FUZZY OPTIMIZATION AND MODELLING 4(4) (2023) 32-47

E-ISNN: 2676-7007

Contents lists available at FOMJ

Fuzzy Optimization and Modelling

Journal homepage: http://fomj.qaemiau.ac.ir/

Paper Type: Research Paper

A Novel Selfish Node Detection Based on Fuzzy System and Game Theory in Internet of Things

Gholam Hossein Abdi[a], Amir Hosein Refahi Sheikhani[a,*], Sohrab Kordrostami[a], Shahram Babaie[b]

[a] *Department of Applied Mathematics and Computer Mathematics, Lahijan Branch, Islamic Azad University, Lahijan, Iran*
[b] *Department of Computer Engineering, Lahijan Branch, Islamic Azad University, Lahijan, Iran*

ARTICLE INFO

Article history:
Received 29 September 2023
Revised 24 December 2023
Accepted 3 January 2024
Available online 3 January 2024

Keywords:
Internet of Things (IoT)
Selfish and Malicious Node
Fuzzy Logic
Game Theory

ABSTRACT

Internet of Things describes a situation in which a large number of devices (things) are connected through a number of sensors via Internet, and lack of cooperation of some nodes in providing service to other nodes might interrupt the connection of some things, degrading network efficiency. A multi-phase mechanism based on Game theory and direct/indirect fame has been designed to motivate the selfish and malicious nodes to cooperate in IoT, which begins by deploying nodes in the IoT network. In the first phase, the nodes are grouped into clusters with cluster-heads for data collection. In the second phase, a multiplayer and dynamic game is executed while forwarding their data packet or others' data packet. Nodes can pick their strategy when data packet forwarding in the third phase (Fuzzy logic reputation). Nodes will determine the neighboring node reputation by using fuzzy system. The amount of reputation of each of the nodes has been realized and finally, with the help of second phase and fuzzy logic, each node is decided to be cooperate or selfish nodes and in case of head clusters and fuzzy logic in some cases, the opportunity node will be reestablished to cooperate in network activities otherwise the node will be isolated. The effectiveness of the proposed solution has been assessed and the parameters of non-cooperative node detection accuracy, positive and negative warning rates, network PDR, and average end-to-end latency perform better compared to other previous methods.

1. Introduction

Internet of things (IoT) is a novel concept and pattern in the world of information and communication technology, where lack of cooperation of some nodes for providing service to other nodes might interrupt the connection of some nodes with each other. The main IoT applications are applied in a wide range of areas, including smart cities, environment, security devices, agriculture, industrial control, etc. One of the issues in IoT

* Correspondig author
E-mail address: ah_refahi@yahoo.com (Amir Hosein Refahi Sheikhani)

DOI: 10.30495/FOMJ.2024.1996589.1122

security is the non-cooperative nodes [7, 8].

In IoT, connections are achieved via cooperation with others. An important challenge that threats IoT networks is the lack of cooperation of some nodes in data transmission in multi-hop transmission, where these nodes are called non-cooperative (selfish or malicious). As the number of these nodes increases, the network efficiency decreases and the network performance is degraded [1, 9, 12].

Considering the performance of non-cooperative nodes in IoT, there are various non-cooperative nodes in this network, which are classified as normal, selfish, and malicious nodes. The normal nodes continue their normal activity to transmit and receive packets. Selfish nodes want to achieve the maximum profit and network channels and misuse other nodes despite they know that they can increase their lifetime through cooperation with other nodes. Malicious nodes are the nodes that carry out subversive objectives in the network [2, 3, 5, 6].

In this study, a multi-phase mechanism is designed to detect selfish nodes based on game theory [13] in a multiplayer game in IoT. The proposed mechanism is a multi-phase scenario in which the clusters have a cluster-head for transmission to the base station. The proposed mechanism is based on game theory and fame. When a node discards the data packet selfishly, it prevents cooperation with that node by reducing its fame, motivating the selfish node to cooperate, and the selfish node finds the opportunity to increase its fame through cooperation with other nodes, providing the opportunity for other nodes to cooperate.

Assumptions of the model are as follows:

1) Each object in the network is considered as a node.
2) Each node is aware of the set of neighboring nodes in its neighborhood.
3) Each node might be normal, transmits with latency, or does not transmit. The nodes are not aware of their neighboring nodes' nature.
4) The radio frequency range of the nodes is limited. Thus, the nodes cooperate with each other to transmit the data packets to the destination.
5) Each node operates intelligently; in other words, it uses all information and expectations of other nodes to find the best strategy.
6) The nodes obtain information about cooperation of the neighboring nodes during the network performance.
7) Each node tries to obtain the best and maximum profit in the network, such that it tries to interact with the normal nodes to transmit its data packet to the destination.

First, the tree and table of the multiplayer game between a limited number of nodes is constituted between nodes M1 and M2, and then the tree and table of other nodes are represented, and the corresponding Nash equation is calculated. Since, we do not know if the node is normal, selfish, or malicious at the beginning, the probability of being normal, mal

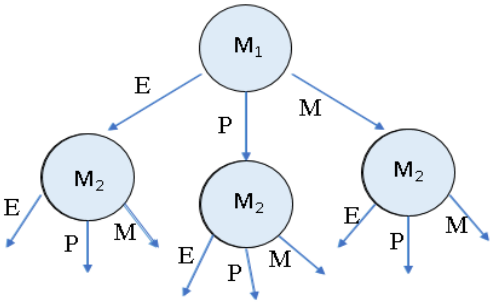

Figure 1. Tree of Game probability with two nodes

The tree corresponding to two nodes of one cluster is shown in Figure 1, and the corresponding table to M1 and M2 is represented in Table 1.

Table 1. Game probability table for node M_1 and M_2

		h_1		h_2		h_3	
	M_1 / $M2$	E		H		M	
h_1	E	$W_1(E,E)$	$W_2(E,E)$	$W_1(E,H)$	$W2(E,H)$	$W_1(E,M)$	$W_2(E,M)$
h_2	H	$W_1(H,E)$	$W_2(H,E)$	$W_1(H,H)$	$W2(H,H)$	$W_1(H,M)$	$W_2(H,M)$
h_3	M	$W_1(M,E)$	$W_2(M,E)$	$W_1(M,H)$	$W2(M,H)$	$W_1(M,M)$	$W_2(M,M)$

For simplicity, it is assumed that cluster m only includes two nodes is obtained as follows:

$$t_1 + t_2 + t_3 = 1 \quad , \quad h_1 + h_2 + h_3 = 1 \tag{1}$$

The expected profit for node M_1 and M_2 is proportional to the total game profit for all cases, which is as follows:

$$\begin{aligned} \pi_1(E,M) = {} & h_1 * t_1 * W_1(E,E) + h_2 * t_1 * W_1(P,E) + h_3 * t_1 * W_1(M,E) \\ & + h_1 * t_2 * W_1(E,P) + h_2 * t_2 * W_1(P,P) + h_3 * t_2 * W_1(M,P) \\ & + h_1 * t_3 * W_1(E,M) + h_2 * t_3 * W_1(P,M) + h_3 * t_3 * W_1(M,M) \end{aligned} \tag{2}$$

$$\begin{aligned} \pi_2(E,M) = {} & h_1 * t_1 * W_2(E,E) + h_2 * t_1 * W_2(P,E) + h_3 * t_1 * W_2(M,E) \\ & + h_1 * t_2 * W_2(E,P) + h_2 * t_2 * W_2(P,P) + h_3 * t_2 * W_2(M,P) \\ & + h_1 * t_3 * W_2(E,M) + h_2 * t_3 * W_2(P,M) + h_3 * t_3 * W_2(M,M) \end{aligned} \tag{3}$$

If selecting one player is the best among all optimal selections, a Nash equilibrium is achieved. Violation of each player form its Nash equilibrium, results in worse consequences. Thus, none of the player tends to violate its Nash equilibrium (see Figure 2).

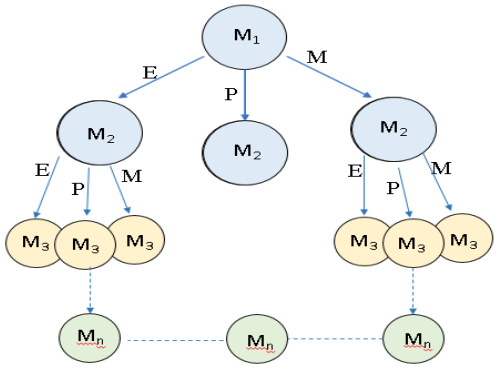

Figure 2. Three of game probability by *n* nodes

The game with mixed strategies and M nodes is represented as $G_{ME} = \{M, (\Delta(S_i)), (w_i)\}$, in which $\Delta(S_i)$ is defined as follows:

$$\Delta(S_i) = \begin{Bmatrix} (h_{i1}, \dots, h_{im}) \in R^m : h_{ij} \geq 0 \\ , \forall j = 1, \dots, m \\ , \sum_{j=1}^{m} h_{ij} = 1 \end{Bmatrix} \tag{4}$$

Expectation of other nodes is also calculated. By setting a hybrid Nash equilibrium is achieved (Matlab and other mathematical software can solve the multi-polynomial equations). By solving the hybrid Nash equilibrium, the probability of selecting each player is calculated. By obtaining these values at each step of the game, each player selects the best action against the opponent. Since none of the nodes knows when the game is finished, a game is an infinite repetition with M players. In a repetitive game, at the beginning of the kth round, the ith node decides based on past behavior of the nodes.

The paper proceeds as follows. In Section 2, a multi-phase mechanism based on game theory and direct/indirect fame is designed to motivate the selfish and malicious nodes to cooperate in IoT. Section 3 evaluates the effectiveness of the proposed solution and summarizes the main results. Section 4 concludes the paper.

2. The proposed method for detecting selfish nodes

2.1. Deployment and clustering phase

The IoT nodes that are distributed in the environment randomly, detect their neighboring nodes by sending a Hello message. The Hello request message is broadcast globally, and the neighboring nodes send information in response to their neighbors upon receiving the Hello message. By receiving this information, each node constitutes a table, called the neighborhood table, which includes node ID, node distance, number of neighbors, energy level of the node, and node status as shown in Figure 3.

Node status	Energy level of the Node	Number of neighbors	Node distance	Node ID

Figure 3. The information framework of each node in the cluster-head

The details of the fields shown in Figure (3) are discussed in the following.

- Node ID: this field is 4 bytes and it is used to store the node ID. Considering the increasing number of nodes in this network, this number of bits can address more than 4 million nodes.
- Node distance: this field is 2 bytes, and it is used to store the number of hops of a node to the main cluster-head.
- The number of neighbors: this filed is 2 bytes, and it is used to store the number of neighbors of a node. Considering the advancements in increasing the radio range of these nodes, this might include a large number of neighbors.
- Energy level of the node: this field is one byte and represents the residual energy level of the node. It is obvious that the energy level of the nodes is low considering the presence of batteries, and its value can be stored in one byte of memory.
- Node status: this field is one byte and describes the direct fame of the neighboring node, which is stored in the node's database, describing cooperation of the neighboring node. The content of this field is 1 by default.

Thus, each neighboring node requires 10 bytes of memory to store its information in the node's database and use it in the subsequent phases for calculating the indirect fame and checking the node status.

In this method, all objects are considered as a node; first, the centers of the initial clusters are determined based on distance and data density is selected by overcoming the local optimal and ensuring the unique clustering result instead of selecting them randomly. Also, instead of using the minimum inter-cluster entropy, the worst cluster is found to improve clustering accuracy. Since convergence or similarity is the main criterion in this clustering algorithm, it can be measured by distance, probability density and etc. Mean distance and mean data density are combined to determine the initial clustering. The candidates are selected canonical to prevent noise of the initial clusters. If the worst clusters continue, they can be combined with other clusters to present the best clustering for IoT [14].

2.2. Data transmission phase and implementing multiplayer game

When a source node has a data packet to transmit to a certain destination or cluster head, a non-cooperative game is performed along the data transmission path, and each node along the path seeks to receive more profit and loose minimum personal profit. After establishment phase and clustering, all nodes have constituted their neighborhood table, and the cluster heads are determined. At the beginning of the game, no player is aware of cooperation of other nodes and the game is performed with all neighbors of a node. During the network lifetime, the game is performed many times, and if a packet is discarded due to error or environmental factors, it can be detected by repeating the game. Although, it is suggested to perform the game for detecting the cooperative nodes from different neighbors of nodes and different paths in each round, but the problem that occurs is the existence of infinite paths that do not reach the destination and only cause closed network circulation and energy consumption. To prevent this problem, packet delay limits and data packet hops are used along the path.

After data packet transmission during the game; the destination node transmits the authentication message to the source. The source node and the intermediate nodes can get information about the status of their neighboring nodes, whether the nodes cooperate or not. In each round of the game, the game results in the third phase are analyzed to determine the direct and indirect fame of the neighboring node.

The result of these analyzes change and update the fame tables of neighboring nodes so that the node delivers its data packet to the destination by selecting a neighbor in other games with a higher probability. In other words, each node transmits the result of successful and unsuccessful data packet transmission in each round of the game to be stored by in the corresponding fame table by the neighboring nodes. Thus, players play more cautiously, as a result of which the non-cooperative node cannot reduce network efficiency by unsuccessful data packet transmission, increase the mean data packet latency and inadvertent use of network bandwidth (Figure 4).

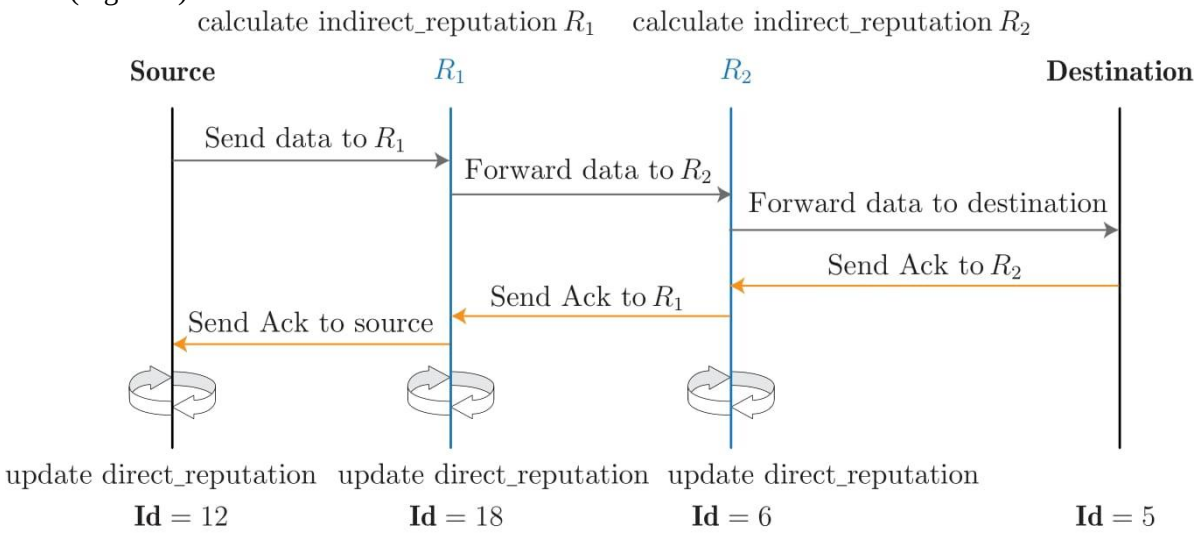

Figure 4. Sending data from source node to destination node

When the source node, like *i*, needs a data packet to transmit, this data packet might be a node for the node or other nodes. Thus, if the node is aware of its neighboring nodes and has played with them before, it can check the indirect fame of the neighboring nodes to increase the probability of successful data packet transmission and use the profit of data packet transmission to increase its direct fame. Direct and indirect fame values are updated in the third phase of the previous game and stored in the fame tables. Considering these values, the node can select the best strategy and optimize its conjecture about the cooperation status of the neighboring nodes. But if node *i* is not aware of the status of all of its neighboring nodes, it continues the game with a neighboring node, about which it has no information, and this procedure is also done for the middle nodes so that all nodes have a chance to play the game and participate in data packet transmission operations and remote nodes can participate in network operations; to determine the nature and status of the nodes' cooperation, and the energy consumption distributed among the nodes throughout the network in a distributed manner. Figure 4 shows performing the operations and data packet transmission by the source node.

2.3. Fuzzy logic reputation

In the decision - making process to facilitate the selection of an appropriate option among existing solutions, the real numbers are first converted into fuzzy terms. In this case, the user will have a more clear understanding of the level of an attribute relative to the extent of its domain. For example, if the number of missed packets can be expressed as a real number, the user cannot judge either too much or lack of it, unless the attribute domain can observe the number of missing packets and then comment on it. However, the expression of the attributes in the form of fuzzy terms helps the user approximate the extent of its value regardless of the values of the variable in domain. The simplest method for converting real numbers into fuzzy terms is to use expert opinions, but it is not always possible because the expert is not always available. Another method of using membership functions

such as functions and trapezoidal, triangular.

The amplitude of the input variable is divided into triangular intervals, and each interval represents an expression quantity, the true value of the input variable is converted into linguistic term, which has the closest distance with the corresponding range. In some of the past work, different fuzzy parts have been used for different features, different features may differ from different types (Continuous, ordinal, and relative). We need at least four key parameters to determine the reputation of nodes because determining the selfishness of nodes and isolating the nodes is not only by specifying one or two parameters.in the proposed method, we use four parameters: number of dropped packets, mean delay, residual energy, cooperation history as a fuzzy system input to consider the effect of more parameters and more efficient parameters in detecting selfish nodes. In Figure 5, the schematic diagram of the fuzzy system is plotted against four input parameters and ultimately determines the system output that identifies the reputation of each node.

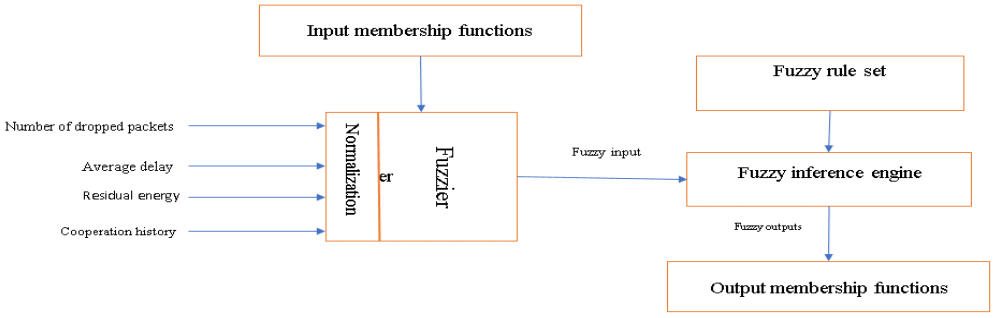

Figure 5. Fuzzy logic system to detect selfish node

2.4 Normalization process of parameters

In the following, we apply the membership functions of four input parameters, considering three levels for each one, For the number of omitted packets, three levels (High, Medium, Low) three levels of delay average parameters (High, Medium, Low) and the third parameter where the remaining energy of the node has three levels (High, Medium, Low) and last parameter which defines cooperation history has three levels (Strong, Medium, weak) whose diagrams are shown in Figures 6 and 7. All parameters have different values and we use the relation (5) to normalize the numbers, after normalization of all numbers between the range [0,1].

$$X_n = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{5}$$

In equation (5), X is the value of parameters that are normalized to use in fuzzy system and X_{min} is the minimum value of that parameter and X_{max} is the maximum value of it. In the form of membership functions, if we are to compute the distance between the membership functions centers, we use the formulation (6) where n represents the number of membership functions. *Max* is the highest level of the interval and *Min* is the lowest level.

$$Interval = \frac{Max - Min}{n - 1} \tag{6}$$

The number of membership functions per one of the fuzzy system inputs is 3, so the distance between the membership functions centers is calculated as formulation (7).

$$Interval = \frac{Max - Min}{n - 1} = \frac{1 - 0}{3 - 1} = \frac{1}{2} = 0.5 \tag{7}$$

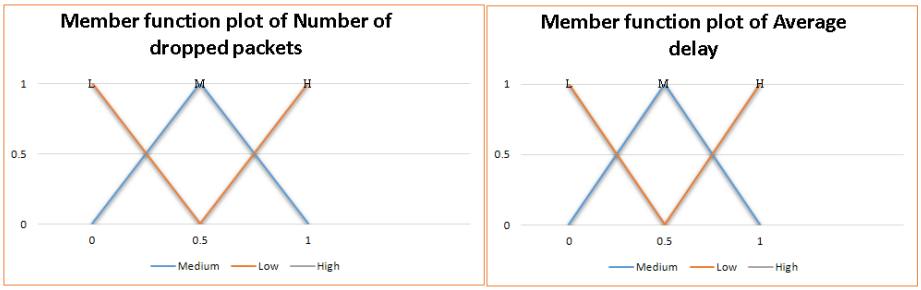

Figure 6. Average delay and number of dropped packets member function chart

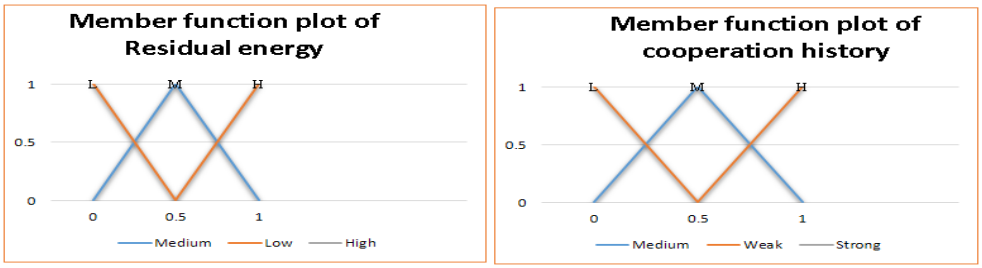

Figure 7. Cooperation history and residual energy member function chart

2.5. Fuzzy rules base

How to compute the number of fuzzy rules follows, we multiply the number of membership functions of all inputs. Equation (8) has shown the number of fuzzy rules and it computed 81 rules to detect selfish nodes.

$$N = N_{DP} * N_{AD} * N_{RE} * N_{HS} = 3 * 3 * 3 * 3 = 81 \quad (8)$$

In Equation (8), N_{DP} is the dropped packets number of member function, N_{AD} is the average delay parameter number of member function, N_{RE} is the residual energy of nodes parameter number of member function and finally N_{HS} is the history status as reputation parameter number of member function. All of them are equal to 3 and means they have 3 level for all parameters. We write different scenarios of each input parameters, in Table 2. Rules 'table indicates how much the reputation of each node acts as a selfish node which must be isolated or stimulate to cooperate.

Table 2, which is the fuzzy rules table, shows that 81 different modes of nodes parameters conditions indicate that VVVH is the best node (normal node) in the network. Thus, in the last row the VVVL node is selfish node in the network, which the nodes will decide about the status of the neighbor nodes. The purpose of this system is to identify the reputation of each node. R output variable defines the degree of selfishness of each node and allocates different values that represent different modes of reputation of each node according to Table 3.

After gaining the reputation of each node by fuzzy system (Phase III) and gathering comments from the second phase (nodes that play game to determine the status of neighbor nodes) to decide to end each node the results of the fuzzy system and the game results together. Table 4 shows the results of fuzzy output and neighbor nodes output situation.

Table 2. Fuzzy role database

Number of dropped packets / Average delay	Cooperation history / Residual energy	*Strong*	*Medium*	*Weak*
Low / Low	*High*	*VVVH*	*VVH*	*VH*
	Medium	*VVH*	*VH*	*H*
	Low	*VH*	*H*	*VVVG*
Medium / Low	*High*	*H*	*VVVG*	*VVG*
	Medium	*VVVG*	*VVG*	*VG*
	Low	*VVG*	*VG*	*G*
High / Low	*High*	*VG*	*G*	*M*
	Medium	*G*	*M*	*L*
	Low	*M*	*L*	*VL*
Low / Mediu	*High*	*VVH*	*VH*	*H*
	Medium	*VH*	*H*	*VVVG*
	Low	*H*	*VVVG*	*VVG*
Medium / Medium	*High*	*VVVG*	*VVG*	*VG*
	Medium	*VVG*	*VG*	*G*
	Low	*VG*	*G*	*M*
High / Medium	*High*	*G*	*M*	*L*
	Medium	*M*	*L*	*VL*
	Low	*L*	*VL*	*VVL*
Low / Low	*High*	*VH*	*H*	*VVVG*
	Medium	*H*	*VVVG*	*VVG*
	Low	*VVVG*	*VVG*	*VG*
Medium / Low	*High*	*VVG*	*VG*	*G*
	Medium	*VG*	*G*	*M*
	Low	*G*	*M*	*L*
High / Low	*High*	*M*	*L*	*VL*
	Medium	*L*	*VL*	*VVL*
	Low	*VL*	*VVL*	*VVVL*

Table 3. Output fuzzy sets range - reputation rate of each node

Symbol	Description
Very Very High (VVH)	Having Maximum reputation (100% cooperative node)
Very High (VH)	Having Above reputation (100% cooperative node)
High (H)	Having High reputation (100% cooperative node)
Medium (M)	Having Average reputation (Maybe cooperative node, need to be checked)
Low (L)	Having Low reputation (Having selfish behavior)
Very Low (VL)	Having very Low reputation (Having selfish behavior)
Very Very Low (VVL)	Having Little reputation (Having selfish behavior

Table 4. Fuzzy output and neighbor nodes output situation

Neighbor node decision / Fuzzy logic output	Cooperative	Non-cooperative
VVH	cooperative	Given second chance
VH	cooperative	Given second chance
H	cooperative	Given second chance
M	Given second chance	Given second chance
L	Selfish	Selfish
VL	Selfish	Selfish
VVL	Selfish	Selfish

3. Evaluation and simulation results

An IoT network has been distributed uniformly and randomly in an environment including dimensions of 1000 * 1000 square meters and there are nodes available for 4 types of networks and there are also different IoT nodes with different numbers and parameters in the environment. There has been a station to collect data in the performed simulation in the center of all four types of networks, and the considered Internet network has been assumed to have motionless objects with limited energy source comparable to wireless sensor networks with four various types of nodes that can be used in agricultural land (controlling water, Soil, air, and temperature). There are wireless connections in all these networks. It should be noted that there is a different model to simulate for each different network; hence, first, we will have clustering by determining the cluster-heads according to the clustering algorithm [14] shown in Figure 8.

Four stations have been considered to simulate in the center of each type of network in order to collect data. The performance of plants has been monitored and controlled using the fixed objects similar to wireless sensor networks with different radio ranges and initial energy levels including four types of sensors in this network with sensory capability in agricultural land. As the mentioned cases explain, different areas have the fixed base station and, in the position, (250, 250), (750, 750), (250,750), (750,250), respectively.

We have repeated the simulation during 100 periods and calculated the energy consumption based on the values listed in Table 5, but it is assumed that the initial energy for the network nodes to be 0.5, 10, 1, and 150 joules, respectively, and 200, 100, 150, 20 with a radio range 80, 70, 90, 70 meters, respectively. But there is a similarity between the energy consumption model and the type of nodes.

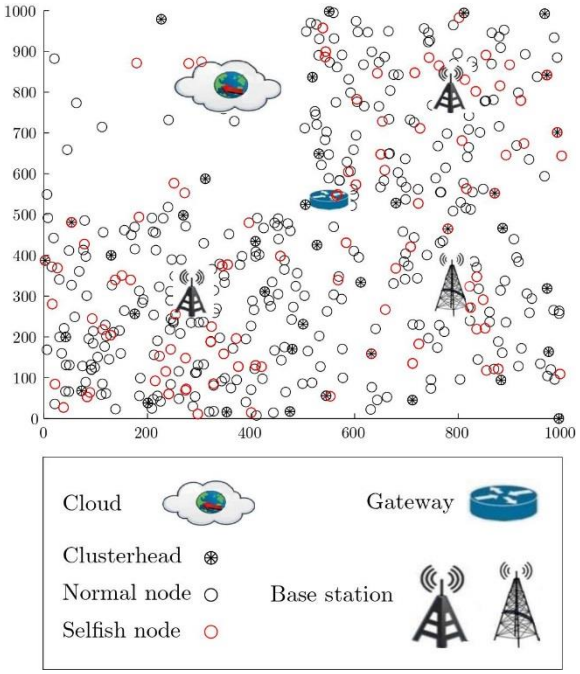

Figure 8. Network nodes in simulated environment

Table 5. Feature of network nodes

Parameters	Values
E_{elec}	50 nJ/bit
E_{DA}	5 nJ/bit/signal
ε_{fs}	10 pJ/bit/m^2
ε_{mp}	0.0013 pJ/bit/m^4
Packet size	1024 bits
d_0	87 m
Initial energy of nodes	0.5 Joule
	10 Joule
	1 Joule
	150 Joule

3.1 Evaluation of designed fuzzy system

The fuzzy system is designed with 4 inputs and an output. The number of missing data packets, the average packet delay, and the remaining energy of the nodes and the association history of the nodes are in forwarding packets. The output of the system shows the degree and level of interoperability within the network, which are stored in head cluster nodes. The proposed protocol can determine the highest level of cooperation of nodes and their lowest cooperation level. The input and output of the fuzzy system and its maximum and minimum are shown in Figure 9 and Table 6 respectively.

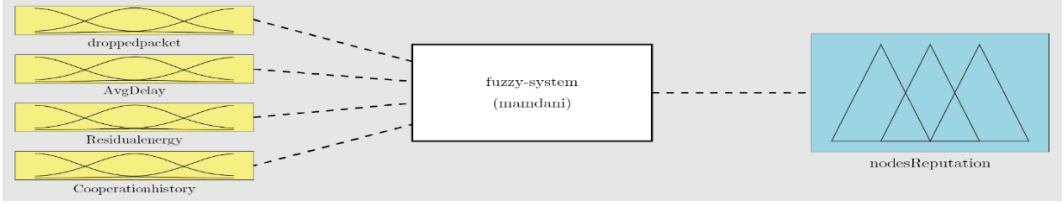

Figure 9. Fuzzy system model with inputs and outputs

Fuzzy system is done on inputs and outputs: The triangular membership functions are used to make Fuzzy the inputs. For different inputs, the fuzzy variable and its input range are shown in Table 6. The four input parameters include the number of missing packets, the mean of delay, remained energy, cooperation history. In other words, the amount of linguistic variable can range from that domain .The number of discarded packets by each node as a linguistic variable with the range of ϵ (Number of dropped packets) T, U [1, 800] domain is 1 to 800 packets. The range is divided into three levels. Each variable in the number of dropped (T) packet is described by a fuzzy set, such as U= [1, 800].Therefore, each level has a fine function that represents the degree of dependency of each value to this level.

Table 6. Input parameters and their interval in proposed Fuzzy system

level	Interval of each level
Number of dropped packets	
L1(Low)	1-200
L2(Medium)	200-700
L3(High)	700-800
Average delay	
L1(Low)	1-20
L2(Medium)	20-40
L3(High)	40-60
Residual energy of nodes	
L1(Low)	0.1j-0.5j
L2(Medium)	0.5j-0.8j
L3(High)	0.8j-1j
History of nodes ‘cooperation	
L1(Weak)	1-40
L2(Medium)	40-80
L3(Strong)	80-100

In the following, we apply the membership functions of four input parameters, considering three levels for each one, For the number of omitted packets, three levels (High, Medium, Low) three levels of delay average parameters (High, Medium, Low) and the third parameter where the remaining energy of the node has three levels (High, Medium, Low) and last parameter which defines cooperation history has three levels (Strong, Medium, weak) whose diagrams are as follows. The optimization of these algorithms is often done through test and error method to achieve the desired performance of the designed fuzzy system. But there is only one output, the level of collaboration of the node, and, like the inputs, the membership function is assigned as a triangular function. In Figures 10 and 11, the membership functions of the inputs and the only output of the fuzzy system have been represented.

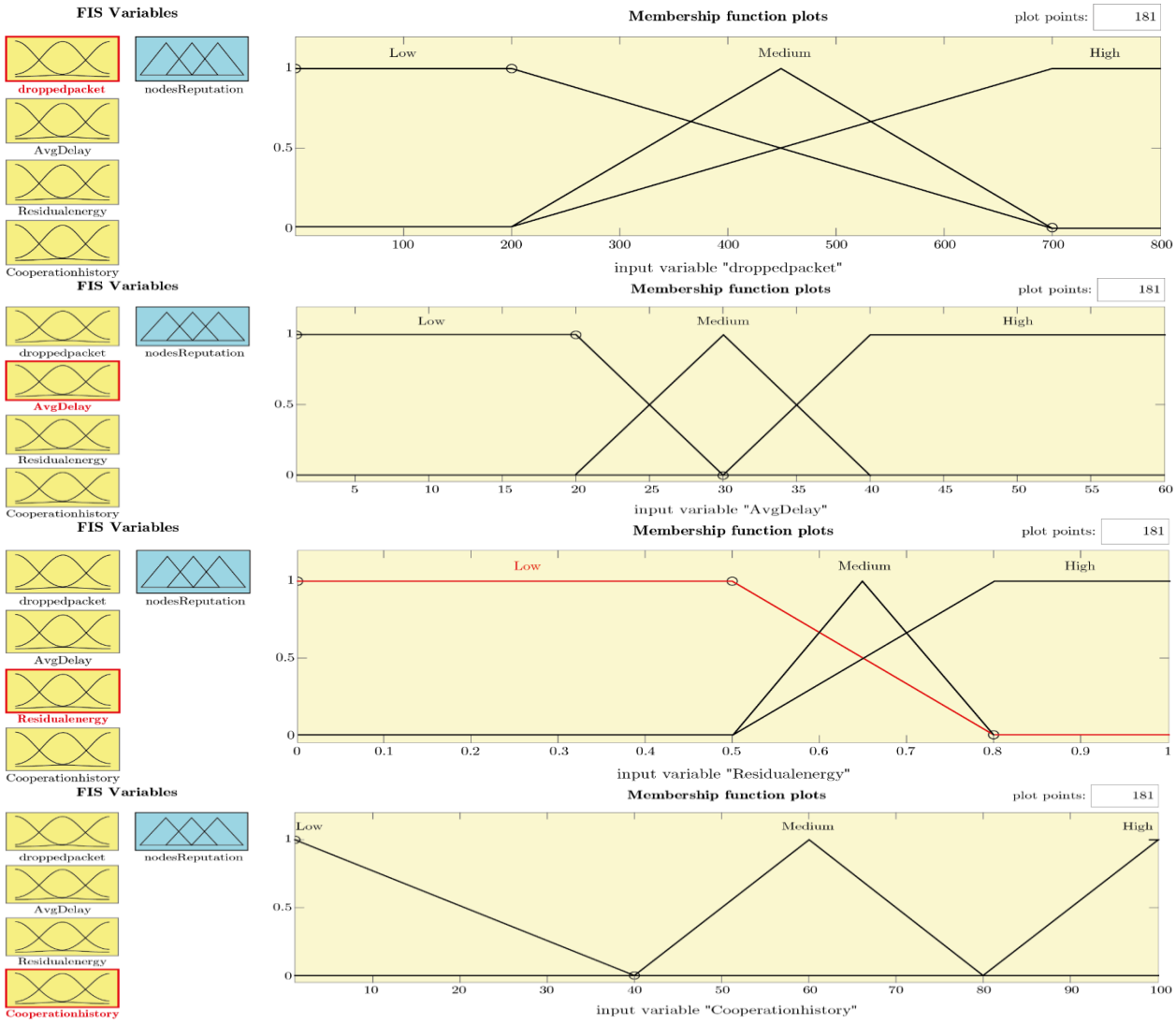

Figure 10. Fuzzy system of inputs membership function

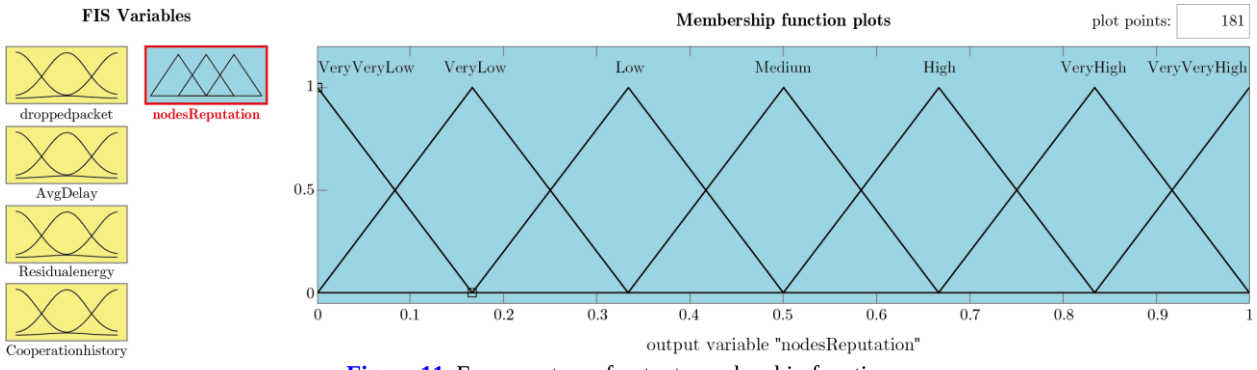

Figure 11. Fuzzy system of output membership function

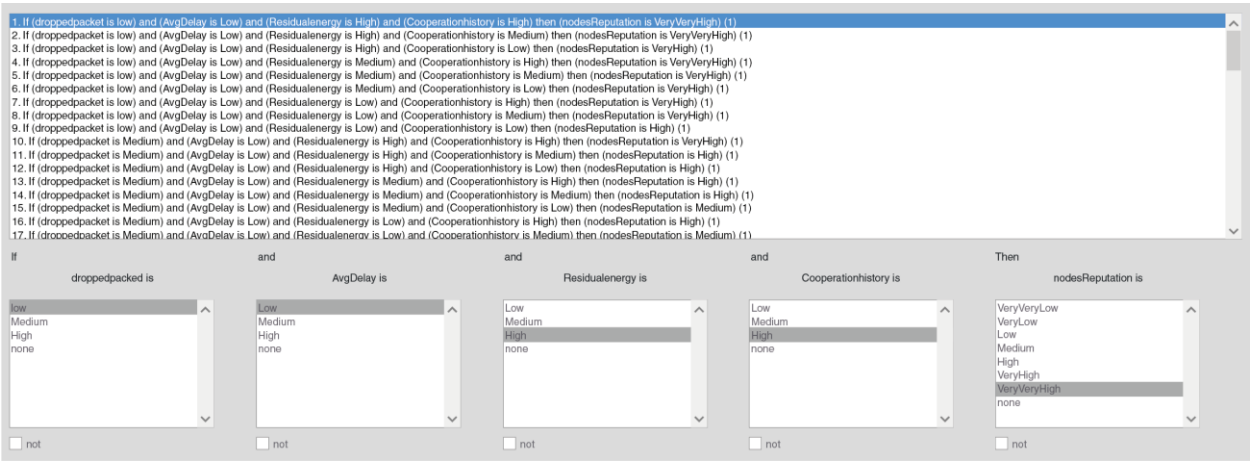

Figure 12. Fuzzy system rules

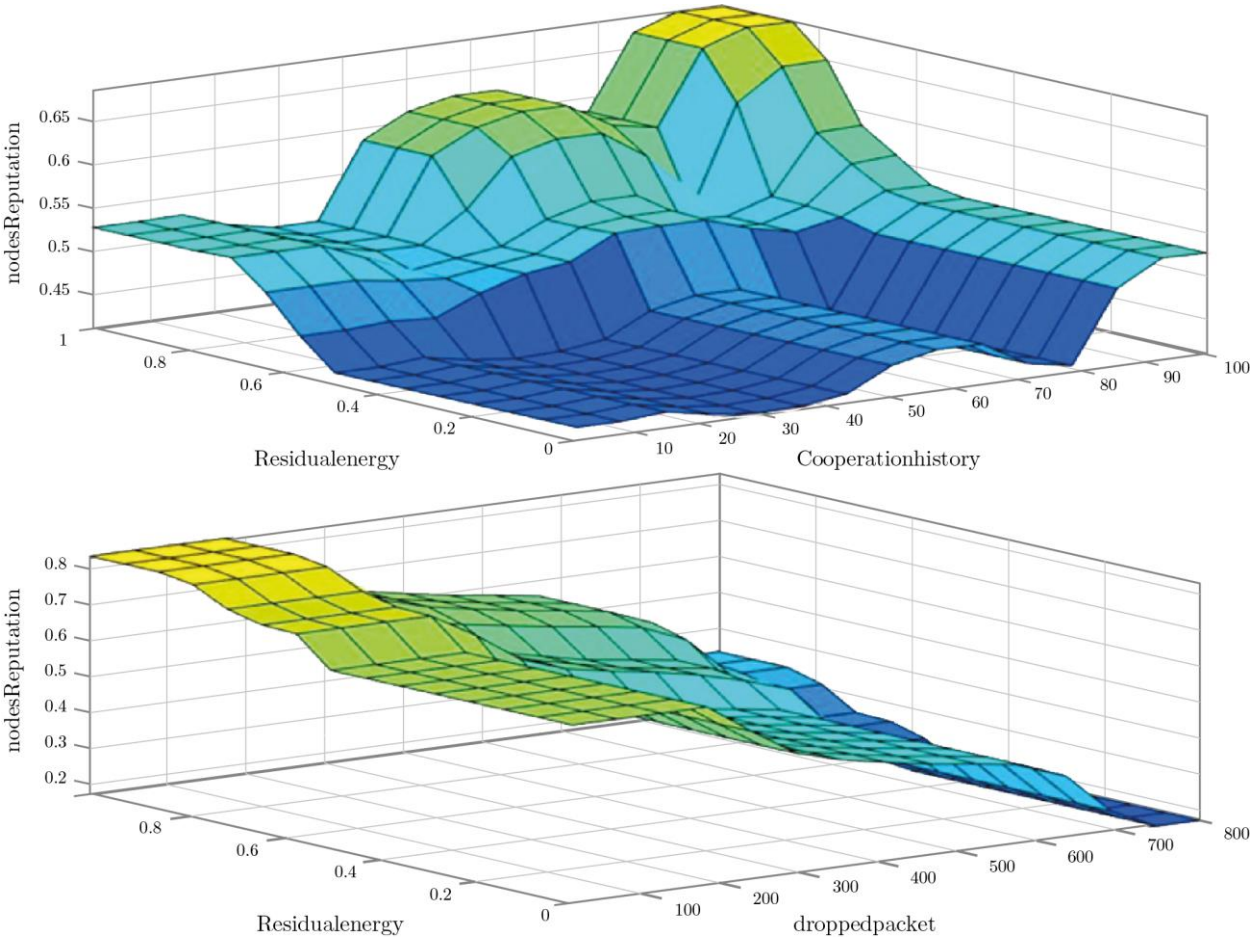

Figure 13. Fuzzy system of different inputs relation

The fuzzy inference method in MATLAB is used to determine the level of cooperation of network nodes and so-called MAMDANI (max-min) method is applied. The fuzzy rules are represented in the optimal performance of the system proportional to the inputs and output of the system in Figure 12. The output, which is node collaboration level for different inputs, for example, remained energy level, removed packets, and cooperation node history (Figure 13).

3.2 simulate evaluation in proposed method to detect selfish node

We have examined the results of simulations that have been implemented in a system with an 8.1 operating system with Intel (R) Core (TM) i7 processor at 2.4 GHz and 16 GB of internal memory in the MATLAB 2018 software environment to assess the performance of the proposed algorithm. We examined the efficacy of the proposed solution and evaluated its performance. Accordingly, the parameters of non-cooperative node detection accuracy, positive, negative warning rate, network PDR and deliver the data packets and average end-to-end latency of Game theory-based [11], Acknowledgment-based [10], TEEM [4] algorithms successfully have been compared with the proposed method. Table 7 shows each comparison metric.

Table 7. Comparison metrics to show evaluated of proposed method

metric	Definition
Detection Accuracy (DA)	$DR = \frac{T_p}{(T_p + F_N)}$
False Positive Rate (FPR)	$\text{FPR} = \frac{FP}{TP + FP}$
False Negative Rate (FNR)	$\text{FNR} = \frac{FN}{TP + FN}$
PDR	$PDR = \frac{R_i}{S_i}$
end-to-end delay	$\frac{sum\ of\ time\ taken\ packet\ to\ recieve\ destination}{number\ of\ recieved\ packet}$

- **Detection Accuracy (DA)**: detection accuracy in finding the selfish node determines the number of detected selfish nodes to the total number of selfish nodes in the network.
- **False Positive Rate (FPR)**: A rate of positive error represents the ratio of the number of normal nodes that have been falsely detected to the sum of the normal nodes number that have been falsely detected (FP) and the number of normal nodes that have been truly detected (TN) in the network.
- **False Negative Rate (FNR):** A rate of negative error represents the ratio of the number of selfish nodes that have been detected as normal node to the sum of the selfish nodes number that have been falsely detected as normal (FN) and the number of normal nodes that have been truly detected (TN) in the network.
- **PDR:** This parameter is basically the number of packets delivered successfully during the routing and transmission process from source to destination, so the packet delivery rate is the average number of packets delivered to the destination from all network nodes to the total number of packets generated in the network.
- **End-to-end Delay:** The average period of time when a data packet is routed from source to destination, expressed in units of time. The average delay is the retention time of data packets from sender to receiver as the scale of different networks. This time is actually the total time elapsed from the transmitter in various steps from the relay nodes to the receiver.

3.3. Packet delivery rate (PDR)

The delivery rate of successful data packets is one of the most important parameters of network evaluation in most working areas in the context of the networks of Internet of things. This parameter is basically the number of packets delivered successfully during the routing and transmission process from source to destination, so the packet delivery rate is the average number of packets delivered to the destination from all network nodes to the total number of packets generated in the network.

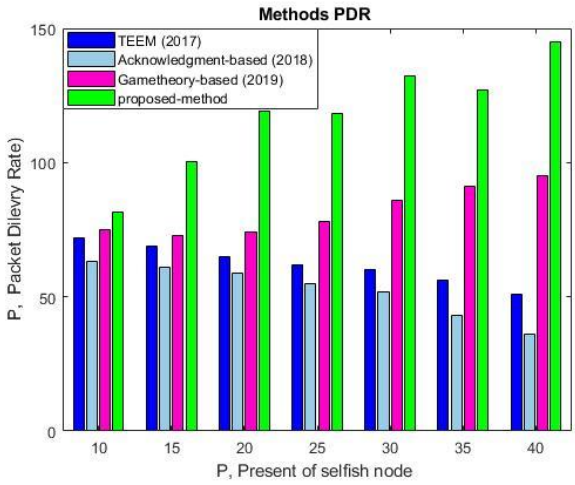

Figure 14: Comparison of PDR metrics

Table 7, PDR metric shows that R_i is the total number of packets received in the destination nodes and S_i is the total number of packets transmitted at the origin nodes. This metric has a significant impact on network performance. The higher the number of packets received at the destination, the lower delay on the number of packets arriving at the destination, the network delivery rate increases. For when the selfish nodes do not send the data packets in the network and the confirmation message is not received at the source, the node is again sent back in the network, increasing the traffic and the total number of packets produced and sent in the network, which has been ineffective. Therefore, the higher the packet delivery rate in the network, the more efficient use of the network resources include the bandwidth or limited energy resources of the nodes. In Figure 14, the high performance of the proposed approach is found to be due to early detection of selfish nodes in the network, which avoid repetitive data packet production and increase network traffic and delay in nodes.

3.4. End-to-end Delay

End-to-end delay of a packet means a time when the packet arrives at a destination from the source. The mean end-to-end latency is equal to the average time needed for packets to arrive at a destination from the source on the network. Figure 15 shows that there is a lower mean latency in the proposed method compared to other methods, particularly in the high percentages of selfish nodes in the network that it can be due to the fact that the efficiency and PDR are higher in the proposed method compared to other methods, and there is less traffic in the network, hence, the mean delivery time for a packet in the network and its mean waiting time have been reduced in network nodes.

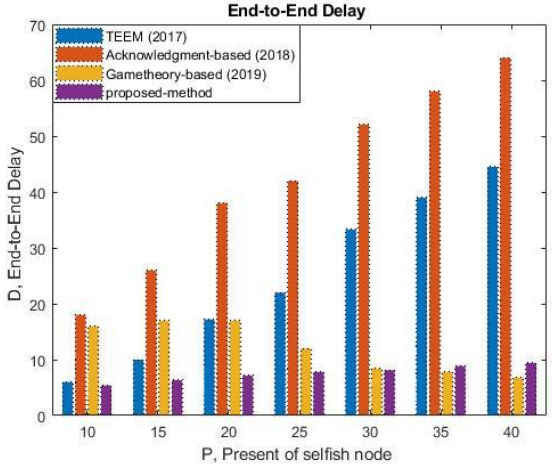

Figure 15. Comparison of Average End-to-End delay metrics

Some selfish nodes are also available in the network with malicious nature that increase the end-to-end delay in the network nodes so that the packet waits in the middle node buffer until the packet survives and is then sent which causes the packet to be dropped in another node due to completing the survival time of the packet, and this issue increases the mean end-to-end latency in network nodes. It is obvious that the shorter time needed to deliver the data packets on the network results in the optimal target on the network. The mean end-to-end latency in the network is one of the general parameters that its optimal objective is to reduce this amount to deliver the packets to the destination with less latency so that the network and the proposed approach can provide the service in that special application in emergency and real-time applications where time plays a critical role.

4. Conclusion

Due to the challenges of the Internet of Things, it is essential to explore non-cooperative nodes in order to increase network efficiency and PDR. In this paper, a multiphase mechanism based on game theory and direct and indirect reputation for stimulating non-cooperative nodes in the IOT has been introduced. The proposed method starts with setting up nodes and sending messages between objects to identify neighboring nodes. In the first phase, nodes in clusters are clustered with the cluster heads for data collection.

Then, they play a multi-person game between the source and destination nodes in the multi-person and data packet sending phase (cluster head) when they perform their own data packet or others' packet data forwarding. As the game runs, the node will learn implicitly about the status of neighboring nodes. If the neighboring node forwards the packets, the node will increase the possibility of selecting the neighboring node by increasing the node's direct reputation for playing in the next round to forward the data packet. Each node updates the reputation of those nodes through the performance of other neighboring nodes in the phase of discovering the direct and indirect reputation update of nodes and makes changes in their reputation table. The efficiency of suggested solution has been assessed and the parameters of selfish and malicious node detection rate, positive and negative warning rate, network PDR, and average end-to-end delay perform better compared to other previous methods in different categories.

Conflict of interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Ahmad, R., & Alsmadi, I. (2021). Machine learning approaches to IoT security: A systematic literature review. *Internet of Things*, 14, 100365.
2. Balakrishnan, K., Deng, J., & Varshney, V. K. (2005, March). TWOACK: preventing selfishness in mobile ad hoc networks. In *IEEE Wireless Communications and Networking Conference*, 2005 (Vol. 4, pp. 2137-2142). IEEE.
3. Charilas, D. E., & Panagopoulos, A. D. (2010). A survey on game theory applications in wireless networks. *Computer Networks*, *54*(18), 3421-3430.
4. Chong, Z. K., Tan, S. W., Goi, B. M., & Ng, B. C. K. (2013). Outwitting smart selfish nodes in wireless mesh networks. *International Journal of Communication Systems*, *26*(9), 1163-1175.
5. Das, D., Majumder, K., & Dasgupta, A. (2015). Selfish node detection and low cost data transmission in MANET using game theory. *Procedia Computer Science*, 54, 92-101.
6. Geetha, V., & Hariprasad, S. A. (2020). A conjectural based framework to detect & defend/classify selfish nodes and malicious nodes in manet using AODV. *International Journal of Innovations in Engineering and Technology (IJIET)*, 8-14.
7. Guo, J., Liu, H., Dong, J., & Yang, X. (2007). HEAD: a hybrid mechanism to enforce node cooperation in mobile ad hoc networks. *Tsinghua Science and Technology*, *12*(S1), 202-207.
8. Khanna, A., & Kaur, S. (2020). Internet of things (IoT), applications and challenges: a comprehensive review. *Wireless Personal Communications*, 114, 1687-1762.
9. Kumar, S., Tiwari, P., & Zymbler, M. (2019). Internet of Things is a revolutionary approach for future technology

enhancement: a review. *Journal of Big data*, 6(1), 1-21.
10. Liu, K., Deng, J., Varshney, P. K., & Balakrishnan, K. (2007). An acknowledgment-based approach for the detection of routing misbehavior in MANETs. *IEEE transactions on mobile computing*, 6(5), 536-550.
11. Nobahary, S., Garakani, H. G., Khademzadeh, A., & Rahmani, A. M. (2019). Selfish node detection based on hierarchical game theory in IoT. *EURASIP Journal on Wireless Communications and Networking*, *2019*(1), 1-19.
12. Rawat, P., Singh, K. D., & Bonnin, J. M. (2016). Cognitive radio for M2M and Internet of Things: A survey. *Computer Communications*, *94*, 1-29.
13. Sani, A. S., & Syeda, R. (2016). Defending selfish node in MANET using game theory approach. *Journal of Information Science and Engineering*, *32*(3), 559-573.
14. Yao, X., Wang, J., Shen, M., Kong, H., & Ning, H. (2019). An improved clustering algorithm and its application in IoT data analysis. *Computer Networks*, *159*, 63-72.

Abdi, G. H., Refahi Sheikhani, A. H., Kordrostami, S., & Babaie, S. (2023). A Novel Selfish Node Detection Based on Fuzzy System and Game Theory in Internet of Things. *Fuzzy Optimization and Modeling Journal*, *4*(4), 32-47.

https://doi.org/10.30495/fomj.2024.1996589.1122

Received: 29 September 2023 Revised: 24 December 2023 Accepted: 3 January 2024