodes to attract traffic from a specific path [15], [12]. As shown in Figure 1, the malicious node claims have the shortest path to the sink. The primary metric for data routing is the best path to the base station in WSNs [6]. Hence, the malicious node can absorb a portion of network packets illegally. In fact, in a sinkhole attack, a malicious node sends false information about the routing to the neighbors to encourage them to choose it as their parent to get the network traffic of

1. H. Khosravi is with the CERT Coordination Center, Univ. of Birjand, Birjand, Iran. Email: h.khosravi@cert.birjand.ac.ir
2. M. GhasemiGol (Corresponding Author) is with Department of Electrical and Computer Engineering, University of Birjand, Birjand, Iran. Email: ghasemigol@birjand.ac.ir

that area [6]. Now, it can launch other attacks, such as selective forwarding, packet drops and packet modifications [15]. Thus, the sinkhole attack decreases the network's lifetime and increases the network overhead [14].

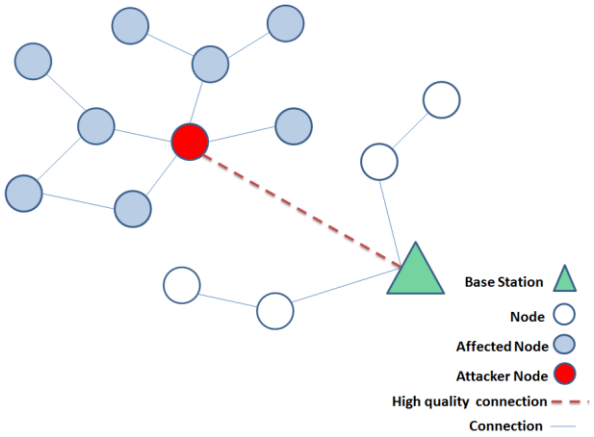

Figure 1. The sinkhole attack in a WSN [9].

There are many methods in the literature to address sinkhole attacks [1]-[2]. Here, we focus on the agent-based methods that apply mobile agents as a self-controlling program to transfer code and data between the sensor nodes [11], [16], [4]. With the aid of agents, we can reduce the communication costs by moving the processing code to the sensors instead of transferring data to a central processor node.

In this paper, we propose a new mobile agent-based method that applies the trust value of each sensor to detect and prevent sinkhole attacks. We use the trust value of each sensor to inform the sensor nodes about their neighbors' reputations. Hence, the main contributions of this paper are as follows:

- We compute the trust value of each node to inform the sensor nodes about their neighbor's reputations.
- We apply mobile agents to reduce the communication costs in WSNs by eliminating unnecessary data transfer.

The rest of this paper is organized as follows. We investigate the related work in Section 2. The proposed method is described in Section 3. In Section 4, we compare our proposed method with two related works regarding packet loss, energy consumption, throughput and agent overhead. Finally, the conclusion and the future direction are mentioned in Section 5.

2. RELATED WORK

So far, many methods have been presented to detect and prevent sinkhole attacks. In this section, we examine some of them and the related work in this area. Sheela et al. detected the sinkhole attacks in WSNs by a mobile agent-based method. The mobile agents collect information from all sensor nodes to make each node aware of the network in terms of the malicious nodes. Normal nodes do not accept the fake information of the compromised nodes. This method has agent navigation and data routing algorithms which the first algorithm describes how to visit all nodes and to give the network information to nodes by the mobile agent. Also, the second algorithm describes how to use a node of this global network information for routing data packets. An essential feature of this method is that it detects the sinkhole attack without any encryption or decryption mechanism. However, if the number of nodes increases, the overhead of this method will be very high [22].

Sharmila and Umamaheswari have presented a solution to detect sinkhole attacks using message digest. In this method, a control scheme can be built for each packet in the network by using hash functions. The digest message is calculated by the source node using a combination of MD5 and SHA1 hash algorithms and is sent through a trustable path to the base station. Then, the message is sent through a node that claims to have the shortest route to the sink. If an adversary modifies the message, it can be detected by checking the modified message and the message digest. Due to the use of SHA1 and MD5 algorithms, overhead caused by the encryption and decryption operations is high and transfer the message is transferred in the two paths causing loss of energy and traffic overhead that are disadvantages

of this method. Also, having a trustable path in this type of network is often not possible due to their nature [21].

Bahekmat et al. have presented an efficient algorithm to detect sinkhole attacks in WSNs. In their proposed algorithm, it is assumed that all nodes in the network are similar, randomly distributed and aware of their locations in the network. Each node sends a control packet to the base station directly before sending the data packet through hop-by-hop routing. If any change is made on the control fields, it indicates that there are malicious nodes in the path. The advantage of this method is the reduction of packet loss rate and energy consumption. The main disadvantage of this algorithm is that each sensor needs to use localization algorithms or have a Global Position System (GPS) in order to know its geographical location, which requires additional costs [5].

Hamedheidari and Rafeh have presented a defensive mechanism by mobile agents to counter sinkhole attacks. Each node is aware of its trusted neighbors using mobile agents with a three-step negotiation. The main purpose of the three-step trusting procedure between the node and the agent is an authentication mechanism that uses unique codes and hash algorithms. In this method, it has been assumed that all nodes are physically protected. This assumption is not very logical due to the nature of this type of network that is placed in remote areas. Now, by omitting this assumption, the attacker can gain access to the node physically. As a result, this method suffers from tampering attacks [9].

Naderi et *al*. detected the sinkhole area according to the energy consumption model in the network and the energy deviation of each node from other nodes. Also, nodes' energy information is collected and analyzed by the sink. Then, a trust evaluation mechanism is used, so that each node calculates the trust value of its neighbors. The trust mechanism starts after observing a contradiction in energy consumption in a limited area of the network and then a trust value is assigned to each node based on security requirements by the sensed event. The advantage of this method is to achieve considerable performance in factors that have higher risk. For example, a network that has more nodes and compromised nodes has a short distance to the sink and delivers more packets to the sink under challenging conditions. The major disadvantage of this method is that it acts only based on energy criteria. In other words, if a node has a higher energy consumption due to more telecommunication capabilities, it is incorrectly detected as a malicious node, which is referred to as a false positive [17].

Jahandoust and Ghassemi proposed an adaptive framework with a combination of subjective logic and an extension of timed automata to counter sinkhole attacks in WSNs. For this reason, they utilized a stochastic extension of the AODV routing algorithm. A subjective logic model is applied to detect the sinkhole nodes and find the most reliable path. Also, a probabilistic model monitors the network behavior to adaptively adjust the algorithm parameters [10].

In recent research, Nwankwo and Abdulhamid applied the ant colony method to detect sinkhole areas [18]. Although they claim that their method can improve the detection rate and false alerts, it applies ant colony as a time-consuming method which is not proper for WSN applications. Wang presented a three-layer detection scheme to monitor the heterogeneous Industrial WSN (IWSN). Unlike the previous method, their scheme does not utilize information and location information from the neighbors. At the first layer, the normal and Sybil nodes are found by a quadratic difference based on the received signal strength indicator (RSSI). The second layer continues the search for nodes detected in the first layer using a method based on residual energy. Finally, the base station detects the first and second high-energy nodes [24]. Jatti and Sonti presented an agent-based algorithm to detect and prevent sinkhole attacks in WSNs [11]. Their work is very similar to Hamedheidari and Rafeh's method [9]. They just apply their presented method to a different routing algorithm and evaluate it through network simulator NS 2.35.

In this paper, we present a new mobile agent-based method to counter sinkhole attacks. Therefore, we focus on the agent-based literature and exceed it to fix the security flaws of previous works and reduce the overhead caused by the presence of the agents in the network. We compare our proposed method with two agent-based related works to show its performance in facing sinkhole attacks.

3. PROPOSED METHOD

Here, we explain our proposed method to apply mobile agents and trust management. In our method, each node computes the trust value of its neighbors. Furthermore, it uses mobile agents to make each

node aware of the reputation values of its neighbors. Consequently, each node identifies its compromised neighbors and does not interact with them.

3.1 Agent Designing

Such as other methods, the mobile agent is an executable script that migrates from one node to another in the format of an agent packet. In the proposed method, we use a simplified type of agent code to detect the sinkhole nodes. As a result, it produces less computational overhead and decreases energy consumption in the nodes as well. Furthermore, agents do not communicate with each other and only interact with the nodes placed on them. So, no traffic overhead will exist due to the agents' communication with each other. It reduces the energy consumption in the nodes significantly [9].

3.2 Agent Migration

The migration allows an agent to move from an agent node (the node that contains the agent) to a neighbor node and back to the original node. Therefore, the agent only moves to a one-hop neighbor and does not need to maintain the agent migration path to return to the original node. As a result, the source and destination storage will suffice. We define here another action that is the agent cycling. Agent cycling is done when a mobile agent migrates to all one-hop nodes around an agent node.

3.3 Algorithm

First, nodes are randomly distributed with a uniform distribution in the network. After that, the base station selects several sensor nodes based on the expected number of agents in the network and sends agent packets to them. After receiving agents, each node sends a HELLO packet to neighbor nodes and creates a neighboring matrix. Figure 2 shows a WSN with nine nodes. In this network, node A consists of an agent and H is the malicious node. Table 1 shows the neighboring matrix of node B after sending HELLO packets. As can be seen, in this step, detected are only neighbors of a node which may be malicious.

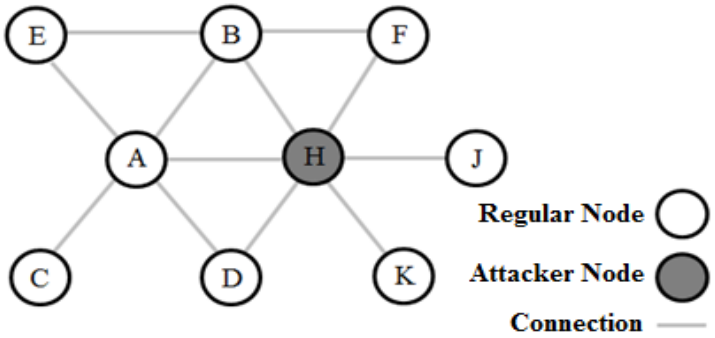

Figure 2. A WSN with nine nodes.

Table 1. Neighboring matrix of node B after sending HELLO packets.

The ID of the Neighbor Node	Node A	Node E	Node F	Node H
Sent packets	0	0	0	0
Correct packets	0	0	0	0
Sent packets 2	0	0	0	0
Trust value	50	50	50	50
Update state	0	0	0	0

Each node increases the number of sent packets variable by one after transmitting a packet to its one-hop neighbor. Next, each sender node (x) monitors its neighbor (y) for a limited time to investigate its forwarding behaviors. If node x detects the correct retransmission, it will increase the number of correct packets variable by one [8]. Before forwarding a new packet, each node checks to know whether the number of sent packets variable has reached a predefined threshold of forwarding, 100 for example, or

not. Then, the direct trust from node *x* to node *y* can be calculated by the formula and variable value 'number of sent packets' proposed in [20] as in Equation (1) 'Number of correct packets' is set to zero.

$$T_{x,y} = \frac{S_{x,y}}{S_{x,y} + U_{x,y}} \times 100 \quad (1)$$

where $T_{x,y}$ is the trust from node *x* to node *y*, $S_{x,y}$ is the number of correct packets forwarded by node *y* and $U_{x,y}$ is the number of packets dropped by node *y*. Also, to reduce the forwarded data and computational overhead, trust values are stored as unsigned integers in the range of 0 to 100 instead of decimal values that represent the lowest and the highest level of trust, respectively. To update the trust for node y at node x (i.e., trust value variable), we apply formula (1) to calculate the trust average. The trust value in node x is calculated and stored in node x as well. Moreover, to get a more accurate trust value, the reputation value is calculated by using mobile agents. For this reason, the trust values of all one-hop neighbors of an agent node are collected by the agent. The reputation values are calculated at the agent node and then new reputation values are submitted to the neighbor nodes with the help of mobile agents.

The update state variable is binary. 0 means that the last update of variable 'trust value' has been done by direct calculation of trust value by the node and one means that it has been updated by the average of provided reputation value by the agent and stored trust value in the trust value variable. At first, the agent checks the value of the update state variable. If it is 0, it submits the reputation value of the node's neighbors to it and receives the values of variables 'trust value' and 'number of sent packets 2' of the node's neighbors and delivers them to the agent node during its migration. But, if the update state variable is set to 1, the mobile agent returns to the agent node without doing any extra operation. The purpose of using the update state variable is that if a node has not been done, the adequate number of interactions (i.e., 100), it can solely update the trust value variable. So, it is better to ignore these trust values, which can prevent the computational overhead imposed by these processes.

Furthermore, this simple binary variable prevents the impact of a fake agent on a node. According to our mechanism of updating the trust value variable, if a node updates the trust value based on the reputation, the new trust value cannot update until it is already updated based on the direct trust value which is calculated by the node. As a result, if an agent hands over incorrect reputation information to a node, the impact of this fake information can be reduced by calculating the trust value based on the direct trust calculated by the node. Initialization and updating of the variable 'update state' are done only by the node. At the first step of updating the trust value variable, the direct trust value is calculated as described earlier by the node. Then, the average of this value and the one stored in the trust value will be held as the new trust value and also the value of the two variables 'number of sent packets' and 'number of correct packets' will be changed to zero. This procedure could be done if the number of interactions reaches 100. On the other hand, if the value of the update state variable is zero, it means that the agent has not read the trust value yet. Hence, the number of interactions is stored in another variable named 'number of sent packets 2' in the node in which values will be multiples of 100 (except the default value). We use this variable to weigh the collected trust values of the nodes for calculations of reputation that are done within the agent node. Three tables are stored in the agent node; table 1 (which is also stored in all other nodes), table 2 and table 3. The data used to calculate the reputation value which has been transmitted to an agent node by the mobile agent is stored in Table 2. For each neighboring node, one dedicated table 2 is stored in an agent node. The reputation value of all nodes is stored in Table 3 as well.

Table 2. Transmitted information of the mobile agent to agent node after collecting data from node B.

The ID of the Neighbor Node	Node $_A$	Node $_E$	Node $_F$	Node $_H$
Trust value	95	90	80	10
Number of sent packets 2	200	300	200	200

Table 3. Reputation table.

The ID of the Neighbor node	Node $_A$	Node $_B$	Node $_C$	Node $_D$	Node $_E$	Node $_H$
Reputation	95	80	65	80	70	15

After agent cycling, the reputation of each node is calculated by using the information gathered by the mobile agent as follows:

$$R1_b = \frac{S_{a,b} \times T_{a,b} + S_{e,b} \times T_{e,b} + S_{f,b} \times T_{f,b} + S_{h,b} \times T_{h,b}}{S_{a,b} + S_{e,b} + S_{f,b} + S_{h,b}} \quad (2)$$

$$R2_b = \frac{R_a \times T_{a,b} + R_e \times T_{e,b} + R_f \times T_{f,b} + R_h \times T_{h,b}}{R_a + R_e + R_f + R_h} \quad (3)$$

$$R_b = \frac{R1_b + R2_b}{2} \quad (4)$$

where $S_{a,b}$ is the number of sent packets from node *a* to node *b*, $T_{a,b}$ is the trust value of node *a* to node *b* or the trust value variable and R_a is the reputation value of node *a*.

Since a malicious node may try to show the number of its interactions (i.e., the '*number of sent packets 2*' variable) extraordinary high to increase the impact of its comment in a weighted mean formula, we apply the following condition with a reasonable threshold of 400, for instance, to prevent this possible disorder.

IF number of sent packets 2 >= threshold, THEN
    number of sent packets 2 = threshold

4. EXPERIMENTAL RESULTS

In this section, we express the security benefits of our proposed method compared with the Hamedheidari and Rafeh [9] and Jatti and Sonti [11] methods. We describe the security weaknesses of the previous methods and their incapability of detecting and resisting some sorts of attack. Moreover, we explain the advantages of our proposed method and the way in which we resolve these weaknesses. Then, we compare the proposed method with the mentioned two methods in terms of various standard evaluation parameters in normal conditions to see whether the proposed method, which brings superior safety, imposes more overhead than the previous methods or not.

In Hamedheidari and Rafeh's method, they assume that all nodes are physically protected. This assumption is not logical due to the nature of WSNs that are placed in risky areas. By removing this assumption, an attacker (i.e., human attacker) can take the node physically. In fact, it will gain access to *code 1*, *code 2 and* the NodeHashFunc(); so by having this information, it can create a fake node and place it in the network. In Hamedheidari and Rafeh's method, a fake node is treated as a normal node; and all actions of it are allowed, even hostile acts. By knowing this information, accessing *code 3* and creating a fake agent is not so hard; so the detection way of the base paper fails in this situation. Furthermore, the agent node multicasts the trust packet and its neighbors only check the sender ID of the packet to ensure transmitting by the agent node. Hence, an attacker can easily create a fake trust packet and pretend that the packet is sent by it *via* changing the ID of the trust packet to an agent node. Therefore, this can easily disrupt the network's ordinary workflow.

As described in detail in the proposed method section, unlike Hamedheidari and Rafeh's method, our proposed method has a reasonable performance in all the above conditions. Also, another positive point of the proposed method is that the value of the threshold to detect an attacker can be set according to the sensitivity of WSN's type. The more security is essential, the higher the threshold should be and *vice versa*.

4.1 Simulation

In this sub-section, we evaluate the performance of the proposed method within a simulation environment. For this purpose, we developed an agent-based simulator and then compared the results of our proposed method with two related works [9], [11].

4.1.1 Simulation Environment

Simulation environment has been considered to be 200 × 200 meters in simulations and we assume N sensor nodes with a uniform distribution that are randomly distributed in the environment and are mobile

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 04, December 2021.

as well. Simulations have been done for N sensor nodes from 100 to 400 with the step of 100. Simulation time is 20 minutes and the results are recorded one time every 30 seconds. Also, each experiment has been repeated for any number of sensor nodes five times and corresponding diagrams are the average of 5-time runs. In each experiment, between 10-20% of nodes are malicious. Simulations were done for each number of nodes with various percents of agents (10%, 15%, 20% and 25%) in each experiment. In table 4, the simulation conditions are shown, so that E_{elect} is consumed energy to activation electronic circuits of transmitter and E_{fs} is the activation energy amplifier of the transmitter.

Table 4. Simulation environment.

Variable	Value
Network scale	$200_m \times 200_m$
Duration	20 minutes
Routing protocol	AODV
Range of transmission	50 m
Speed	10 m/s
Initial energy	1 joule
E_{elect}	50 nj/bit
E_{fs}	10 pj/bit/signal

4.1.2 Network Model

Base station: It is static and located in coordinates (100,100). In simulations, we found that this place has more efficiency. The base station is entirely safe and has infinite energy.

Sensor nodes: All nodes in the simulation are homogeneous and are not better than another. Nodes are distributed with a uniform distribution in the environment. They are mobile and move with a speed of 10 m/s by a random waypoint algorithm in all experiments.

Mobile agent: Only one type of agent is used in this method. The agents are randomly placed on nodes done by the base station at the beginning of the network creation. The agents perform agent cycling every 5 to 10 seconds.

Malicious node: Malicious nodes are the regular nodes in the network that generate sinkhole attacks. In each experiment, 10% to 20% of total nodes are malicious and distributed randomly around the network environment.

4.1.3 Experimental Result

Here, we compare the simulation results of our method with Hamedheidari and Rafeh's method [9] and Jatti and Sonti's method [11] in terms of packet loss, energy consumption, throughput and agent overhead. As shown in the experiments, our proposed method generates better results in terms of packet loss ratio and the agents' overhead. It also leads to acceptable energy consumption and throughput.

4.1.3.1 Energy Consumption

Since energy is the most vital resource for sensor nodes, the methods and approaches proposed for sensor nodes need to be economical in terms of energy consumption. Figures 3-6 show the energy consumption of our proposed method in comparison with the previous methods. As shown in these figures, increasing the number of agents increases the consumed energy. However, the amount of increase is less with 100 nodes than with 400 nodes in the network. It is because of more scattering between nodes in the large-scale networks with a few nodes. As a result, their neighbors are less and agents visit fewer nodes in every cycling. The energy consumption of the entire network is still low. But in dense networks; i.e., networks with a large number of nodes (because of having more neighbors), agent cycling is performed more, so more energy is consumed. The Hamedheidari and Jatti methods consume less energy than the proposed method because regular nodes know their trusted neighbors through trust packets that are sent by the agent node. Still, in the proposed method, a regular node calculates the trust value of its neighbors. So, in the proposed method, each node consumes more energy.

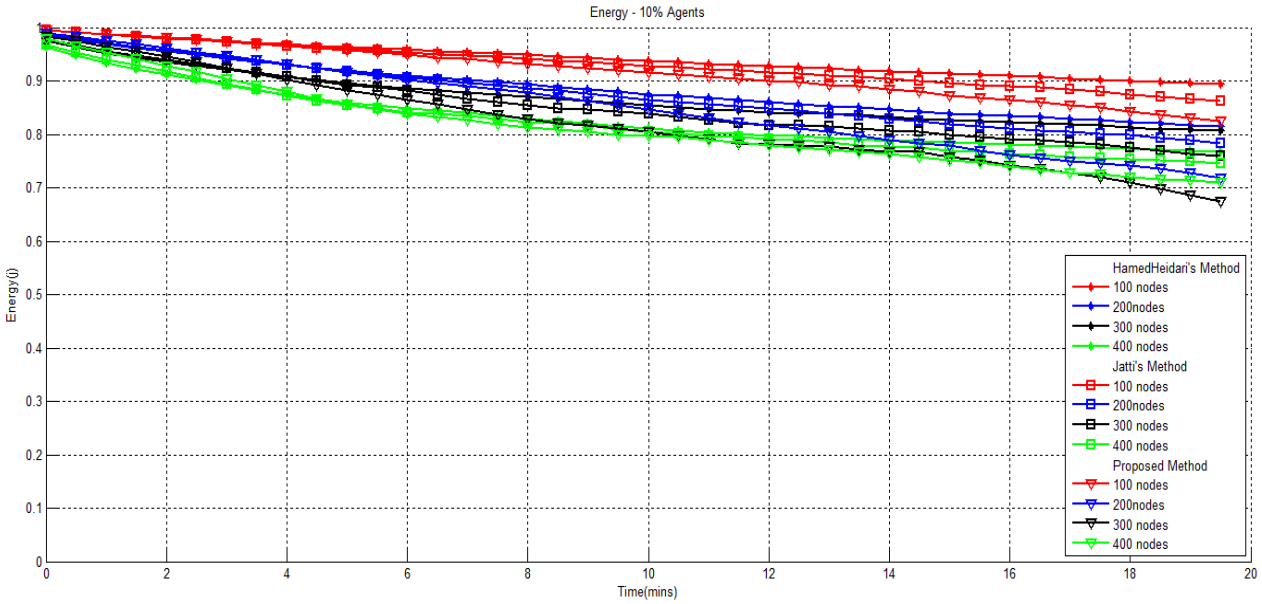

Figure 3. The energy consumption in the compared methods with 10 percent agent.

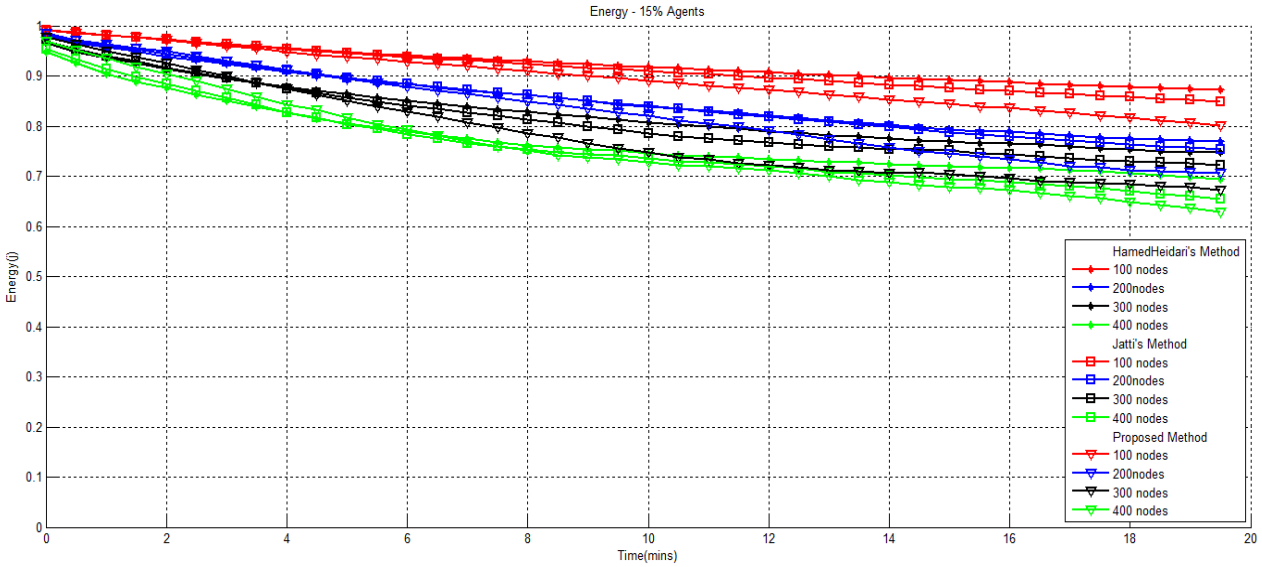

Figure 4. The energy consumption in the compared methods with 15 percent agent.

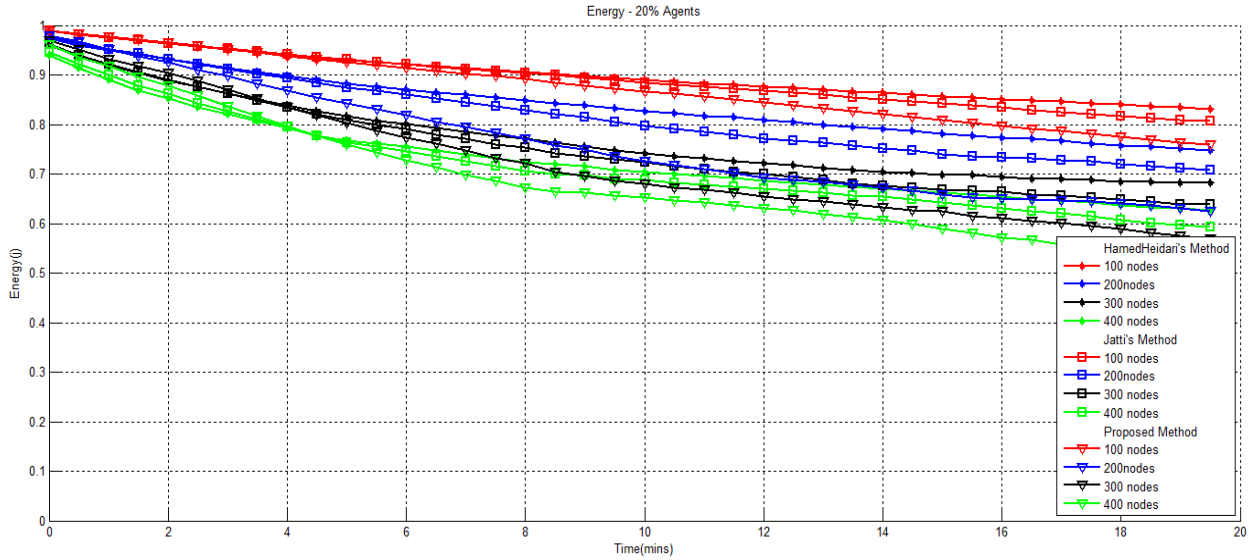

Figure 5. The energy consumption in the compared methods with 20 percent agent.

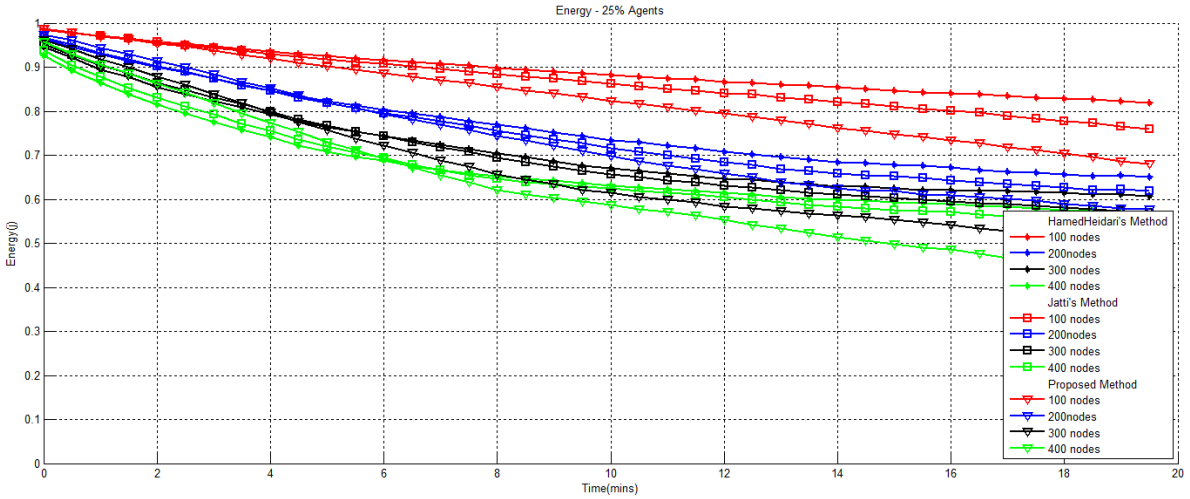

Figure 6. The energy consumption in the compared methods with 25 percent agent.

4.1.3.2 Packet Loss Rate

The most possible related problem in sinkhole attacks is packet loss. The attacker, after receiving the packets, does not transmit them. Packet loss is a vital problem in many applications. We compare the packet loss rate of our proposed method with the previous methods in Figures 7-10. The packet loss in the Hamedheidari and Jatti methods is caused by the presence of uncovered nodes for the agent. The uncovered nodes assume that all their neighbors are attackers and do not interact with them.

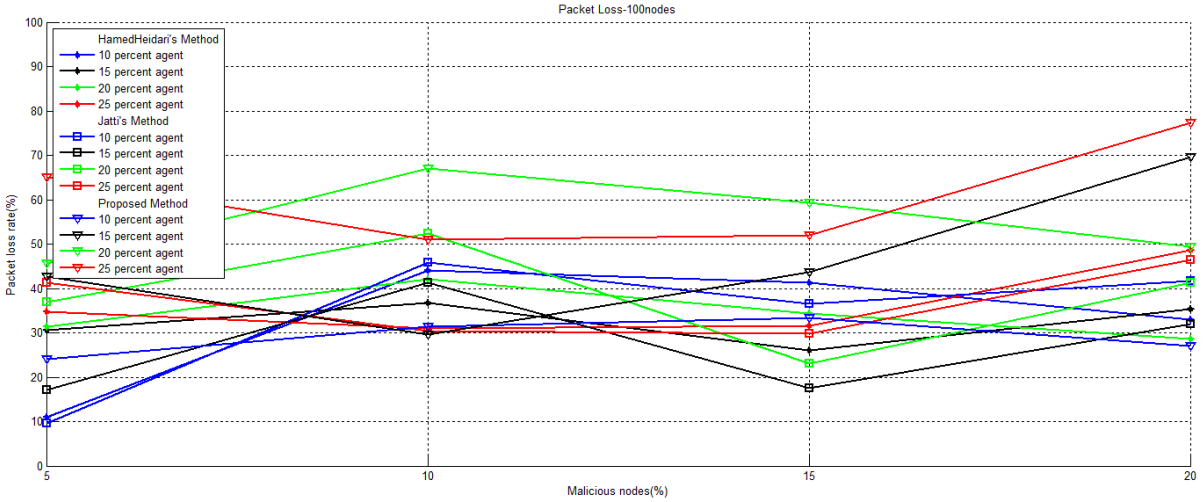

Figure 7. Comparison of the packet loss rate in the compared methods with 100 nodes in the network.

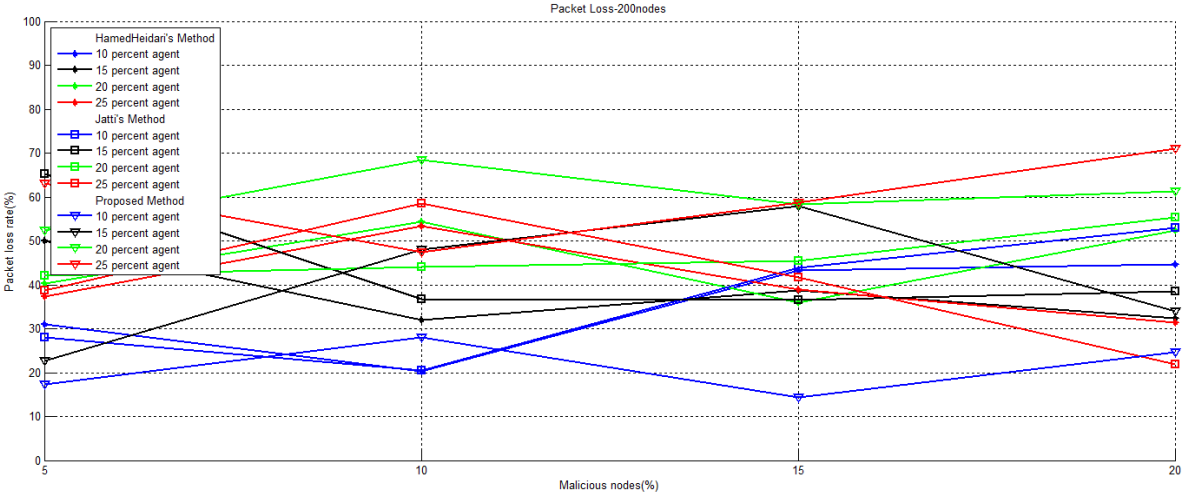

Figure 8. Comparison of the packet loss rate in the compared methods with 200 nodes in the network.

Therefore, by increasing the number of agents, the network is covered better and the packet loss rate gets reduced in the Hamedheidari and Jatti methods. Since the uncovered nodes can recognize their malicious neighbors, so in a network with a large number of nodes and a low percentage of agent nodes, the number of lost packets in the proposed method is less than in the compared methods, as shown in Figure 10. Another reason for packet loss is the end of energy of intermediate nodes in a data path. The energy consumption in the proposed method is more than the Hamedheidari and Jatti methods, so from this point of view, the number of packet losses in the compared methods is less than in the proposed method.

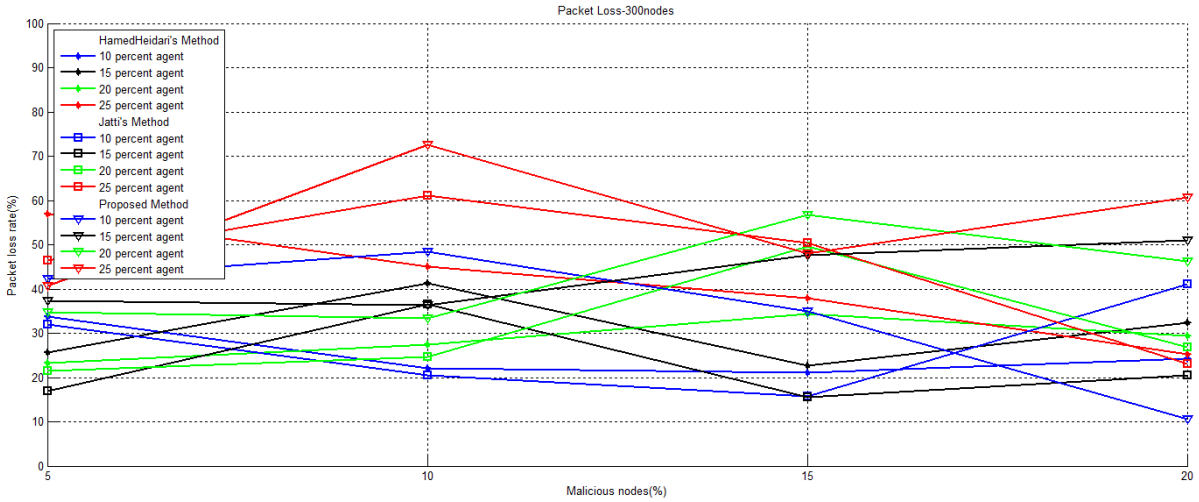

Figure 9. Comparison of the packet loss rate in the compared methods with 300 nodes in the network.

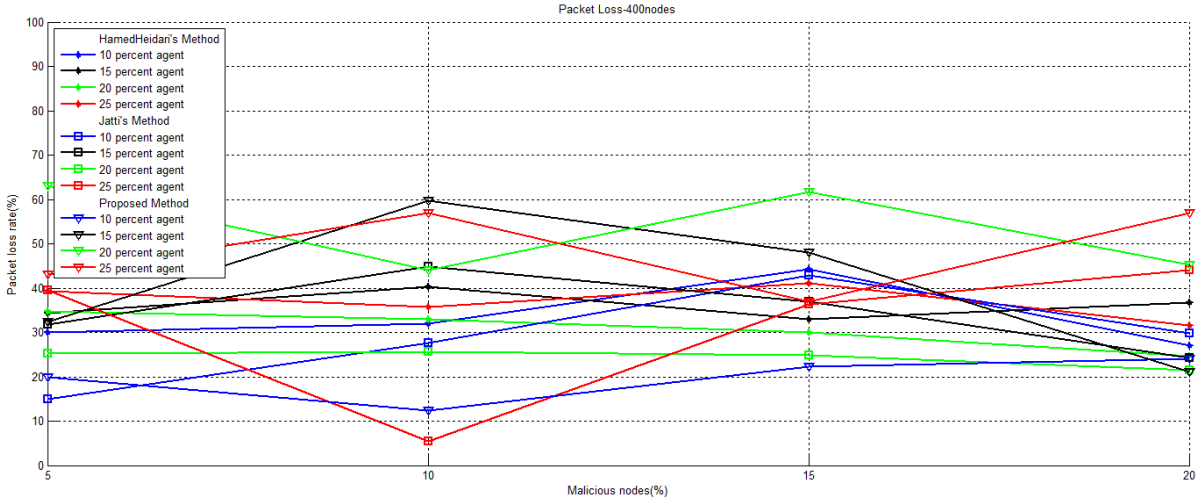

Figure 10. Comparison of the packet loss rate in the compared methods with 400 nodes in the network.

4.1.3.3 Throughput

Throughput is the average of successful message delivery in a communication channel. Since the sinkhole attack forwards the packets in the wrong paths or does not transmit them, throughput is reduced. We can measure the throughput by data packets per second or data packets per interval. Comparison of throughput with the two methods is depicted in Figures 11-14. In the Hamedheidari and Jatti methods, only the agents are responsible for detecting malicious nodes, whereas in our method, this process is done by agents and nodes. As shown, by increasing the number of agents, the compared methods have better performance than our method.

4.1.3.4 Mobile Agents' Overhead

The next criterion that we review in simulations is the average mobile agents' overhead in the network. A comparison of the average of mobile agents' overhead between the proposed method and the compared methods is shown in Figures 15-18. This criterion has been calculated by the rate of the number of

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 04, December 2021.

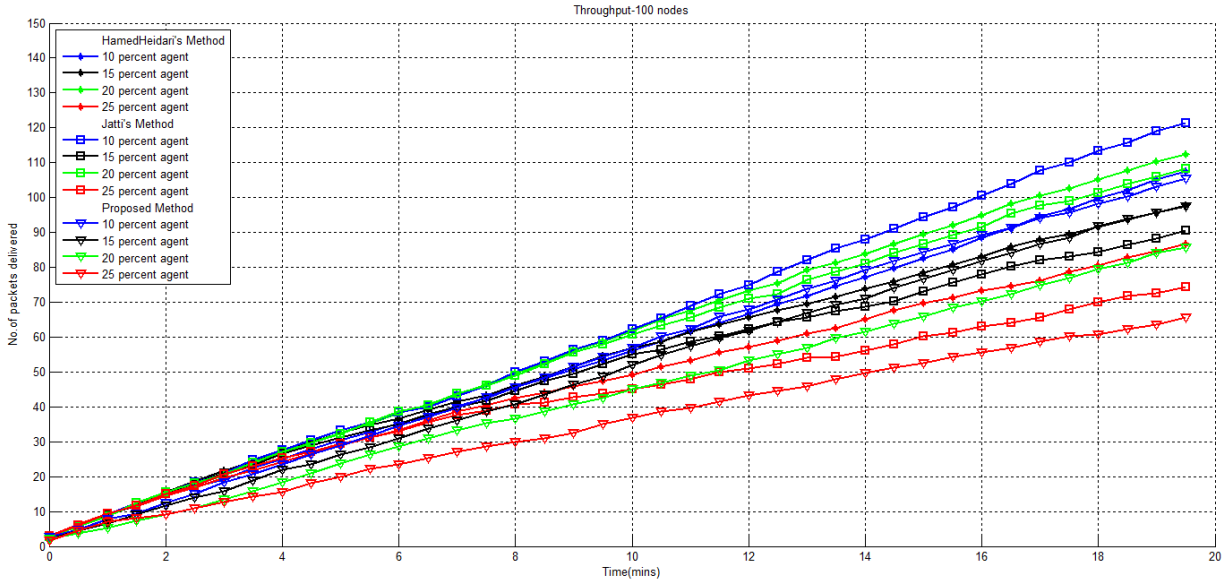

Figure 11. Comparison of throughput in the compared methods with 100 nodes in the network.

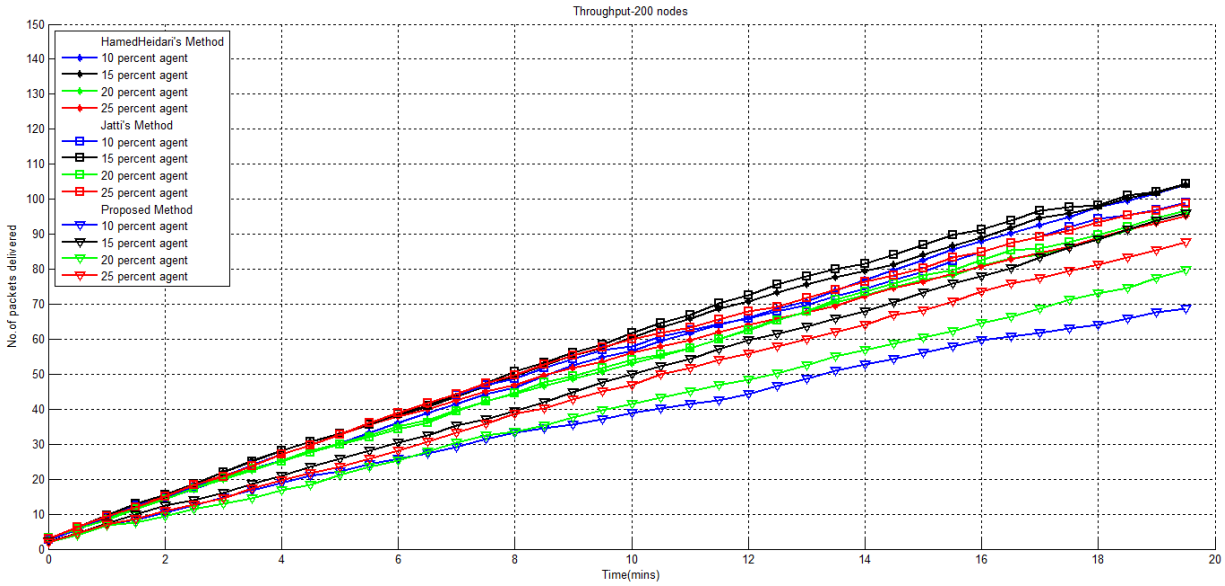

Figure 12. Comparison of throughput in the compared methods with 200 nodes in the network.

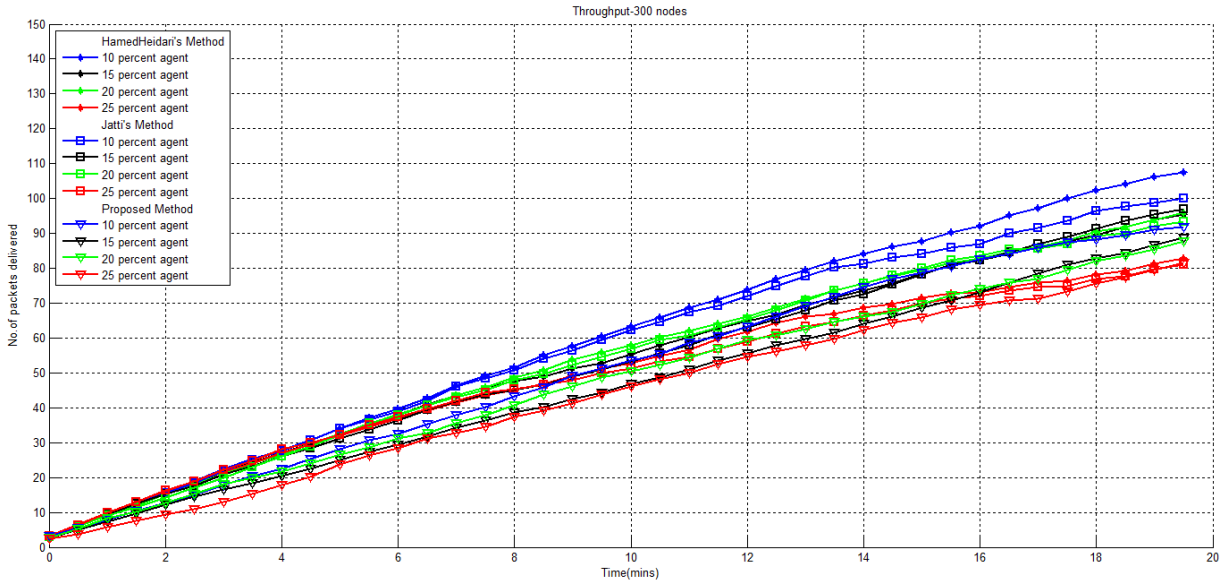

Figure 13. Comparison of throughput in the compared methods with 300 nodes in the network.

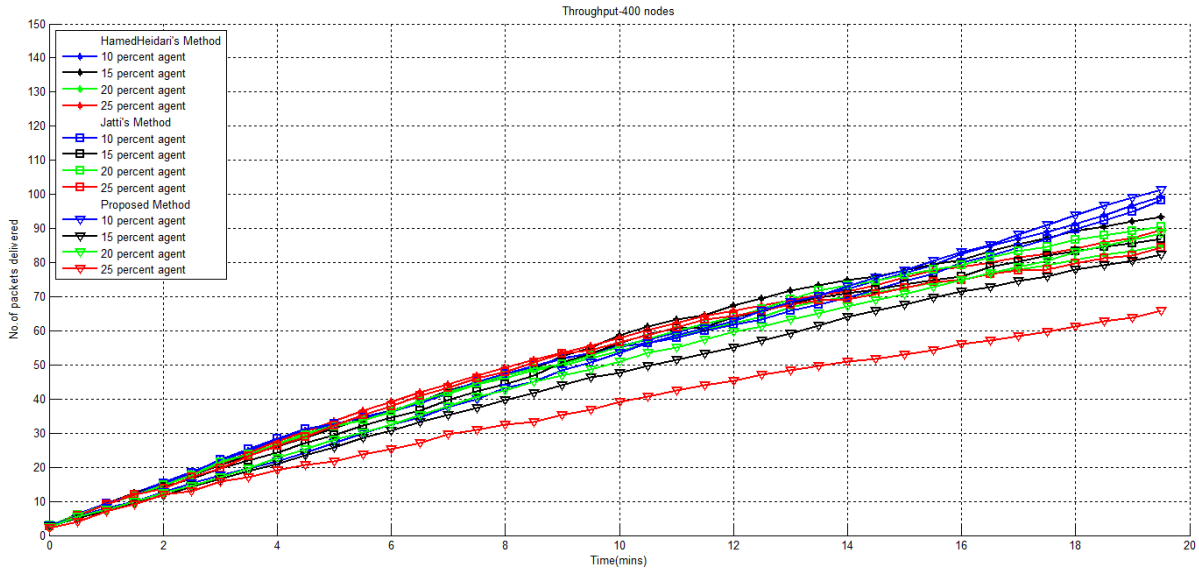

Figure 14. Comparison of throughput in the compared methods with 400 nodes in the network.

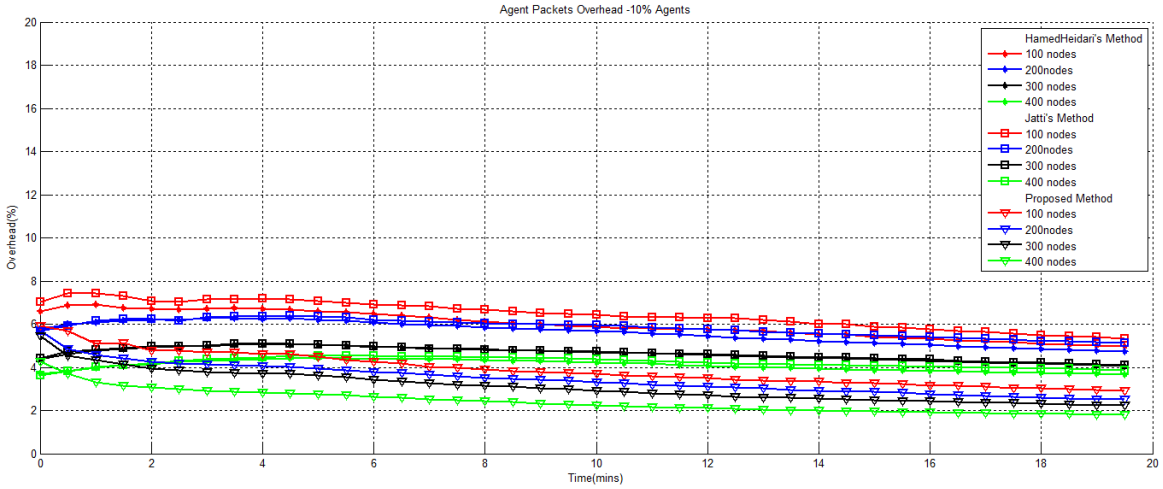

Figure 15. The mobile agents' overhead in the compared methods with 10 percent agent.

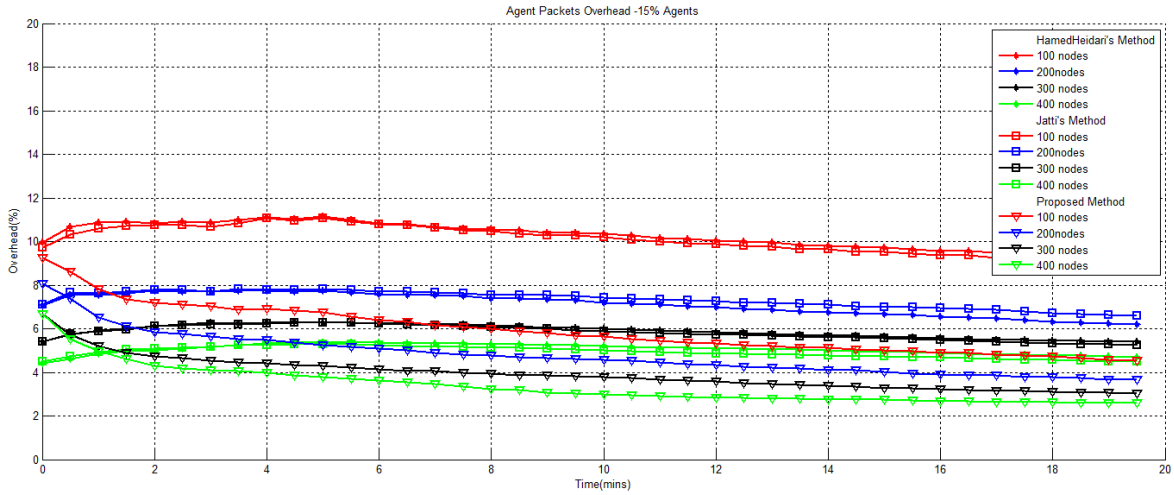

Figure 16. The mobile agents' overhead in the compared methods with 15 percent agent.

control packets). As shown, the agent overhead increases while the number of agent migrations becomes more. In other words, agents' overhead is reduced by increasing the number of sensor nodes. Moreover, there is no trust packet in our method; so, the agent's overhead is reduced more than the Hamedheidari and Jatti methods.

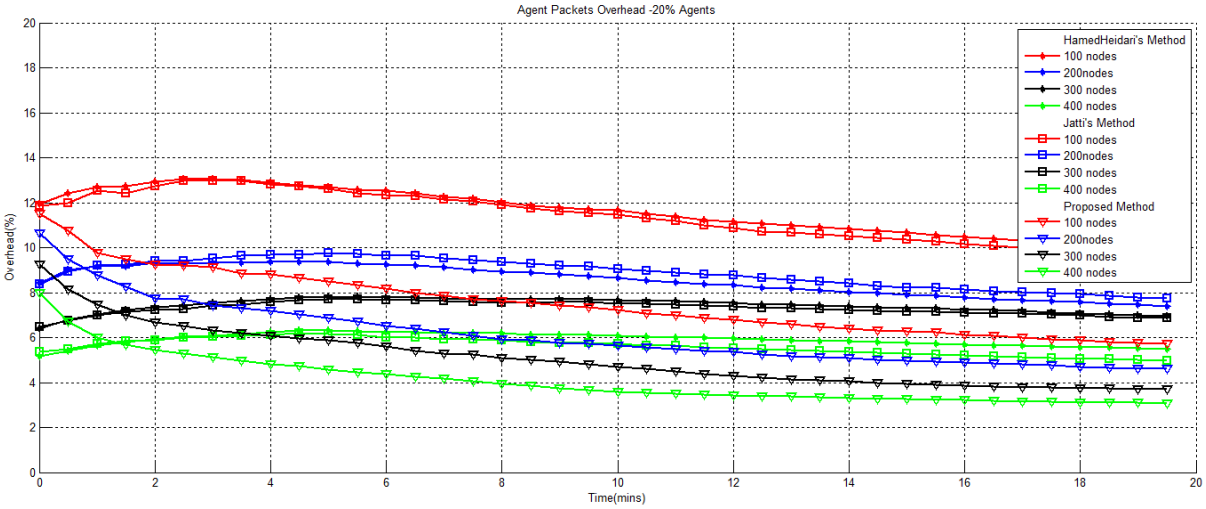

Figure 17. The mobile agents' overhead in the compared methods with 20 percent agent

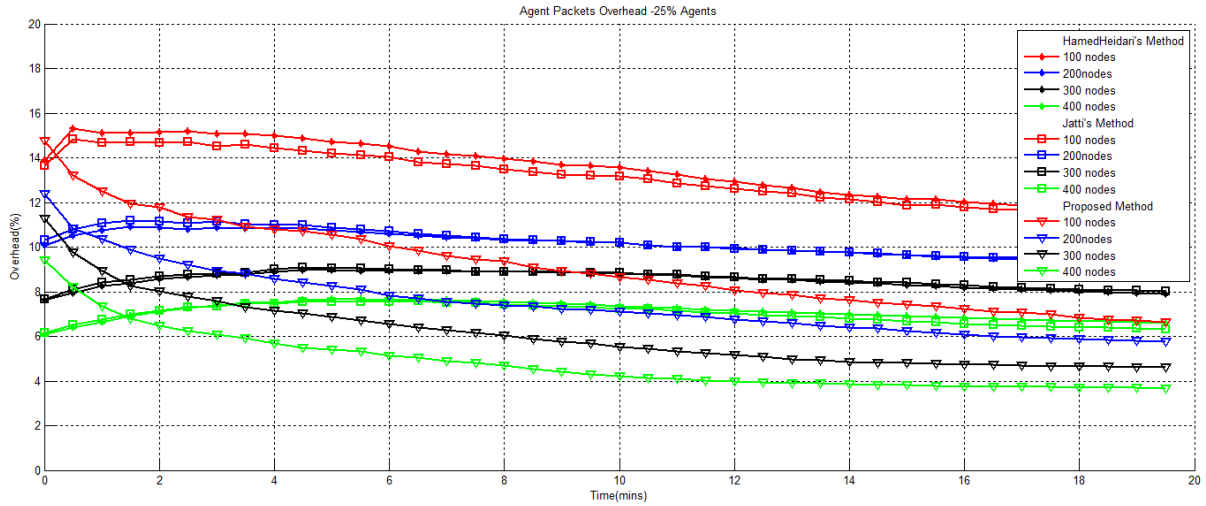

Figure 18. The mobile agents' overhead in the compared methods with 25 percent agent.

5. Conclusion and Future Work

In this research, we proposed a novel mobile agent-based technique to counter sinkhole attacks in WSNs. We carefully examined the most relevant method to the subject of this paper and issues which could challenge its security. We then presented our solution, which covered the security flaws of previous methods. The simulation results showed that our method could improve the overhead caused by the agents in the network and the packet loss ratio in comparison with the previous methods. At the same time, other criteria, such as energy consumption and throughput remained almost the same. Furthermore, our method resolved the issue of uncovered nodes in the previous methods by equipping each node to have the ability to detect adversaries on its own. In the future, we plan to apply fuzzy logic to improve the detection algorithm of the malicious nodes. Moreover, we want to extend our method to support the other routing protocols.

References

[1] M. Ali, M. Nadeem, A. Siddique, S. Ahmad and A. Ijaz, "Addressing Sinkhole Attacks in Wireless Sensor Networks: A Review," Int. Journal of Scientific & Technology Research, vol. 9, no. 8, pp. 406-411, 2020.

[2] R. Almesaeed, A. Al-Nasser and H. Al-Junaid, "A Comprehensive Survey on Routing and Security in Mobile Wireless Sensor Networks," International Journal of Electronics and Telecommunications, vol. 67, no. 3, pp. 483-496, 2021.

[3] I. Almomani and K. Sundus, "The Impact of Mobility Models on the Performance of Authentication Services in Wireless Sensor Networks," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 1, pp. 75-93, 2020.

[4] S. Aryai and G. S. Binu, "Cross Layer Approach for Detection and Prevention of Sinkhole Attack Using a Mobile Agent," Proc. of the 2nd IEEE International Conference on Communication and Electronics Systems (ICCES), pp. 359-365, DOI: 10.1109/CESYS.2017.8321299, Coimbatore, India, Oct. 2017.

[5] M. Bahekmat, M. H. Yaghmaee, A. S. H. Yazdi and S. Sadeghi, "A Novel Algorithm for Detecting Sinkhole Attacks in WSNs," International Journal of Computer Theory and Engineering, vol. 4, no. 3, pp. 418-421, 2012.

[6] J. A. Chaudhry, U. Tariq, M. A. Amin and R. G. Rittenhouse, "Dealing with Sinkhole Attacks in Wireless Sensor Networks," Advanced Science and Technology Letters, vol. 29 (SecTech 2013), pp. 7-12, 2013.

[7] H. M. A. Fahmy, Wireless Sensor Networks: Concepts, Applications, Experimentation and Analysis, 1st Ed., ISBN-13: 978-9811004117, Cairo, Springer, 2016.

[8] G. P. Gupta, M. Misra and K. Garg, "Energy and Trust Aware Mobile Agent Migration Protocol for Data Aggregation in Wireless Sensor Networks," Journal of Network and Computer Applications, vol. 41, pp. 300-311, DOI: 10.1016/j.jnca.2014.01.003, 2014.

[9] S. Hamedheidari and R. Rafeh, "A Novel Agent-based Approach to Detect Sinkhole Attacks in Wireless Sensor Networks," Computers & Security, vol. 37, pp. 1-14, DOI: 10.1016/j.cose.2013.04.002, 2013.

[10] G. Jahandoust and F. Ghassemi, "An Adaptive Sinkhole Aware Algorithm in Wireless Sensor Networks," *Ad Hoc* Networks, vol. 59, no. C, pp. 24-34, DOI: 10.1016/j.adhoc.2017.01.002, 2017.

[11] A. V. Jatti and V. K. Sonti, "Sinkhole Attack Detection and Prevention Using Agent Based Algorithm," Journal of University of Shanghai for Science and Technology, vol. 23, no. 5, pp. 526-544, 2021.

[12] G. Kalnoor, J. Agarkhed and S. R. Patil, "Agent-based QoS Routing for Intrusion Detection of Sinkhole Attack in Clustered Wireless Sensor Networks," Proc. of the 1st International Conference on Computational Intelligence and Informatics, pp. 571-583, Springer-Singapore, 2017.

[13] S. Kaur and N. Goyal, "A Survey on Security Attacks in Wireless Sensor Networks," International Journal of Advanced Research in Computer Science, vol. 7, no. 6, pp. 94-96, 2016.

[14] G. Kim, Y. Han and S. Kim, "A Cooperative-sinkhole Detection Method for Mobile *Ad Hoc* Networks," AEU-International Journal of Electronics and Communications, vol. 64, no. 5, pp. 390-397, 2010.

[15] I. Krontiris, T. Giannetsos and T. Dimitriou, "Launching a Sinkhole Attack in Wireless Sensor Networks; the Intruder Side," Proc. of the International Conference on Wireless and Mobile Computing, pp. 526-531, DOI: 10.1109/WiMob.2008.83, Avignon, France, 2008.

[16] L. Mechtri, F. T. Djemili and S. Ghanemi, "Agent-based Intrusion Detection in Wireless Networks," Implementing Computational Intelligence Techniques for Security Systems Design, pp. 97-130, DOI: 10.4018/978-1-7998-2418-3.ch005, IGI Global, 2020.

[17] O. Naderi, M. Shahedi and S. M. Mazinani, "A Trust Based Routing Protocol for Mitigation of Sinkhole Attacks in Wireless Sensor Networks," International Journal of Information and Education Technology, vol. 5, no. 7, pp. 520-526, 2015.

[18] K. E. Nwankwo and S. M. Abdulhamid "Sinkhole Attack Detection in A Wireless Sensor Networks Using Enhanced Ant Colony Optimization to Improve Detection Rate," Proc. of the 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf), pp. 1-6, Zaria, Nigeria, Oct. 2019.

[19] H. Salameh, M. Dhainat and E. Benkhelifa, "A Survey on Wireless Sensor Network-based IoT Designs for Gas Leakage Detection and Fire-fighting Applications," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 2, pp. 60-72, 2019.

[20] R. A. Shaikh, H. Jameel, B. J. d'Auriol, H. Lee, S. Lee and Y. J. Song, "Group-based Trust Management Scheme for Clustered Wireless Sensor Networks," IEEE Transactions on Parallel and Distributed Systems, vol. 20, no. 11, pp. 1698-1712, 2009.

[21] S. Sharmila and G. Umamaheswari, "Detection of Sinkhole Attack in Wireless Sensor Networks Using Message Digest Algorithms," Proc. of the IEEE International Conference on Process Automation, Control and Computing (PACC), pp. 1-6, Coimbatore, India, DOI: 10.1109/PACC.2011.5978973, 2011.

[22] D. Sheela, K. C. Naveen and G. Mahadevan, "A Non Cryptographic Method of Sink Hole Attack Detection in Wireless Sensor Networks," Proc. of the IEEE Int. Conf. on Recent Trends in Information Technology (ICRTIT), pp. 527-532, DOI: 10.1109/ICRTIT.2011.5972397, Chennai, India, 2011.

[23] G. Thirumalaimuthu, E. E. Lawrence and S. Meenakshi, "Security in Wireless Sensor Networks: Issues and Challenges," International Journal of Computer Application, vol. 6, no. 2, pp. 145-151, 2016.

[24] H. Wang, "A Three-tier Scheme for Sybil Attack Detection in Heterogeneous IWSN," Proc. of the Int.

Conf. on Computer Science Communication and Network Security (CSCNS2019), MATEC Web of Conferences, vol. 309, p. 02005, pp. 1-8, EDP Sciences, 2020.

[25] S.-H. Yang, Wireless Sensor Networks - Principles, Design and Applications, ISBN-13: 978-1447169321, London, Springer, 2014.

ملخص البحث:

شبكات المجسّات اللاسلكية تشكّل تقْنيةً مطبّقةً على نطاقٍ واسع في مجالاتٍ متعدّدة. وفيما يتعلّق بمحدِّدات شبكات المجسّات اللاسلكية، فهي تواجه العديد من الهجمات. وتعدّ هجمات "البالوعة" أكثر هذه الهجمات شيوعاً وأشدّها خطراً في تسيير شبكات المجسّات اللاسلكية. وهناك الكثير من الطّرق لمواجهة هذا النّوع من الهجمات في الأدبيّات المتعلّقة بالموضوع. وتعطي الطّرق المعتمدة على العوامل الديناميّة نتائج أفضل في مواجهة هذا النّوع من الهجمات والتغلّب على محدِّدات شبكات المجسّات اللاسلكية.

في هذه الورقة، نقدّم طريقةً تستند على العوامل الديناميّة وتستخدم قيمة الثّقة لكلّ مجسّ للكشْف عن هجمات "البالوعة" ومنْعها. نقوم بحساب قيم الثّقة لإعلام عُقَد المجسّات عن مكاناتِ جاراتها.

وكما تبيّن التجارب، فإنّ الطّريقة المقترحة في هذه الورقة أسفرت عن نتائج أفضَل من حيث معدّل فَقْد الحُزَم. كذلك فهي تُصْلِح نواقص الأعمال السّابقة بخصوص أمْن الشّبكة، وتقلّل تكلفة العوامل في الشّبكة مقارنةً بالطّرق السّابقة.