sciendo RIGA TECHNICAL UNIVERSITY *Applied Computer Systems*

ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
December 2018, vol. 23, no. 2, pp. 118–127
doi: 10.2478/acss-2018-0015
https://content.sciendo.com

A Credit-based Method to Selfish Node Detection in Mobile Ad-hoc Network

Sanaz Nobahary[1], Shahram Babaie[2*]
[1,2]*Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran*

***Abstract* – Ad-hoc networks are a set of mobile nodes that are connected via a wireless channel. Some of the nodes in this network behave selfishly and do not send data to other nodes so that in order to increase network performance these nodes must be identified. A credit-based algorithm is proposed to detect the selfish nodes. Three watchdog nodes are selected to monitor suspicious nodes in each cluster. The cluster head nodes detect the existence of selfish nodes by controlling general features of network, such as delay, the total number of sent packets, the total number of received packets, throughput, and network traffic. The watchdog nodes send their comment on selfishness or cooperation of the node to the cluster head. Cluster head makes decisions with a majority vote on a suspicious node. The simulation results show that the rate of detection accuracy and the life time of network are considerably high and the false alarm rate and energy consumption are low comparing to that of similar methods.**

***Keywords* – Credit, detection accuracy (DA), false alarm rate (FAR), selfish nodes, watchdog nodes.**

I. Introduction

Ad-hoc networks are a set of mobile nodes that are connected via a wireless channel [1]. Each node can move independently in any direction, and therefore will frequently change its links to other devices [2]. In a mobile ad-hoc network, data transition between nodes is done through other nodes, so the nodes work as a router in the network [3]. These types of networks are temporarily formed and may operate independently or connect to another network, such as the Internet [4].

Due to lack of energy, central controller and multi-step to forward the data packets in the mobile ad-hoc networks, a selfish node attack occurs in the network [5]. In this attack, the nodes tend to get the most benefit from the network while at the same time trying to maintain their resources, such as bandwidth, energy supply [6]. A selfish node is only associated with other nodes to send its data packets but refuses to forward the data packets of other nodes and cooperate with the other nodes [7]. Whenever it receives the data packets, it does not show any interest in forwarding them, so the data packets are discarded or retransmitted [8]. The selfish nodes should detect and isolate in the network to prolong the network lifetime and improve the network performance [9].

All the nodes in mobile ad-hoc networks are divided into three groups in cooperation: normal, selfish, and malicious nodes. The normal nodes conduct their normal activities in the network and send all the data packets [10]. The malicious nodes are for sabotage purposes, i.e., the data packets are either eliminated or ignored, routing the packets in the wrong direction, hiding their identity, wasting energy of other nodes, holding the bandwidth, interfering with network performance [11]. One of the main challenges in mobile ad-hoc networks is non-cooperation of some nodes and selfish behaviour. The selfish nodes use the network resources to personal tasks, do not participate in sending and receiving data, just send their own data packets [12]. The selfish nodes decrease the network performance and increase the delay of data packets [13].

In this paper, which is devoted to a credit-based approach, network nodes are clustered to better monitor nodes. To select the cluster head, three parameters – energy, the number of neighbours, and the location of the node – are important. The cluster head detects the existence of selfish nodes by controlling the general characteristics of a network, such as average end-to-end delay, the total number of sent packets, throughput, the total number of received packets, and network traffic. The credit of each node is stored in a table. A low credit is suspected of selfishness. Of the neighbours of the suspected node, three nodes with high credit are selected as watchdog nodes. The watchdog nodes monitor node performance. The watchdog nodes send their comment on selfishness or cooperation of the node to the cluster head. Cluster head, with a majority vote, identifies the selfish node.

Section I provides an insight to the selfish nodes. Section II, describes different strategies to detect the selfish nodes, as well as the algorithms to identify the selfish nodes. In Section III, the proposed algorithm to discover the selfish node is described. In Section IV, the proposed algorithm is simulated and evaluated with the other algorithms.

II. Related Work

Different strategies have been proposed to detect the selfish nodes: credit-based mechanism, reputation-based mechanism, game theory mechanism, punishment-based mechanism, hybrid and specification mechanism, acknowledgment mechanism [14]. The credit-based mechanism allocates credits to each node. The methods use currency to pay the nodes to forward the packets and nodes can gain more currency by cooperating with other nodes and it stimulates the nodes to cooperate and forward the node packets. The nodes in the high credit show the reliable nodes and the low credit determines unreliability of the node [15], [16]. The reputation-based mechanism uses the nodes as watchdogs to monitor the node actions to forward data packets of the other nodes. The network nodes work together to provide

* Corresponding author's e-mail: hwww.tab@gmail.com

feedback to specific nodes. Each node receives a reputation value with feedback and stores them in the table. Thus, the nodes with high reputation cooperate with other nodes [17]. The third group of selfish node detection is punishment-based method. In this strategy, the nodes that cooperate with the other nodes in the network are rewarded but the nodes that do not cooperate with other nodes to forward the data packets are punished. Punishing the nodes stimulates them to cooperate with the other nodes in the network [18]. The other group of the detection method is an acknowledgment-based mechanism that is used to acknowledge a message (Ack) in order to control the network nodes. In an acknowledgment-based mechanism, the destination nodes send an Ack packet to report the source node as a receipt of the data packets. The most disadvantage of the strategy is the high traffic load in the network [19]. Game theory based mechanisms use a game theory to detect the misbehaviour nodes [20]. The nodes in the network act as players and forwarding the data packets is an action in the game by evaluating the utility in the game, which can detect the misbehaviour nodes [21], [22]. Hybrid and specification mechanism are the last group, which uses the combination of the other nodes. This strategy commonly uses reputation-based mechanism, credit-based mechanism, and game theory based mechanism, so this mechanism takes advantage of the other methods [23].

As mentioned before, some of the proposed methods in the last group model the node behaviour to detect the misbehaviour nodes. Sengathir et al. proposed a futuristic trust coefficient-based semi-Markov prediction (FTCSMP) mechanism to detect and predict the node behaviour. This algorithm detects and isolates the selfish nodes based on semi-Markov prediction. The data packet is sent through the path, which has a higher trust ratio. The coefficients are assigned to each node, which determines the likelihood ratio of becoming the selfish node. This algorithm detects the selfish nodes by forecasting the number of the forwarding packets, the number of received packets to the destination, and the remaining energy. FTCSMP has low energy consumption and high false positive rate [24]. A probabilistic behavioural model (PBM) is the method, which models the node behaviour. The algorithm models the behavioural probability based on the node energy and the network global properties. The method provides a balance between energy and the network properties. Probability behaviour of the node, which has a high dependence on the packet forwarding and the rate of discarded packet is predicted. To predict the node behaviour, a dynamic table is used, which is called a neighbourhood table. The table regularly estimates the rate of neighbouring nodes. In the table, the average rate of the sent packet and average end-to-end delay of the packet are recorded. By using the neighbourhood table, selfish nodes are detected and then isolated. PBM energy consumption and false positive rate are high [25]. Azni proposed a correlated node behavior model (CNBM) to detect the selfish nodes in the networks. The selfish nodes are detected by using the table and then isolated in the network, and the other node prevents to cooperate with the selfish nodes. The model specifies the node behaviour based on the likelihood of selfishness, the probability of sent packet, the likelihood of the packet dropped. CNBM has low energy consumption and false positive rate [26]. Epidemic modelling for correlated node behaviour model (ECNBM) is proposed by Azin et al. to identify the node behaviour. The proposed method monitors the node behaviour in the routing path, so a suspicious node is detected. The suspicious nodes are isolated in the network in two steps. In the first step, node properties are checked. In the second step, based on the predicting according to the behaviour of the current state, the selfish nodes are isolated. ECNBM has low energy consumption and a high false positive rate [27].

Wahab has introduced a cooperation watchdog model based on Dempster-Shafer QOS-OLSR to detect the selfish nodes. The method uses clustering to achieve better monitoring. A node, which has a higher reputation, is selected as a cluster head in a short time. The data sent by the source node are stored in a table. Each node sends data; a watchdog node compares data with the data in the table. If they are not the same, a node is known as a suspicious node. By changing some watchdogs, data aggregation is done. Dempster-Shafer theory is used for data aggregation. QOS-OLSR energy consumption and false positive rate are low [28]. In 2015, Jesudoss designed a method based on a reward and punishment mechanism for collaboration between nodes to stimulate the nodes to cooperate with the protocol called a payment punishment scheme (PPS). Every node has a credit. When a node cooperates with other nodes, its credit increases. If a node does not cooperate with other nodes, its credit decreases. Node packets with high credit are sent earlier than other packets. For all nodes, there are incentives to cooperate. Clustering is performed based on a specific algorithm. In every node, a cluster head and an assistant cluster head are selected based on three parameters: energy remaining, the number of neighbours and the distance. Three watchdogs, including previous forwarding node, assistant cluster head, one of the neighbouring nodes, are selected by a round robin. Three watchdogs send their opinions about cooperation or non-cooperation of the node. The node credits are kept in the credit table. This method has high overhead energy consumption [29]. A green approach is designed in two phases. In this method, the discovery of selfish nodes is done in two phases: general and local phases. Base station discovers selfish node existence in the network. Base station uses the general properties of the network, such as delay, network load, throughput, forwarding packets, and received packets. In local phase, by controlling properties of average end-to-end delay and average network load, selfish nodes are detected. Green approach has high energy consumption and a low false positive rate [30]. Random two Ack is the proposed method in the acknowledgment-based category to detect the selfish nodes and malicious nodes. Hash chain is used to detect malicious nodes. The receipt of a packet is sent to the source node. In the acknowledgment message, the hash number is packed. If the hash number is equal, the node is not malicious. If the source node receives the acknowledgment message, neighbour node is not selfish. Due to high overhead, the request for an acknowledgment message is randomly placed in the packet [31]. Abdelkader proposed a method to handle node selfishness by detecting and motivating nodes to cooperate

in data forwarding. This method has two phases; the nodes which are refusing to forward data are identified in the first phase. In the second phase, the selfish nodes are stimulated to forward data packets. In this method, a false positive rate and energy consumption are high [32]. Fuzzy-based scheme is proposed to detect selfish nodes; each node monitors its one-hop neighbour actions, then computes the trust of them. The trust values are sent to a fuzzy function, which is mapped into different classes. Each class shows the trust value of each node. If the node trust is low, data packets are not forwarded by the node. The false positive rate of this method is low, but energy consumption is high [33]. Kerrache et al. proposed the strategy that used direct and indirect trust to detect misbehaviour node while the nodes could define the direct trust by interactions between neighbour nodes. The nodes define indirect trust, by the evaluation of direct interactions. This method has a low false positive rate and high energy consumption [34]. Clustered-based method is proposed to detect the misbehaviour nodes by watchdog nodes. Each cluster has a cluster head to help nodes to transmit and receive packets. To identify malicious nodes, a trust management protocol, QoS trust with some social trust, is used. In this method, a false positive rate and energy consumption are high [35]. Lupia et al. proposed a method called TEEM that used a time division-based monitoring method to get high security levels. In this method, a monitoring period is divided, so the energy consumption is low. The network nodes are commonly monitored from the beginning. This method has a low false positive rate [36].

All aforementioned schemes and algorithms are important and cannot be ignored; however, each of them has weaknesses in some circumstances that must be improved. To provide an efficient algorithm to detect selfish nodes in MANET, the strengths of them can be beneficial. We present a new approach to detect a selfish node in MANET in this paper.

III. Proposed Method

The selfish nodes are nodes that send their own data but refuse to send the data from other nodes. The existence of such nodes will paralyze the network, disturb the normal process of the network, and reduce the network performance. In order to solve this problem, it is attempted to give this node some motivation to encourage it for cooperation and to reduce the number of selfish nodes. The proposed algorithm is composed of three phases: setting up and clustering phase, general phase, and local phase. The existence of the selfish nodes is determined in the general phase and then they are detected in the local phase. These phases are described in more detail.

A. Setting up and Clustering Phase

In this phase, the mobile nodes that are able to move in every direction are distributed in the environment. The remaining energy and the credit are assumed to be the same. Each node stores some data about its neighbour nodes, which include the items that are shown in Fig. 1.

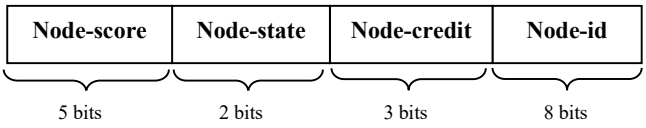

Fig. 1. Information on neighbour nodes.

Node identifier: this is the 8-bit field used to store the identifier of the neighbouring nodes.

Node-credit: each time it sends data to the neighbour node, it listens to the channel. If it makes sure that the neighbour node has sent the data, the credit will increase. This field is 3-bit.

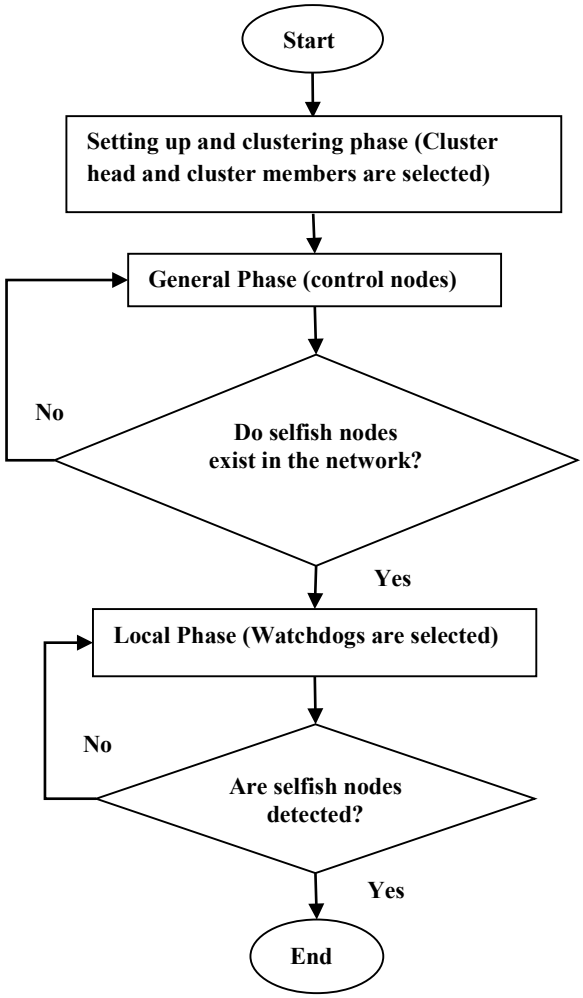

Fig. 2. Flowchart of the proposed method.

Node-state: this field is 2-bit and if a neighbour node is identified as a selfish node by the cluster head then it will be announced to every other neighbour node. The selfish node field is filled as ‘s’ state and no data will be sent to this node.

Node-score: this is a 5-bit field and according to (1) will fill the field per neighbour it has so to select the highest number as the cluster head among those.

In the proposed method after identifying and deploying network nodes for clustering, a clustering method based on distributed scores is used in the ad-hoc network [32]. The basics of the clustering in this method are composed of three parameters: the remaining energy, the number of neighbours, and credits. Each of the parameters is taken into account in the equation. Each node calculates its score using (1):

$$\text{score} = \left((Br \times C_1) + (N_\text{n} \times C_2) + (S \times C_3)\right) \quad (1)$$

$$C_1 + C_2 + C_3 = 1\,, \quad (2)$$

where C_1, C_2, and C_3 are the weight factors for system parameters, Br (remaining energy) – the energy of the nodes consumed through sending and receiving the data and messages, N_n (number of neighbours) – the number of neighbours in the radio range of the node, S (credits of node) – per sending and receiving moment, the credit data of the node will increase.

The main structure of the clustering is a three-step protocol: each node calculates its scores using (1) and broadcasts its score by an 'id. Score' message.

Updating the neighbouring tables: each node will update the neighbouring tables as soon as it receives a score message as shown in Fig. 1.

Each node selects a node with the highest score according to the neighbouring table as the cluster head and broadcasts it as a 'my-id, my-ch-id' message.

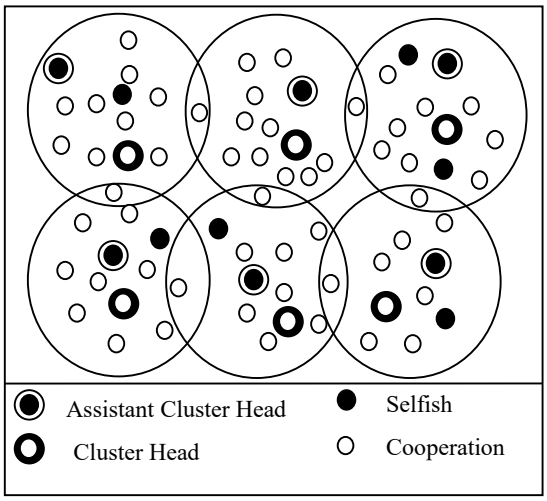

Fig. 3. Cluster and cluster member.

The nodes having the same cluster head field will be placed in one cluster as shown in Fig. 3. If this field is not the same, the majority vote is taken and after setting up the cluster the second highest score is selected as a cluster head assistant. The first phase is finished by selecting a cluster head and cluster head assistant. Table I shows notations.

TABLE I
NOTATIONS

Notation	Description
S	Selfish nodes
C	Cooperation nodes
LS	Suspected selfish nodes
CH	Cluster head
ACH	Assistant cluster head
θth	Threshold in the general phase
ηth	Threshold in the local phase
round	Counter of data collection
State	State of node announced by watchdogs
general count	Counter of collecting data in the general phase
local count	Counter of collecting data in the local phase
total state	Final state of node: S/C

B. General Phase

The suggested method is based on the analysis of the effect of the selfish nodes on the network. The detection of the selfish nodes in this algorithm is composed of two general and local phases. In this general phase, the existence of these nodes is identified. In the general phase, no selfish node is identified. Existence or absence of the selfish nodes in the clusters is determined. If there is a selfish node, then the local phase is called to identify the selfish nodes. The watchdog and cluster head nodes aim at detecting the selfish nodes. This phase causes the existence of selfish nodes in the network to be announced timely and prevents the destructive operation of these nodes in the network. Equation (3) shows the calculation method for GP:

$$\text{GP} = \alpha \times \frac{D_\text{normal}}{D_\text{present}} + \beta \times \frac{L_\text{normal}}{L_\text{present}} + \gamma \times \frac{R_\text{present}}{R_\text{normal}} + \\ + \theta \times \frac{S_\text{normal}}{S_\text{present}} + \varphi \times \frac{T_\text{present}}{T_\text{normal}} \quad (3)$$

$$\alpha + \beta + \gamma + \theta + \varphi = 1 \quad (4)$$

where D is the average delay of the packet (μs), L is the network traffic (byte/s), R is the number of received packets, S is the number of sent packets, T is the throughput in byte/s, and the subscript 'present' is measurement at the current time and the subscript 'normal' is for the normal conditions where there are no selfish nodes. Selfish nodes increase average delay, the number of sent packets and the network traffic, but decrease the number of received packets and average throughput.

The average delay, the number of sent packets and the network traffic have positive relation with GP but the number of received packets and the throughput have negative relation with GP. If the GP_present value deviates from the GP_normal by a predefined threshold, then this situation is marked as having a selfish node in the cluster.

The weight factor is given based on the importance of the node. Parameters with a higher importance are given a higher weight so the algorithm can discover itself faster. In the case of detecting the existence of selfish nodes on the network by the cluster, the local phase is executed and the selfish nodes are detected in the network and their destructive operation is prevented. The semi-code relating to the general phase is as follows:

```
Step 0: Clustered nodes in setup phase
Step 1: general_count=0
Step 2: Cluster head  monitors the network every t
time
START: Compute D,T,S,R,L
Compute GPpresent
      if |GP_present − GP_normal| < θ_th  then
          general_count=1
           for   i=1   to  round-1      do
             Compute D,T,S,R,L
                 Compute GPpresent
                  if |GP_present − Gp_normal| < θ_th  then
                      general_count=general_count+1
                  End if
           End for
               if   general_count=round then
                       go to local phase
               Else
                       go to START
               End if
       End if
```

In the general phase, the network is monitored to the extent that the identity of the selfish node is detected; therefore, at line 4 and 5 parameters such as the delay, throughput, the number of sent packets, the number of received packets, and network traffic will be calculated, and if there is a threshold between $GP_{present}$ and GP_{normal} the local phase will be run.

C. Local Phase

Network nodes periodically send credit to the cluster head for the neighbouring nodes stored in the tables. The cluster head updates the node credits. The new credits will also be sent to the assistant node of the cluster head when the cluster head is dead or its energy is finished or when it is out of the cluster the credits of the nodes will not be eliminated. If the existence of the selfish node is detected in the general phase, then the cluster head first controls the behaviour of the nodes with less credit and then will review all the nodes to detect the selfish nodes. Three watchdog nodes are used to monitor the behaviour of the nodes. To select the watchdog nodes when the considered node is close to the cluster head or the cluster head assistant the cluster head itself or the assistant will be act as the watchdog node. If the node is not close to the cluster head or cluster head assistant, then those three nodes with higher credit will be selected among the u and they will be asked to witness for their cooperation or selfishness behaviour.

$$\mathrm{LP} = Q - P \tag{5}$$

where Q shows the number of the received packets and P indicates the number of sent packets. If LP is larger than a predefined threshold, a suspicious node is marked as a selfish node.

Three watchdog nodes send their comment to the cluster head after monitoring their neighbour node behaviour. The cluster head with the majority of votes will decide on its selfishness or cooperation behaviour. If the node is identified as selfish, it will announce it to the other nodes. If the node has less than three neighbours, the number of neighbours will inevitably be fixed upon its opinion. In case of existence of a selfish node, the data sending will be stopped. If the cluster head node is selfish, the neighbours of the cluster node will reduce its credit and inform the other nodes of the cluster; the cluster head assistant will be selected as the new cluster head node and the cluster nodes will take votes regarding the selection of a new cluster head assistant. The semi-code relating to the local phase is as follows:

```
Each watchdog monitors node
local_count=0
state=C
if  Q - P > η_th then
          local_count=1
          state=LS
            for   i=1   to  n-1      do
                  Compute Q,P
                   if    Q - P > η_th then
                             local_count=
local_count +1
                    End if
                      if  local_count= =n  then
                            state=S
                       End if
             End for
    Cluster head checks results from watchdogs
                       if s-number>= =c-number then
                              total-state=S
                       Else
                              total-state=C
 End if
```

After identification of selfish node in the network by the cluster heads, the local phase will be conducted to identify the selfish node. Three watchdog nodes will be selected and they will monitor the function of the suspicious nodes. At line 24, the number of the sent packets and the number of received packets is calculated from the suspicious node. At line 29, LP must be less than the threshold η. The behaviour of the node will be controlled n times and in case of getting the same results in these n times, the selfish node will be detected by the watchdog node. This result will be announced to the cluster head and each of the watchdog nodes will announce their reports to the cluster head node. The cluster head node using majority vote at line 41 will decide on the selfishness or cooperation state of the node and will announce it to the cluster nodes.

If one of these happens in the network, the following cases are due:

New node: in case of emergence of a new node in the cluster, the cluster head will take a report from the previous cluster head node in relation to the new node and will record it in the credit tables. If the report shows selfish behaviour, the report will be

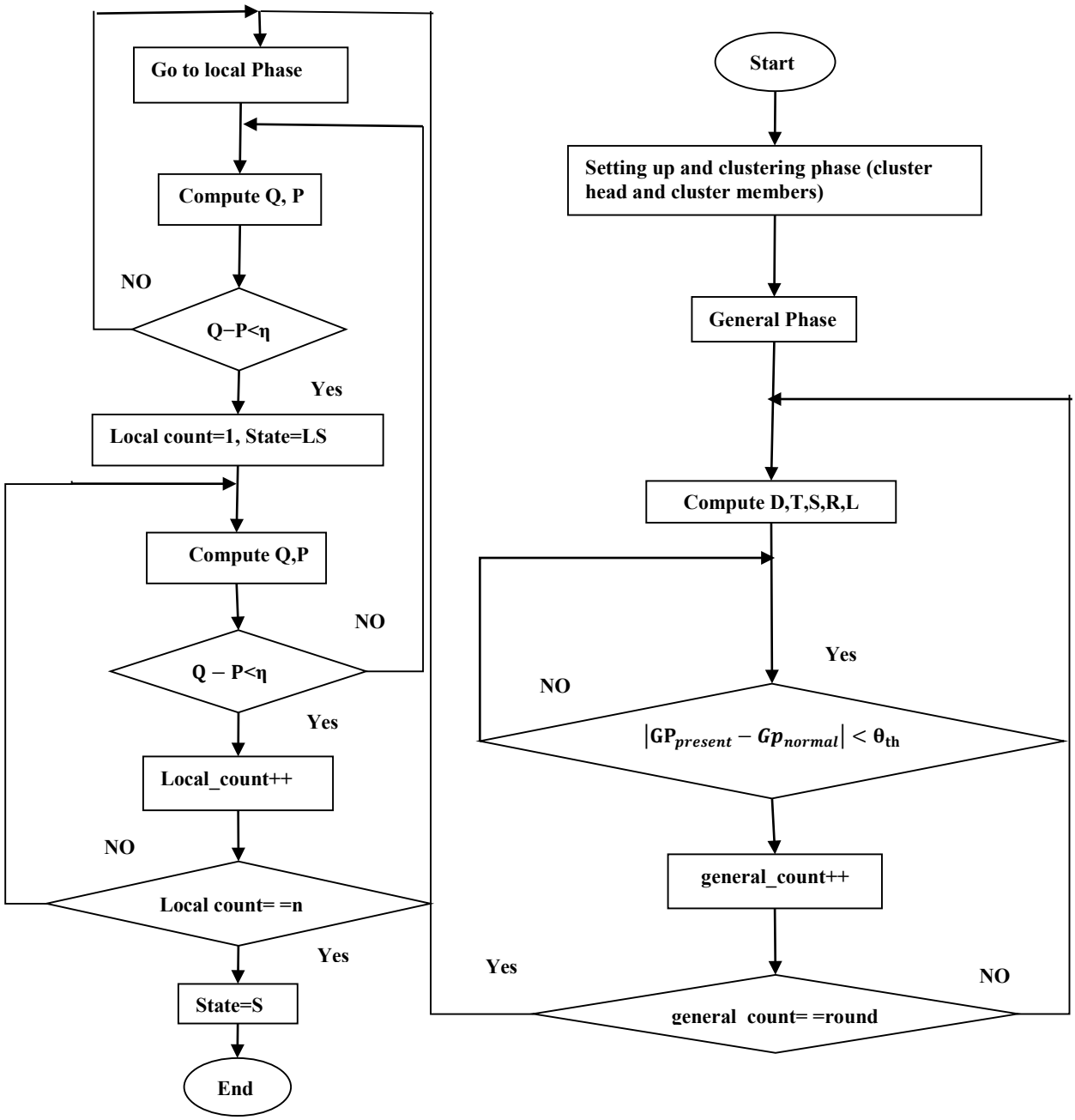

Fig. 4. Flowchart of the proposed method with details.

sent to the other cluster nodes. Even if the new node has a high credit it will not be selected as the cluster head or the assistant node.

Cluster head: if the cluster head node is identified as selfish by the other nodes, or if its energy is finished, then it will be removed from the cluster and the assistant cluster head will be assigned as the new cluster head node.

Cluster head assistant: if the cluster head assistant is known as the selfish node by the other nodes, or if its energy is finished or if it is removed from the cluster then a new cluster head assistant must be selected. To select the new cluster head assistant, the nodes will send their remaining energy and will announce the number of the neighbours in one message. The neighbour node will calculate the scores of the neighbour nodes given the credit according to equation (1). The node that has the highest credit will be selected as the assistant cluster head and it will be announced to the other nodes by a message. The node with the highest vote will be selected as the assistant node and the other nodes of the cluster will follow this decision. Figure 4 shows the flowchart of the proposed method.

IV. Simulation Results

In this section, the proposed method is compared to four similar methods to find out the performance of the algorithm. The simulation is implemented in an 8.1 operating system with

Intel (R) Core (TM) i7 processor at 2.4 GHz and 8 GB internal memory in the MATLAB 2016 software environment.

A simulation area is 150×150 m and consists of 150 nodes, which are randomly distributed. Nine clusters are assumed in the network, the range of node transmission is considered the same, and each node has a unique identifier. Table II presents the simulation conditions.

TABLE II
SIMULATION PARAMETERS

Simulation parameters	Values
Network field 2-D size (area)	150×150 m
Number of mobile nodes (M)	150
Maximum radio range	20 m
Initial energy of a sensor node (E0)	50 J
Nodes speed	10 m/s
Packet size	512 bits
Noise	0.02 dB

The proposed method with detection accuracy, false alarm rate, energy consumption, average end-to-end delay, and network throughput is compared with PBS, FTCSM, ECNBM, and CNBM algorithms. In the most previous methods, if the number of selfish nodes in the network is low, they show good performance and the detection accuracy is high, but in the case the number of selfish nodes increases in the network, they affect the algorithms quickly and their performance goes down, so for comparison algorithms are chosen that, despite the increasing selfish nodes, their performance is acceptable in the network. Table III shows the weights used in the simulation.

TABLE III
WEIGHTS

Weights	Values
C1	0.6
C2	0.2
C3	0.2
α	0.2
β	0.1
θ	0.1
γ	0.2
φ	0.4

D. Detection Accuracy

The accuracy of selfish node detection represents the ratio of self-detected nodes to all nodes. The numerical data resulting from the comparisons indicate that by increasing the selfish node percentage in the network the detection accuracy of the algorithm will increase as compared to the other algorithms. In the proposed method by increasing the number of the selfish nodes in the network, the detection accuracy will remain unchanged and the diagram will vary with a gentle slope and an increase of the selfish nodes will not interrupt the discovery of the selfish nodes. The proposed method can detect more misbehaviour nodes in high repeated rounds. Figure 5 shows the detection accuracy of the proposed method.

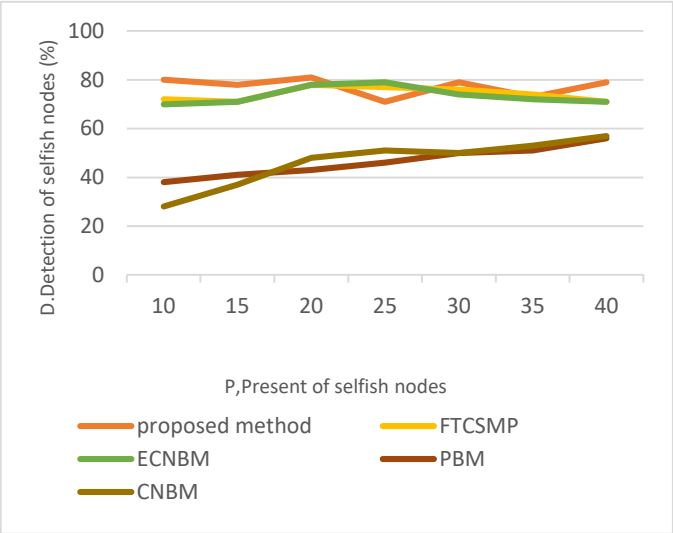

Fig. 5. Selfish node detection.

In higher periods, as the behaviour of nodes is determined the algorithm will detect more selfish nodes. Despite an increase of the number of the nodes in the network, the detection accuracy still remains high because the behavioural patterns of the nodes will be determined over time.

E. False Alarm Rate

False alarm rate represents the ratio of false-selfish-detected nodes to all nodes. If a cooperative node is detected selfish, it will be isolated in the network, so the performance of the algorithm is reduced and fewer packets are delivered. By increasing the number of the selfish nodes in the network, the false alarm rate will remain unchanged. False alarm rate of the proposed method is low as compared to the other algorithms. Figure 6 shows the false alarm rate of the proposed method.

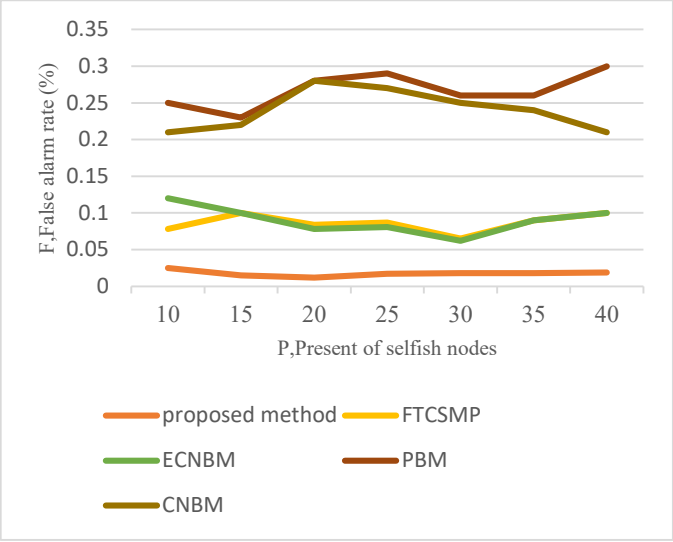

Fig. 6. False alarm rate.

In lower percentage, the false alarm rate is high, but by discovering patterns of the selfish node behaviour, the number of the packets lost and monitoring by the watchdog nodes in higher repetitions, the rate of false alarm is reduced. When node speed decreases in the proposed method, the detection accuracy decreases and the false alarm rate (FAR) increases.

F. Energy Consumption

Energy consumption lies in the number of rounds. Lower energy consumption has shown higher performance of the proposed method. Energy consumption in the number of rounds is high during the simulation, but the average energy consumption is steadily rising. The proposed method consumes less energy compared to similar algorithms. Figure 7 shows energy consumption of the proposed method.

When the number of the selfish nodes is low in the network, lower energy is needed to detect selfish nodes because fewer packets are sent between the nodes. When the number of the selfish nodes is high in the network, higher energy is needed to detect selfish nodes because more packets are sent between the nodes. The proposed method can detect selfish nodes in lower periods, which causes less packet dropping, so the network has less energy to resend packets. As Fig. 7 shows, energy consumption in the proposed method is lower than that in other methods.

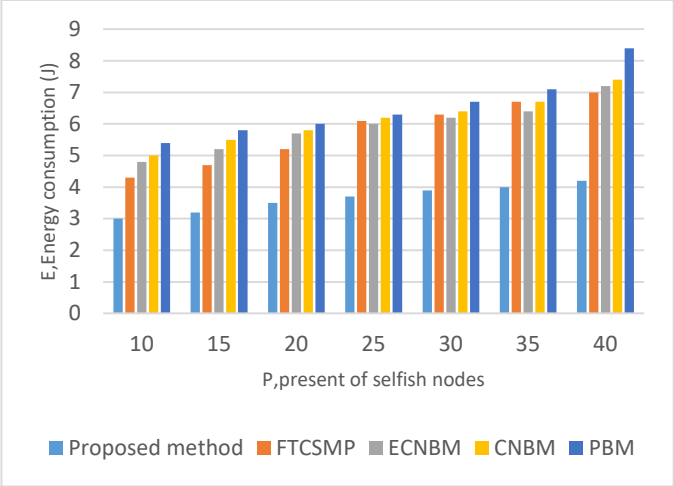

Fig. 7. Energy consumption.

G. Average End-to-End Delay

The average end-to-end delay is the arrival time of a packet from the source node to the destination. The average end-to-end delay of the proposed method is lower than other algorithms. When the number of the selfish nodes increases, the average end-to-end delay is high. As Fig. 8 shows, end-to-end delay in the proposed method is lower than that in other methods.

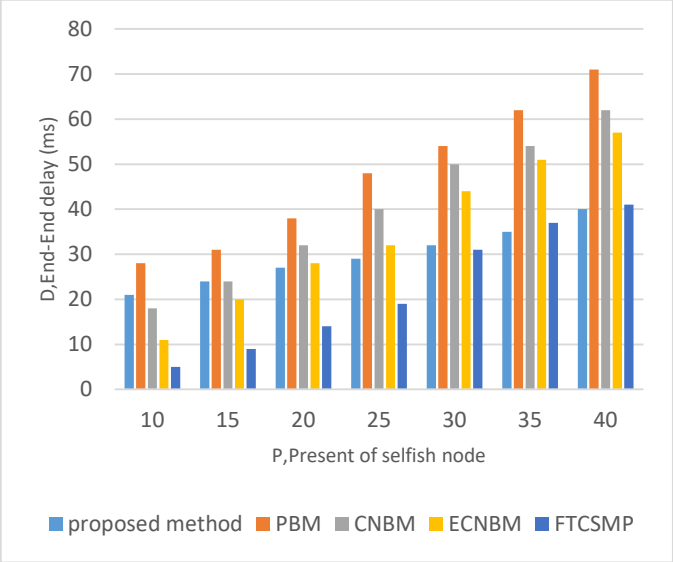

Fig. 8. Average end-to-end delay.

As the number of selfish nodes increases, it takes a lot of time to get a packet to the destination node. If there is the selfish node in the network, the packets are dropped or delivered late by the selfish nodes so the network has to resend the data packets. Retransmitting data packets causes network power loss and reduces network lifetime and increases the average end-to-end delay. When the selfish nodes are rapidly detected, the average end-to-end delay will be reduced.

H. Average Throughput

The average throughput is the number of the packets that arrives to the destination node. Maximum throughput indicates the high performance of the proposed method. As Fig. 9 shows, the average throughput in the proposed method is higher than that in other methods.

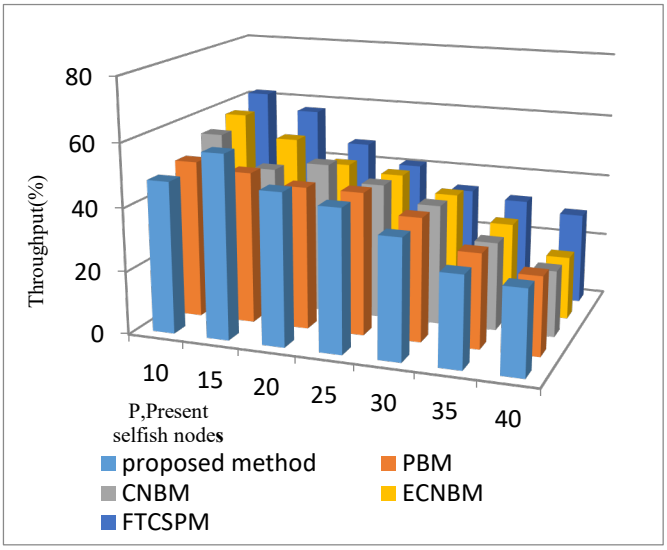

Fig. 9. Average throughput.

The selfish nodes drop the packets or send them late. When the number of the selfish nodes is low, packet dropping is low but the network average throughput is high. If the number of the selfish nodes is high, packet dropping is high and the network average throughput is low. The advantage of the proposed method is that it can detect the selfish nodes at high speeds, so selfish node effects are less on the network. Selfish nodes are detected fast, so fewer packets are dropped than in other methods. The high throughput of the algorithm has shown high performance. The high throughput indicates the optimal bandwidth usage.

V. Conclusion

The existence of the selfish nodes will reduce the performance of the network. The selfish nodes are required to be detected and removed from the network to maintain the security of the network. A credit-based method to detect the selfish nodes is proposed; the proposed approach is compared with parameters such as detection accuracy, false alarm rate, energy consumption, end-to-end delay and throughput. The previous methods consider the cluster head as the cooperate node but in the proposed method the one-hub cluster head neighbours monitor it and if it is known as the selfish node, a new cluster head will be reselected and it is the first advantage of the proposed method and the other one is that when the new node enters the network, the cluster head takes a report from the previous cluster head in relation to the new node and will record it in the credit tables, so the detection rate is much higher than that in the previous methods. The throughput and the end-to-end delay of previous methods are better than that of the proposed method in lower percentage of selfish nodes, but as the selfish nodes increase in the network, the proposed method shows better performance; so the proposed method works well in high percentage of selfish nodes. The disadvantage of the proposed method is that in the case the nodes are isolated and inaccessible in the regions the detection rate is lower because the nodes do not participate in data transmission. For future studies, we suggest that the malicious nodes are also added to the proposed algorithm; after discovering the selfish nodes these nodes can be stimulated for further cooperation so that they can again be inverted to a co-operator node as for using a tit-for-tat method.

References

[1] K. Patel and J. M. Rathod, "Effective Utilization of Bandwidth for Mobile Ad Hoc Network," *Indian J. Sci. Technol.*, vol. 9, no. 27, 2016. https://doi.org/10.17485/ijst/2016/v9i27/92429

[2] S. Stieglitz and C. Fuchß, "Challenges of MANET for mobile social networks," *Procedia Comput. Sci.*, vol. 5, pp. 820–825, 2011. https://doi.org/10.1016/j.procs.2011.07.112

[3] N. Raza, M. U. Aftab, M. Qasim Akbar, O. Ashraf, and M. Irfan, "Mobile Ad-Hoc Networks Applications and Its Challenges," *Commun. Netw.*, vol. 8, no. 8, pp. 131–136, 2016. https://doi.org/10.4236/cn.2016.83013

[4] D. Helen and D. Arivazhagan, "Applications, Advantages, and Challenges of Ad Hoc Networks," *J. Acad. Ind. Res.*, vol. 2, no. 8, pp. 453–457, 2014.

[5] N. Mohammed, H. Otrok, L. Wang, M. Debbabi, and P. Bhattacharya, "Mechanism design-based secure leader election model for intrusion detection in MANET," *IEEE Trans. Dependable Secur. Comput.*, vol. 8, no. 1, pp. 89–103, 2011. https://doi.org/10.1109/TDSC.2009.22

[6] C. Chakrabarti, A. Banerjee and S. Roy, "An observer-based distributed scheme for selfish-node detection in a post-disaster communication environment using delay tolerant network," in *2014 Applications and Innovations in Mobile Computing (AIMoC)*, 2014, pp. 151–156. https://doi.org/10.1109/AIMOC.2014.6785534

[7] U. Shevade, H. H. Song, L. Qiu, and Y. Zhang, "Incentive-aware routing in DTNs," *Proc. Int. Conf. Netw. Protoc. ICNP*, pp. 238–247, 2008. https://doi.org/10.1109/ICNP.2008.4697042

[8] D. Das, K. Majumder, and A. Dasgupta, "Selfish Node Detection and Low Cost Data Transmission in MANET using Game Theory," *Procedia Comput. Sci.*, vol. 54, pp. 92–101, 2015. https://doi.org/10.1016/j.procs.2015.06.011

[9] R. I. Ciobanu, C. Dobre, M. Dascălu, Ş. Trăuşan-Matu, and V. Cristea, "SENSE: A collaborative selfish node detection and incentive mechanism for opportunistic networks," *J. Netw. Comput. Appl.*, vol. 41, no. 1, pp. 240–249, 2014. https://doi.org/10.1016/j.jnca.2014.01.009

[10] M. Schütte, "Detecting Selfish and Malicious Nodes in MANETs," pp. 1– 7, 2006.

[11] S. Bama B, and Indira K., "Detection of Selfish and Malicious Node in Mobile Ad-Hoc Network with NS-2 Using Chord Algorithm," *Int. J. Eng. Technol.*, vol. 9, no. 2, pp. 466–471, 2017. https://doi.org/10.21817/ijet/2017/v9i2/170902326

[12] S. Mittal, and S. Dahiya, "Identification Technique for All Passive Selfish Node Attacks In a Mobile Network," 2015. [Online]. Available: http://www.ijarcsms.com/docs/paper/volume3/issue4/V3I4-0011.pdf [Accessed: December 13, 2018].

[13] D. Li, Y. Xu, and J. Liu, "Distributed relay selection over multi-source and multi-relay wireless cooperative networks with selfish nodes," *Comput. Commun.*, vol. 33, no. 17, pp. 2145–2153, 2010. https://doi.org/10.1016/j.comcom.2010.07.017

[14] P. P. Patel and R. H. Jhaveri, "Various schemes to detect selfishness in wireless ad-hoc networks: A survey," *Proc. 2015 Int. Conf. Green Comput. Internet Things, ICGCIoT 2015*, pp. 881–886, 2016. https://doi.org/10.1109/icgciot.2015.7380587

[15] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks," in *IEEE INFOCOM 2003, Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies,* pp. 1987–1997, 2003. https://doi.org/10.1109/INFCOM.2003.1209220

[16] R. Kaushik *et. al*., "Enhanced node cooperation technique for outwitting selfish nodes in an ad hoc network," *IET Networks*, vol. 4, no. 2, pp. 148–157, 2015. https://doi.org/10.1049/iet-net.2013.0103

[17] F. Wang, F. Wang, B. Huang, and L. T. Yang, "COSR: A reputation-based secure route protocol in MANET," *Eurasip J. Wirel. Commun. Netw.*, vol. 2010:258935, July 2010. https://doi.org/10.1155/2010/258935

[18] N. Samian, Z. A. Zukarnain, W. K. G. Seah, A. Abdullah, and Z. M. Hanapi, "Cooperation stimulation mechanisms for wireless multihop networks: A survey," *Int. J. Adv. Res. Comp. Sc. Soft. Eng.,* vol. 5, no. 8, pp. 306–106, 2015. https://doi.org/10.1016/j.jnca.2015.04.012

[19] S. S. Shinde, B. D. Phulpagar, "A Comparative Study of Selfish Node Detection Methods in Manet," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 5, no. 8, pp. 306–310, 2015.

[20] Z. Li and H. Shen, "Game-theoretic analysis of cooperation incentive strategies in mobile ad hoc networks," *IEEE Trans. Mob. Comput.*, vol. 11, no. 8, pp. 1287–1303, 2012. https://doi.org/10.1109/TMC.2011.151

[21] K. Komathy and P. Narayanasamy, "Best neighbor strategy to enforce cooperation among selfish nodes in wireless ad hoc network," *Comput. Commun.*, vol. 30, no. 18, pp. 3721–3735, 2007. https://doi.org/10.1016/j.comcom.2007.07.004

[22] M. Touati, R. El-Azouzi, M. Coupechoux, E. Altman, and J. M. Kelif, "A Controlled Matching Game for WLANs," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 3, pp. 707–720, 2017. https://doi.org/10.1109/JSAC.2017.2672258

[23] Z. Ullah, M. S. Khan, I. Ahmed, N. Javaid, and M. I. Khan, "Fuzzy-based trust model for detection of selfish nodes in MANETs," *Proc. IEEE 30th Int. Conf. Adv. Inf. Netw. Appl. AINA*, pp. 965–972, 2016. https://doi.org/10.1109/AINA.2016.142

[24] J. Sengathir and R. Manoharan, "A futuristic trust coefficient-based semi-Markov prediction model for mitigating selfish nodes in MANETs," *EURASIP J. Wirel. Commun. Netw.*, vol. 2015:158, 2015. https://doi.org/10.1186/s13638-015-0384-4

[25] K. Komathy and P. Narayanasamy, "A Probabilistic Behavioral Model for Selfish Neighbors in a Wireless Ad Hoc Network," *IJCSNS International Journal of Computer Science and Network Security*, vol. 7, no. 7, July 2007 .

[26] A. Azni, R. Ahmad, Z. Noh, and A. Basari, "Correlated Node Behavior Model based on Semi Markov Process for MANETS," *J. Comput. Sci. Issues*, vol. 9, no. 1, pp. 50–59, 2012.

[27] A. H. Azni, A. Rabiah, and M. N. Zul Azri, "Epidemic Modeling for Correlated Node Behavior in Ad Hoc Networks Center for Advanced Computing Technology," vol. 2, no. 1, pp. 22–30, 2013.

[28] O. A. Wahab, H. Otrok, and A. Mourad, "A cooperative watchdog model based on Dempster-Shafer for detecting misbehaving vehicles," *Comput. Commun.*, vol. 41, pp. 43–54, 2014. https://doi.org/10.1016/j.comcom.2013.12.005

[29] A. Jesudoss, S. V. Kasmir Raja, and A. Sulaiman, "Stimulating truth-telling and cooperation among nodes in VANETs through payment and punishment scheme," *Ad Hoc Networks*, vol. 24, Part A, pp. 250–253, 2015. https://doi.org/10.1016/j.adhoc.2014.08.018

[30] T. Hayajneh, G. Almashaqbeh, and S. Ullah, "A Green Approach for Selfish Misbehavior Detection in 802.11-Based Wireless Networks," *Mob. Networks Appl.*, vol. 20, no. 5, pp. 623–635, 2015. https://doi.org/10.1007/s11036-015-0605-4
D. Djenouri, N. Ouali, and A. Mahmoudi, "Random Two-hop ACK to Detect Uncooperative Nodes in MANETs," *Security*, vol. 2, no. 7, pp. 165–175, 2017.

[31] D. AbdelMohsen, and T. Abdelkader, "Detecting selfish nodes and motivating cooperation, Mobile Ad-hoc Networks," *Computer Engineering & Systems (ICCES)*, 2015 IEEE Tenth International Conference, pp. 301–306, 2015. https://doi.org/10.1109/icces.2015.7393064

[32] Z. Ullah *et. al*., "Fuzzy-Based Trust Model for Detection of Selfish Nodes in MANETs," *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, 2016. https://doi.org/10.1109/AINA.2016.142

[33] K. Chaker, A.Lakas, and N. Lagraa. "Detection of intelligent malicious and selfish nodes in VANET using threshold adaptive control," *2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA)*, 2016. https://doi.org/10.1109/icedsa.2016.7818492

[34] C. Nimje, and P. Junghare, "A review on node activity detection, selfish & malicious behavioral patterns using watchdog algorithm," *2017 International Conference Inventive Systems and Control (ICISC),* 2017. https://doi.org/10.1109/icisc.2017.8068663

[35] A. Lupia, "TEEM: Trust-based Energy-Efficient Distributed Monitoring for Mobile Ad-hoc Networks," *2017 Wireless Days,* IEEE, pp. 133–136, 2017. https://doi.org/10.1109/WD.2017.7918128

[36] S. Adabi, S. Jabbehdari, A. Rahmani, and S. Adabi, "A novel distributed clustering algorithm for mobile ad-hoc networks," *J. Comput. Sci.*, vol. 4, no. 2, pp. 161–166, 2008. https://doi.org/10.3844/jcssp.2008.161.166

Sanaz Nobahary received her B.Sc. degree in Software, Computer Engineering from Islamic Azad University, Tabriz, Iran, in 2014. She received her M.Sc. degree in Software, Computer Engineering from Islamic Azad University, Tabriz, Iran, in 2017. Her research areas include WSN, Manet, Vanet, IOT, security, and cloud computing. She is a co-author of several research papers in these areas.

Shahram Babaie received his B.Sc. in Computer Engineering from Sajad University in 2003 and the M.Sc. and Ph.D. degrees in Architecture of Computer from Islamwic Azad University. Currently, he is an Assistant Professor at the Department of Computer Engineering of the Islamic Azad University of Tabriz. His research interests include computer networks, ad-hoc and wireless sensor networks, and information security.
ORCID iD: https://orcid.org/0000-0001-7830-2496