Journal of Optimization of Soft Computing (JOSC)

Vol. 1, Issue 2, pp: (26-38), December-2023

Journal homepage: https://sanad.iau.ir/en/Journal/josc

Research paper

# FTRTA : Fault Tolerance and Reliable Transmissions Algorithm based on the Internet of Things

Mohsen Mozafari Vanani[1] and Pouya Khosravian Dehkordi [1*]

*1. Department of Computer Engineering, Faculty of Engineering, Islamic Azad University, Shahrekord Branch, Shahrekord, Iran.*

**Article Info**

**Article History:**
*Received:2024/2/9*
*Revised:*
*Accepted: 2024/4/4*

**Keywords:**
*Internet of Things, fault tolerance, routing, data exchange, protocol, energy.*

**Corresponding Author's Email Address: Khosravyan@gmail.com*

**Abstract**

The limitations of IoT have resulted in increased failures and the need for guaranteed fault tolerance to ensure adequate network performance. While previous studies have effectively improved fault tolerance by addressing various aspects of this area, previous methods are ineffective in ensuring the stability and accuracy of data exchange in the event of failure. The existence of this problem highlights the necessity of proposing a new method that can guarantee the stability and accuracy of data exchange to ensure network performance stability in case of failure. To address this, this research introduces a method called FTRTA, which is based on the enhancement of the RPL protocol and data distribution techniques. These distribution techniques are effective in improving load balancing and fault tolerance of network traffic. FTRTA is developed based on this technique and involves three operational steps. Firstly, it evaluates and analyzes the status of network nodes similar to when sending DIO messages. In the second step, it creates a network communication graph. Finally, in the third step, data transmission is performed using a distribution technique to ensure fault tolerance. Simulation results using Cooja software demonstrate the high performance of FTRTA in ensuring the stability and accuracy of data exchange, improving factors such as successful receptions and network throughput compared to similar studies.

## 1. Introduction

IoT has enabled all physical elements to communicate and interact with each other. In IoT, each member of the network has a separate digital identifier through which access, management, and communication with other members of the network are possible [1]. The IoT has become very important in today's world due to the wide range of benefits it has provided. One of the most basic applications of this technology in the current era is its use in medical, military, industrial, and general intelligence of physical elements [2]. This high progress makes it likely that many objects will join this network in the not-too-distant future [3]. However, it is noteworthy that this new network has extensive challenges and issues due to the incompleteness of technologies and standards [4]. One of the most important issues in these networks is the reliability and guarantee of tolerance. Due to the great importance of intelligent physical element management, IoT is very important in today's world, and the importance and applications of this new technology are increasing day by day. In IoT, due to severe resource limitations, wireless communications, network variable topology, and other related constraints, fault occurrence is very likely. Therefore, support for fault tolerance is a very important issue. However, the importance of this field is doubling due to the important areas of application of IoT [5]. Therefore, providing measures to improve reliability and support for fault tolerance is considered an undeniable necessity for IoT to ensure the continuity and accuracy of network performance [33]. IoT is a diverse network with many limitations. The existence of these unique features and limitations raises a number of issues related to this network. One of the most important of them is related to fault tolerance of data transmission. [32] On the other hand, the existence of these special and different features has made the use of traditional techniques (practical techniques in other wired and wireless networks) not suitable for these networks [6 and 7]. Therefore, given the importance and necessity of fault tolerance in maintaining the continuity and accuracy of IoT performance, extensive research has been provided to improve issues related to this area. Most of these studies have focused on improving routing and increasing the reliability of data exchanges. In IoT, nodes are unable to communicate directly with each other and root nodes due to limitations in equipment and their communications. For this reason, data exchanges are performed through other members of the network in multi-hop [8 and 9]. This performance makes network activity dependent on correct routing and reliable data exchange. Accordingly, a large section of the research in the field of fault tolerance has focused on this issue and has tried to increase the reliability and guarantee the accuracy of this vital category [10-17]. To this end, most research is based on the development of the RPL protocol [18] to improve the reliability of this routing protocol. It is worth noting that RPL is the most important IoT routing application protocol and is widely used in these networks. However, studies of past research have shown that there were some important issues associated with the protection of fault tolerance that make it necessary to provide more effective research in this area. In fact, most past research has focused on improving parent node choices and increasing the reliability of intermediate routes. However, improving parent choices and increasing the reliability of intermediate routes are very important issues, but neglecting other aspects of fault tolerance, especially fault coverage and ensuring the accuracy of exchanges, will lead to instability and loss of network performance. In fact, the safe choice of parents is a necessary condition, but it is not enough on its own.

In order to improve this issue, this article introduces a method called Fault Tolerance and Reliable Transaction Algorithm for IoT (FTRTA). FTRTA is based on the optimization of the RPL protocol and is based on the efficiency of data distribution techniques, and based on this, it tries to improve the reliability combined with the support of fault tolerance. FTRTA performance is generally divided into three main steps. In the first step, the reliability of the network nodes is assessed. In the second step, the DODAG graph is formed based on the proposed FTRTA measures. In the third step, data exchange is based on the data distribution technique [34].

The primary highlights of the Fault Tolerance and Reliable Transaction Algorithm for IoT (FTRTA) are the following:

- The main goal of FTRTA is to improve the tolerance fault of IoT network data exchanges.
- RPL optimization is used to improve routing and increase the reliability of intermediate routes.
- The efficiency of FTRTA is evaluated using COOJA simulator.

In the second section of this article the past works will be examined. Details of the proposed FTRTA will be presented in the third section. In the fourth section, the proposed method based on the Cooja software will be simulated and its performance will be evaluated. At the end, the article will be concluded.

## 2. Related Works

As noted, most reliability-based research has been developed based on the RPL protocol to focus on improving routing and data exchange. Some of these articles have focused on enhancing the RPL objective function and have attempted to increase the reliability of this protocol [10, 14-16, and 19-21]. Many studies have concentrated on energy-efficient routing and data exchange [23-26]. Others have been designed to evaluate link stability [11, 17, and 22] and have been developed based on assessing the condition from end to end of the intermediate routes [12 and 15]. However, methods that focus on improving the RPL objective function have shown relatively better performance in enhancing the reliability of data exchanges. Some of the most significant studies in this area are discussed below. Sennan et al. proposed a protocol called EDADA (Energy and Delay Aware Data Aggregation) [19]. This two-step method involves data compression techniques in the first step and evaluation of energy status and delay in the second step to select intermediate routes based on node assessment. While examining the energy situation has been effective in improving parent selections, focusing solely on this criterion may not meet all needs.

Sanmartin and colleagues introduced a method called SIGMA-ETX (SIGMA Expected Transmission Count) to enhance reliability [15]. This approach, based on RPL development and SIGMA evaluation, performs routing elections by analyzing the expected transmission count (ETX) and its variance for intermediate route nodes. This method has been effective in assessing the end-to-end state of intermediate routes, leading to improved reliability in data exchanges.

A method named MRHOF (Minimum Rank with Hysteresis Objective Function) was introduced by Lazarevska et al., based on RPL protocol development to enhance reliability [16]. In MRHOF, the network communication graph formation is based on evaluating the ETX index, node energy efficiency, and signal quality, followed by parent selections. Simulation results of this method demonstrate improved reliability in data exchanges. To enhance reliability, Sousa et al. introduced ERAOF (A New RPL Protocol Objective Function) [20]. This method optimizes the RPL protocol's objective function by evaluating ETX index, remaining energy metrics, and connection quality to make decisions. This approach has been effective in improving quality and increasing reliability in data exchanges.

The MIQRP protocol (Multiple Instances QoS Routing in RPL) introduced by Nassar et al. [10] aims to improve exchange reliability by introducing a new objective function called mOFQS. This function evaluates nodes based on ETX, energy, and delay, leading to appropriate parent selections. Simulation results indicate improved data interactions and reduced energy consumption.

In 2019, a method called EEMA (Energy Efficient and Mobility Aware) was introduced based on RPL protocol development to improve stability and reliability in communication, particularly for mobile networks. EEMA evaluates node mobility based on received signal quality and makes decisions by combining energy evaluations. This design has proven effective in enhancing reliability and stability of links, especially in mobile networks, though it lacks fault tolerance support. Vaziri et al. introduced a method called Brad-OF (Enhanced Energy-Aware Method for Parent Selection and Congestion Avoidance) to control congestion and enhance communication ability [21]. In Brad-OF, high-density node presence in the network communication graph is prevented, and nodes are evaluated based on ETX, remaining energy, and delay for parent selection. Simulation results show congestion control and improved reliability in data exchanges.

The MAPS protocol (Mobility-Aware Parent Selection for Routing) proposed in 2019 aims to improve network communication graph stability, especially in mobile networks. MAPS evaluates node mobility to select and form communication graphs based on signaling power for link quality and stability assessment. Parent elections are made based on stability and signaling quality predictions. Previous studies have attempted to improve routing reliability and data exchanges based on various measurements. However, they did not propose measurements to cover faults and guarantee the accuracy and continuity of data exchanges. This paper focuses on optimizing the RPL protocol and distribution techniques to address this fundamental issue. The studies are analyzed and discussed in Tables (1) in terms of evaluation metrics, purpose, simulation tools, and applied strategies.

**Table 1. The review of introduced articles focusing on the RPL OF**

| Reference | Evaluation metric | Simul<br>Ator | Goal |
|---|---|---|---|
| EDADA [19] | Energy, delay | Cooja | Energy optimazation |
| SIGMA [15] | ETX | Cooja | PDR Improvment |
| MRHOF [16] | Energy, ETX, RSSI | Cooja | QoS Improvment |
| ERAOF [20] | Energy, link quality, ETX | Cooja | QoS Improvment |
| MIQRP [10] | ETX, Energy, delay | Cooja | QoS Improvment |
| EEMA [17] | mobility | Cooja | PDR Improvment |
| Brad‑OF [21] | Energy, ETX, Delay | Matlab | Congestion control |
| MAPS [30] | mobility | Cooja | PDR Improvment |

## 3. Definitions

### 3.1. Communication Model

The desired network includes m nodes and a DODAG root. The nodes are randomly located in the networked environment. FTRTA follows the tree-based approach to exchanges. In this approach, the sensors, after collecting data, send it to the root node of the tree structure in response to the proposed measures. Data is distributed and sent to parents for the purpose of maintaining fault tolerance. How to distribute and send is determined by the fault tolerance and FTRTA decision-making. Finally, the root node receives the sent data and recovers it based on the applied technique, even if an error occurs. FTRTA is based on the RPL protocol, so the FTRTA communication model is consistent with this routing protocol [18]. RPL supports three different types of communication: point-to-point communication (P2P) for the connection between two nodes in the DODAG graph, point-to-multipoint communication (P2MP) to send traffic from the root node to the pages, and multipoint-to-point communication (MP2P) to collect and send data from network nodes to the DODAG root.

## 3.2. Energy Consumption Model

In networks with energy limitations, in order to design a routing technique, it is necessary to determine the network energy consumption model. Sending and receiving data on the IoT network are associated with the energy consumption of the nodes. The energy consumption for this purpose is determined by a function of the distance between the sender and receiver. In addition, the process of listening to the media to receive data and the sleep state of the nodes is accompanied by energy consumption. In the standard of IEEE 802.15.4 WSNs, energy consumption of sensor (a) on the link of $e\,(a,b) \in E$ for processing a message is evaluated by equation (1) [27].

Where $E_c^a$ is Energy Consume of node a, $E_l^a$, $E_t^a$, $E_r^a$ and $E_s^i$ are the energy consumed during the periods of listening, transmitting, receiving and sleeping, respectively. $I_t$, $I_r$, $I_l$ and $I_s$ are the current drawn in the transmitting receiving, listening and sleeping modes, respectively. $t_s^a$, $t_l^a$ are the current drawn in the transmitting, receiving, listening and sleeping modes, respectively. V is the battery voltage of the nodes, L (bits) is the packet length and BR (Kbps) is the data rate in the WSN.

In networks with energy limitations, in order to design a routing technique, it is necessary to determine the network energy consumption model. Sending and receiving data on the IoT network are associated with the energy consumption of the nodes. The energy consumption for this purpose is determined by a function of the distance between the sender and receiver. In addition, the process of listening to the media to receive data and the sleep state of the nodes is accompanied by energy consumption. In the standard of IEEE 802.15.4 WSNs, energy consumption of sensor (a) on the link of $e\,(a,b) \in E$ for processing a message is evaluated by equation (1) [27].

Where $E_c^a$ is Energy Consume of node a, $E_l^a$, $E_t^a$, $E_r^a$ and $E_s^i$ are the energy consumed during the periods of listening, transmitting, receiving and sleeping, respectively. $I_t$, $I_r$, $I_l$ and $I_s$ are the current drawn in the transmitting receiving, listening and sleeping modes, respectively. $t_s^a$, $t_l^a$ are the current drawn in the transmitting, receiving, listening and sleeping modes, respectively. V is the battery voltage of the nodes, L (bits) is the packet length and BR (Kbps) is the data rate in the WSN.

$$E_c^a = E_l^a + E_t^a + E_r^a + E_s^a = \left(t_l^a I_l + (I_t + I_r)\frac{L}{BR} + t_s^a I_s\right) V \qquad (1)$$

Based on reference [28], values of $t_s^a$ and $t_l^a$ are based on equation (2) and (3):

$$t_s^a = BI - SD = aBaseSD \times (2^{BO} - 2^{SO})\text{symbols} \qquad (2)$$

$$t_l^a = BI - (t_t^a + t_r^a + t_s^a) \qquad (3)$$

We assume that $E_r^a = 0$ (or $t_r^a = 0$) if a is a source node (Tx) and $E_t^a = 0$ (or $t_t^a = 0$) if a is a destination node (Rx).

Therefore, the consumed energy to send and receive of data is evaluated based on the equations (4) and (5).

$$E_c^a = t_l^a I_l + I_t \frac{L}{BR} + t_s^a I_s \qquad (4)$$

$$E_c^a = t_l^a I_l + I_r \frac{L}{BR} + t_s^a I_s \qquad (5)$$

## 3.3. The Network Assumptions

- Network nodes are homogeneous and fixed.
- Nodes are randomly located on the network.
- The network topology is various.
- Nodes don't have GPS equipment.
- Nodes have a unique identifier the energy of them is limited.
- It is assumed that fault occurs for 10 percent of sent data.

## 4. Fault Tolerance and Reliable Transaction Algorithm for IoT (FTRTA)

FTRTA is designed based on optimizing RPL protocol and distribution technique. Its goal is improvement of reliability and data fault tolerance. FTRTA to implement is compatible with IoT networks and RPL protocol. FTRTA is segmented to three steps:

1. Analysis of node status along with the process of sending DIOs.
2. Create a DODAG graph based on the proposed FTRTA measures.
3. Data transmission according to data distribution technique.

In the following is described detailed description of each step with the relevant flowchart.

## 4.1. Network Nodes Status Analysis

This step is based on the developed ROL protocol DIO messages, and its most important purpose is to assess the reliability and position of nodes in the network. Figure (1) presents this step of the FTRTA and describes its details below.

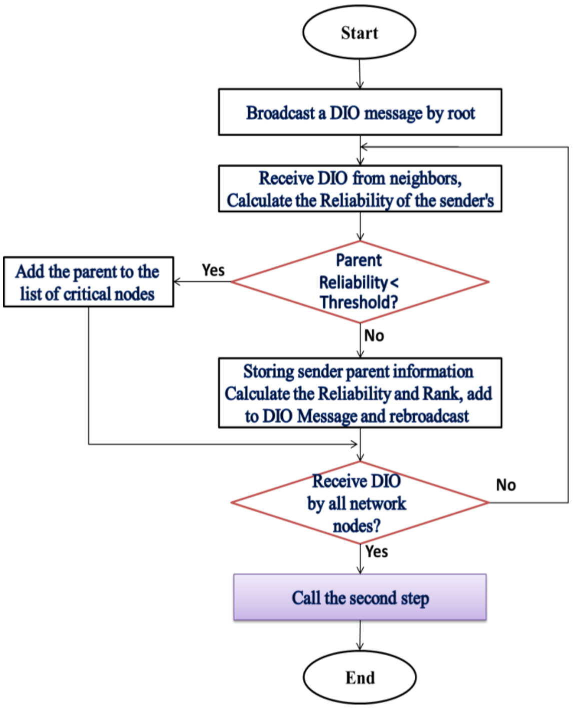
Start
Broadcast a DIO message by root
Receive DIO from neighbors, Calculate the Reliability of the sender's
Parent Reliability< Threshold?
Yes
Add the parent to the list of critical nodes
No
Storing sender parent information Calculate the Reliability and Rank, add to DIO Message and rebroadcast
Receive DIO by all network nodes?
No
Yes
Call the second step
End

**Figure 1. The first step of the proposed FTRTA**

According to the flowchart in Figure (1), when the network starts activity, the root node (based on sending DOI message of RPL protocol [18]) creates a DIO message and sends it to the network in the form of a broadcast. The purpose of distributing this message in the RPL protocol is to consider the node status from the root. In FTRTA, this goal includes considering reliability in addition to the node status. The sending of DIO process in FTRTA is based on RPL. Therefore, within a specified period of time, it is repeated to update the network topology. In the RPL protocol, after sending the DOI message from the root, it is shared among nodes so that all members of the network receive it. Sending DIOs in FTRTA has three differences compared to in the RPL protocol:
1. The first difference: In addition to performing operations related to the process of sending DIOs in accordance with the RPL, nodes also assess their reliability and add the result of this evaluation to the DIO message.
2. The second difference: If the DIO message is received from a parent whose reliability is less than the threshold, the parent will be added to the list of critical nodes and will be removed from the list of parent collections. This is intended to prevent the network graph from being constructed by low-reliability nodes. In FTRTA, the value of the confidence threshold is 0.1.
3. The third difference: The rank in the RPL protocol is evaluated according to the position of the nodes relative to the root. In FTRTA, however, in addition to location, node reliability is also influential in ranking. Accordingly, the ranking of nodes, in addition to location, also depends on the reliability of nodes. Figure (2) provides an overview of the DIO message sent to FTRTA. The details of the sent DIO message fields in FTRTA are as follows:

- IoT Graph Information-Base on DIO: This field corresponds exactly to the content of the DIO message of the RPL protocol and includes DADOG graph information, including root ID, graph ID, version, and other related items. Sufficient details are provided in [18].
- Rank: This field contains the sender's node rank. Rank refers to the position of the node relative to the root. If the rank of a node is lower, the node is closer to the root. Rank in RPL is rated for node distance from the root, but in FTRTA, in addition to node position, its level of reliability is also involved in ranking rating.
- RL: This field contains the reliability of nodes and is calculated according to the three concepts of energy, fault rate, and probability of data loss. Then it is added to the DIO.

| | | |
|---|---|---|
| IoT Graph Information_Base on DIO | Rank | RL |

**Figure 2. DAO message in FTRTA**

According to the flowchart presented in Figure (1), after receiving DIO messages by network nodes, first the reliability of the sender of the message is checked and if it is less than the threshold, the parent is removed from the list of total parents. . Otherwise, the receiver node stores the parent's information and puts it in the parent collection list. In equation (6), the details of detection and eliminating critical parents are presented. RLi is equivalent to i node reliability (reliability value is between zero and one, so the smaller the value, the lower the node reliability) and DF is the critical node detection flag.

$$DF = \begin{bmatrix} 0 & \text{IF} & (\text{RL}_i > 0.1) \\ 1 & \text{Else} & \end{bmatrix} \qquad (6)$$

After performing the process of identifying critical parents, the receiver node evaluates the reliability based on the equation (7) and the rank according to the equation (8) and adds the result of these calculations to the DIO message. The message is then resent on the network in the form of broadcast. In equation (7) RLi is equivalent to node i reliability, Eri is remain energy of node i, EInit is equivalent to i node energy at the first instant, ERi is the number of fault occurring during i node previous interactions, $\sum \text{No. of success Sending}$ is the total number of messages that k node has been successful in sending them. No.of all Packet Received is equivalent to the total number of messages received by k node, and α is the various valuation coefficient according to the reliability evaluation criteria and has a value between zero and one.

$$\mathrm{RL_i} = \alpha.\left(\frac{\mathrm{Er_i}}{\mathrm{E_{Init}}} \times \frac{1}{\mathrm{Log(ER_i)}}\right) + (1 - \alpha).\left(\left[\frac{\sum \text{No.of Successl Sending}}{\text{No.of all Packet Received}}\right]\right) \qquad (7)$$

In equation (8) Rki is equivalent to the rank of a node, RkF is the rank of parent node i, MinRkInc is the constant rate of increase in children's rank, and ω is an influential indicator of reliability in node rank evaluation and has a value between zero and one. The larger the value of it, the greater the effect of reliability on node rank evaluation.

$$\mathrm{Rk_i} = \mathrm{Rk_F} + \mathrm{MinRk_{Inc}} + \omega \times \left(\frac{1}{\mathrm{RL_i}}\right) \qquad (8)$$

The process provided in connection with sending DIOs is repeated until all the nodes of the network finally receive the DIO message. Based on the performance of this step, unreliable parents are identified, nodes are informed of their location and network topology, and nodes are informed of their parents' reliability. After sending the DIOs, the second step of FTRTA is executed to form the DODAG graph.

## 4.2. Creating a DADOG Graph

The purpose of this step is to select parents and create an improved DADOG graph based on the proposed FTRTA measures. Figure 3 of presents this step of the proposed method.

This step of the FTRTA is in accordance with the sending process of DAO in the RPL protocol. The difference is that the objective function and the parents are evaluated and selected according to the FTRTA measures.

According to the flowchart presented in Figure (3) after completing the sending of DIOs, the nodes first check the number of parents after creating the DAO message (in accordance with the basic RPL protocol). Based on the results of this study, two different cases are presented as follows.

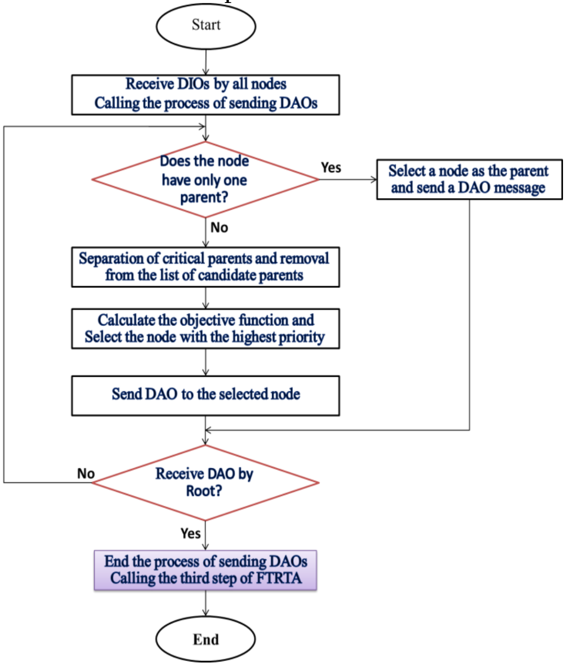
Start
Receive DIOs by all nodes
Calling the process of sending DAOs
Does the node have only one parent?
Yes
Select a node as the parent and send a DAO message
No
Separation of critical parents and removal from the list of candidate parents
Calculate the objective function and Select the node with the highest priority
Send DAO to the selected node
No
Receive DAO by Root?
Yes
End the process of sending DAOs
Calling the third step of FTRTA
End

**Figure 3. Flowchart of the second step of the proposed FTRTA**

1) The node has only one parent. In this case, the parent is selected as the original parent and the DAO message is sent to them.

2) The node has more than one parent. In this case, first the critical parents are separated from the other parents and then the best parent is selected as the main parent from the remaining parents. After selecting the parent, the DAO is sent to the selected parent.

The main parent in the RPL is selected based on the objective function. The RPL objective function is evaluated based on the indicator called ETX, and the main parent is selected based on the results of this evaluation. In the proposed FTRTA, the objective function has been improved, and in addition to the ETX index, node rank and reliability also play important roles in the evaluation and selection of the main parent. This is intended to improve the reliability of the DODAG graph in FTRTA. Equation (9) provides details of the objective function assessment in FTRTA. In this regard, $\mathrm{OF_i}$ is equivalent to the objective function evaluated for parent I, $\mathrm{RL_i}$ is equivalent to parental value k in terms of capability, $\mathrm{ETX_i}$ is equivalent to the expected transfer rate of node i and $\mathrm{Rk_i}$ is equal to parent value of i. $w_1$, $w_2$ and $w_3$ are equivalent to variable valuation coefficients to the evaluation criteria of the proposed objective function. These coefficients have value between 0 and 1 that sum of them is 1.

$$\mathrm{OF_i} = (\mathrm{w_1} \times \mathrm{RL_i}) + (\mathrm{w_2} \times \mathrm{ETX_i}) + \left(\mathrm{w_3}, \frac{1}{\mathrm{Rk_i}}\right) \qquad (9)$$

ETX is a factor for elections of parents in RPL protocol that is evaluated based on successful possibility for sending packet and receiving ACK. Details of considering this factor is in equation (10). So, $\mathrm{df_i}$ is the measured probability that a packet is received by the neighbor and $\mathrm{dr_i}$ is the calculated probability that the acknowledgment packet is received successfully.

$$\mathrm{ETX_i} = \frac{1}{\mathrm{df_i} \times \mathrm{dr_i}} \qquad (10)$$

Based on the result of the equation (9), each child chooses the best parent as the main parent and sends the DAO message to him. By receiving the DIO message, the parent stores the child's information and this process is repeated until the DAO message is finally received by the root. By receiving DAO messages from the root, the main parents are determined and network graph are formed. After this step, the final step of the FTRTA is for data transmission.

### 4.3. Data Transmission Based On Data Distribution Technique

After forming the DODAG graph, whenever the node intends to send data, it performs the sending process based on the performance of this process. The main purpose of this step is to ensure data interaction fault tolerable. Figure 4 shows the flowchart of this step and then analyzes of its performance details have been prepared.

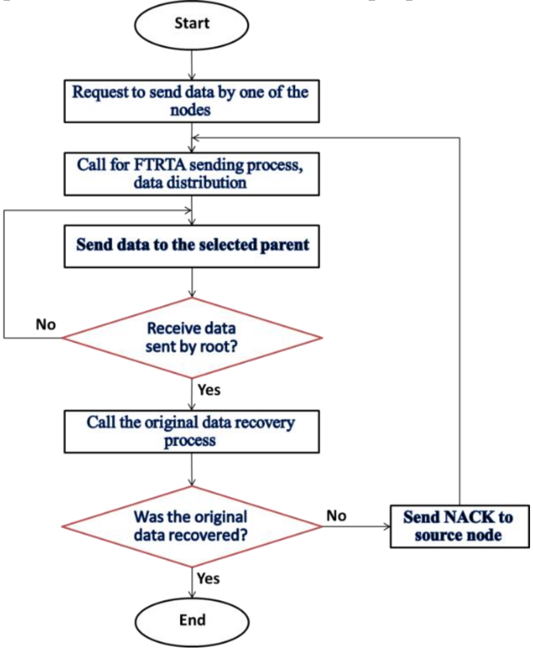
Start
Request to send data by one of the nodes
Call for FTRTA sending process, data distribution
Send data to the selected parent
Receive data sent by root?
No
Yes
Call the original data recovery process
Was the original data recovered?
No
Send NACK to source node
Yes
End

**Figure 4. The flowchart of the third step of the proposed FTRTA**

The data transmission in FTRTA is based on data distribution techniques, aiming to ensure fault tolerance even in the event of errors and data loss. In this technique, the sender node in a distribution process divides the data into several sections (n sections) and sends them to the root through the parent nodes and the network communication graph. The root node can recover the main data by receiving a certain number of these sections ($k < n$). During the sending process, if one or more sections are corrupted or lost, the root node can retrieve and recreate basic information by receiving other sections, ensuring fault tolerance.

Additionally, the distribution technique can be adaptably adjusted to meet fault tolerance requirements, making it a prominent benefit of this technique. Data distribution and retrieval processes can be defined so that receiving the minimum sent sections still allows for the retrieval of basic information. It is important to note that the greater the need for fault tolerance, the higher the resource consumption, as the amount of network consumption resources is related to the required level of fault tolerance.

To model the proposed distribution technique, a multi-sentence linear equation is required. This equation is adjusted as a variable according to the fault tolerance requirements. For instance, if we want the root node to recover the original information by receiving 3 sections during the send processes, the number of linear equation components must be considered as 4 components. Similarly, if we want the root node to recover the original information by receiving two sections, then the number of components of the linear equation must be considered as 3 components. The source node with different values of these components can create any number of required sections for sending. Note that the value of the components is random, but prime numbers must be used.

The number of sections created for sending is not related to the ability to retrieve main information. The recovery process is independent of the number of sections created and is done with a certain number of sections. The basis of the data recovery process at the destination is determined by the number of linear equation components. For example, if the linear equation has 3 components, the root node can recover the main information by receiving three sections. However, if it receives less than 3 sections in this scenario, the main information will not be recoverable.

Algorithm (1) provides details of the FTRTA data distribution process, intended for a scenario in which the information is divided into three sections, and the root node will be able to recover the main information by receiving the second section.

**Algorithm 1. Send data based on FTRTA**

**data distribution process**

```
Every node Request to send data  {
Scenario for (3,2);
//  Divide the data into 3 sections and retrieve the original data based on the 2 sections
For This Scenario → $\mathbf{F(x) = c_1, x + c_0 \text{ Mod } u}$
// $\mathbf{c_1}$ and $\mathbf{c_0}$ are randomly selected (for example Respectively 2 and 4)
//$\mathbf{u}$ A Prime number is selected that is greater than all values (for example 7)
Assign the Prime numbers to the variable c to create sections;
$\mathbf{F(2) = (4 + 4 \text{ Mod } 7) \rightarrow F(2) = 1}$;        $\mathbf{F(3) = (6 + 4 \text{ Mod } 7) \rightarrow F(3) = 3}$;
$\mathbf{F(5) = (10 + 4 \text{ Mod } 7) \rightarrow F(5) = 0}$;
After create sections, send sections for Root by Selected father;
After recevied sections by Root;
original data recovery Process Summon based on Lagrange Equ;
For example two sections $\mathbf{F(3)}$ and $\mathbf{F(2)}$ Recevied, and $\mathbf{F(5)}$ Lost;
Recovery by two sections;
$\mathbf{\begin{matrix} F(3) = 3 \\ F(2) = 1 \end{matrix} \rightarrow F(x) = c_1, x + c_0 \text{ Mod } u \rightarrow \begin{bmatrix} 3c_1 + c_0 = 3 \\ 2c_1 + c_0 = 1 \end{bmatrix} \rightarrow}$
$\mathbf{c_1 = 2 \rightarrow \begin{matrix} (3 * 2 + c_0) \text{ Mod } 7 = 3 \\ (2 * 2 + c_0) \text{ Mod } 7 = 1 \end{matrix} \rightarrow c_0 = 4}$
```

Based on what has been provided, data exchange is performed and fault tolerance is ensured during the send process.

## 5. Performance Evaluation

### 5.1. Simulation Setup

To implement and evaluate FTRTA performance, this method has been simulated with the Cooja software [29] and compared with the MIQRP [10] and RPL [18] method]. For the experiments, we used the Contiki IPv6 / 6loWPAN model and the RPL text protocol called ContikiRPL [30]. The configuration parameters of the simulation scenarios are presented in Table (2).

**Table 2. Simulation parameters.**

| Parameter | Value |
|---|---|
| Operating system | Contiki master version (2.7) |
| Loss Model | Distance loss |
| Sensors | Skymote |
| Adaptation | 6LoWPAN |
| Communication protocol | CSMA, RDC contikimac, IEEE 802.15.4, ContikiRPL, IPv6 |
| OF | OFQS, FTRTA (Proposed OF), ETX |
| The number of sensors | 15, 30, 45, 60, 75, 90 |
| Network area | 500M*500M |
| Microcontroller unit | ARM Cortex M3, 32-bits, 72 MHz, 64 kB RAM |
| Data packet size | 30 bytes |
| DIO, DAO and DIS size | 16, 16, 4 byte |
| Band width | 250 Kbps |
| Transmission layer | UDP |
| Initial energy of sensors | 1500 mA |
| Radio model | Unit Disk Graph Medium (UDGM) |
| Data errors during sending | 10% of all data |
| Simulation time | 600 s |
| Result | Avg 20 round |

The simulations used Contiki Power trace to investigate the energy consumption of nodes. Power trace output is the total energy consumption of the nodes while they are active [31]. Table (3) shows the amount of energy consumption when the nodes are active.

**Table 3. The power consumption when mote sky is active**

| Mode | Energy Consume |
|---|---|
| The MCU is active and the radio unit is in reception mode | 2.18 mA |
| MCU is active and the radio unit is in sending mode | 19.5 mA |
| MCU is active and radio unit is inactive | µA 1800 |
| The MCU is idle and the radio unit is inactive | µA 54.5 |
| MCU ready to operation | µA 5.1 |

The energy in the 6LowPAN network is checked by the duty cycle in which the radio units are only active at the time of sending and receiving. Contiki MAC is used for this purpose. Contiki MAC is an MAC standard in which the radio unit is 6% active when there is no traffic. Based on the presented topics, the equation (11) presents energy consumption based on time.

$$\text{Energy Consume}_i(\text{mj}) = \text{Send} \times 19.5\text{mA} + \text{Received} \times 21.8\text{mA} + \text{Cpu} \times 1.8\text{mA} + \text{Lpm} \times 54.5\text{mA} \qquad (11)$$

### 5.2. Result

This section details the simulation results. In the experiments, the number of nodes for different scenarios varies between 15 and 90, and the result of each scenario is examined and displayed with an average of 20 cycles. In order to evaluate the performance of the methods, the criteria of percentage of data loss, energy consumption, delay and transit have been used. Details of these criteria are provided below.

**Data loss rate**: This criterion is defined in relation to the lost data rate in relation to all sent data and is evaluated according to Equation (12).

$$\text{PDR} = \frac{\sum \text{No.of Packet Drop}}{\sum \text{No.of Packet Send}} \qquad (12)$$

**Network delay:** This criterion depends on the average time required to receive the data by the root node and is evaluated based on the equation (13).

$$\text{Delay} = \frac{\sum_{j=1}^{\text{No,of send data}} \text{Arrival Time}_j\text{-Send Time}_j}{\text{No,of send data}} \qquad (13)$$

**Energy consumption**: This factor is related to the total energy consumption of network sensors and is evaluated based on the equation (14).

$$\text{Energy Consume} = \sum_{i=1}^{n} \text{Energy Consume}_i(\text{mj}) \qquad (14)$$

**Network throughput**: This factor is related to the network throughput (network actual interaction rate) and is evaluated based on equation (15).

$$\text{Network Throughput} = \frac{\sum \text{No of byte Recceive}\times 8}{\text{Time (s)}}\,\text{bps}$$

(15)

### 5.2.1. The effect of network density

In this section, the effect of node density on the compared methods is investigated. Therefore, the number of nodes in different scenarios is considered to be between 15 and 90 nodes, randomly placed in the network. The traffic rate in the network is set at 40 PPS (packets per second) for different scenarios.

Data loss rate: Figure (5) illustrates the effects of node density variation on the percentage of network data loss. It is observed that in all three methods, as the number of nodes increases, the percentage of data loss also increases. This increase can primarily be attributed to the rise in network traffic rate and its interruptions. Furthermore, with an increasing number of nodes, the length of intermediate routes (the number of hops in the intermediate routes) also increases, leading to a higher likelihood of data loss. FTRTA outperforms MIQRP and RPL in terms of establishing reliable routing based on parental choices and ensuring fault tolerance. Additionally, FTRTA offers the capability to identify unreliable nodes, further reducing the risk of data loss. Although MIQRP excels in selecting parents and creating secure routes, it lacks the ability to tolerate faults and mitigate the negative impacts of uncertain nodes. On the other hand, RPL, as a fundamental method, does not incorporate measures to enhance reliability and fault tolerance, resulting in inferior performance compared to the other two methods.

When the network consists of 15 nodes, the data loss percentage for FTRTA is approximately 10%, surpassing MIQRP and RPL by 3.5% and 6%, respectively. However, with 90 nodes present, the data loss rate for FTRTA reaches 40%, marking a success rate 9.5% and 15% higher than MIQRP and RPL, respectively. The study indicates that FTRTA demonstrates greater success in scenarios with an increasing number of nodes.

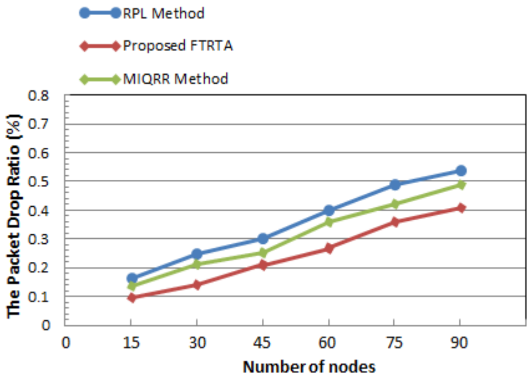

**Figure 5. Packet drop ratio by changing the number of nodes**

Network energy consumption: Figure (6) illustrates the effects of varying node presence on network energy consumption. In all three methods, the level of network energy consumption has increased with the rising number of nodes. However, the consumption rate in FTRTA is higher than that of the MIQRP and RPL methods, and it escalates as the number of nodes increases. This increase is attributed to the rise in network traffic rates facilitated by FTRTA, consequently leading to an increased network energy consumption, aligned with its measures aimed at ensuring fault tolerance. This may present a limitation in terms of guaranteeing fault tolerance. Comparatively, MIQRP outperforms RPL in this aspect. MIQRP has been more successful in reducing energy consumption compared to RPL, achieved by considering the energy status of nodes during parental selections and optimizing delay-based exchanges.

By analyzing the energy status and making appropriate decisions based on requirements, both the concept of optimization and the amount of consumption are well supported. EDADA focuses on examining the energy levels of parent nodes but does not provide the capability to manage energy. RPL has not implemented any measures in this area and has resulted in increased energy consumption compared to the other two methods.

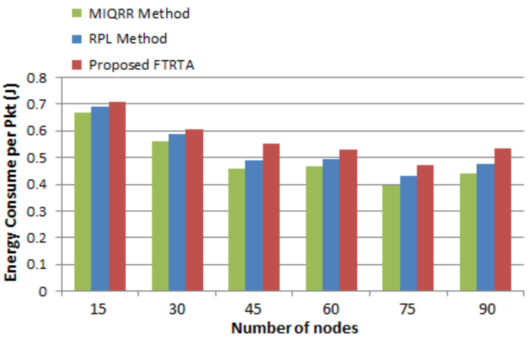

**Figure 6. Network energy consumption by changing the number of nodes**

**Network Delay**: Figure (7) shows the effect of node density on delay. FTRTA optimizes intermediate routs during the formation of graph according to rank in elections of parents. It also guarantees communication continuity and fault tolerance during transmission of data based on the distribution technique. On this basis, in addition to optimizing intermediate routs, service connectivity is guaranteed in different situations, which has resulted in improved end-to-end delay. MIQRR has been successful in improving parental choices and optimizing delay based on improving the objective function-based delay index. However, this method does not take proper measures to maintain the continuity of service. The ETX-based RPL objective function is evaluated and the parents are selected based on this. Therefore, this protocol has a higher delay than the other two methods.

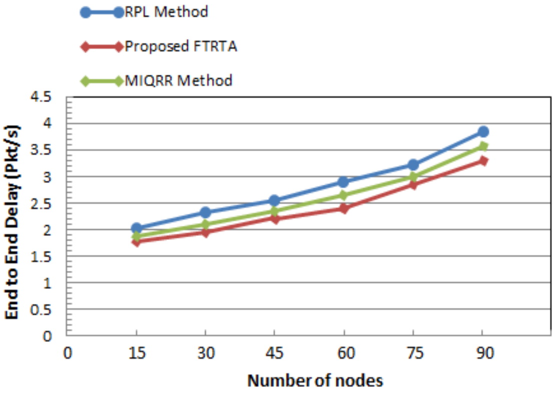

**Figure 7. End to end delay by changing the number of nodes**

**Network throughput**: Figure (8) shows the effects of the presence of variable nodes on the network throughput. FTRTA, based on its proposed steps, provides the ability to detect critical nodes, increases the reliability of the network communication graph as much as possible, and ensures fault tolerance of exchanges. The result of this three-step design is a total improvement in exchanges and an increase in network throughput. MIQRP is effective in graph reliability, but this method does not cover other aspects, especially fault tolerance during exchanges. RPL has not any measures to support reliability and fault tolerance, and due to issues caused by this inefficiency, it has been associated with a further decline in network throughput compared to the other two methods.

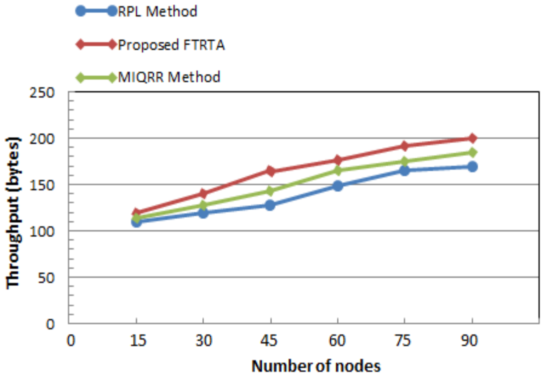

**Figure 8. Network throughput by changing the number of nodes**

### 5.2.2. The effect of traffic loads

In this section, the effect of traffic load on the methods being compared is investigated. So, the number of nodes in different scenarios is considered to be 50 nodes that are randomly placed in the network. The rate of traffic sent in the network for different scenarios is among 20-100 PPS for different scenarios.

- Packet Drop Ratio (PDR)

Figure (9) shows the effects of traffic load on PDR. With increasing traffic load, PDR has increased in all three methods. The reason for this is disruptions and problems caused by increased traffic and congestion. But this increase for FTRTA was lower compared to the other two methods. This is because of the advantages that FTRTA provides in terms of data distribution. Due to the measures of the distributed data exchanges for providing fault tolerance, FTRTA has caused the traffic in the intermediate routes is distributed and balanced. This distributed transmission is very effective in improving congestion issues, especially in high-traffic scenarios. This performance, along with other FTRTA measures to support transaction reliability, has led to improved successful delivery and reduced PDR for the proposed method. This is while neither of the two methods provides the ability to balance traffic. This inefficiency has increased PDR, especially in high-traffic scenarios. However, the increase in PDR for RPL is more severe than for MIQRP. RPL is primarily designed for low traffic networks and is inefficient in high traffic scenarios. When the traffic sent through the network was 20 PPS, PDR for FTRTA was about 10% that it was 3.5% and 6% more successful than that of MIQRP and RPL, respectively. At a traffic rate of 100 PPS, the PDR for FTRTA was 40%, which was 9.5% and 15% more successful than for MIQRP and RPL, respectively. This study concludes that the effect of traffic distribution is greater in scenarios with higher traffic rates.

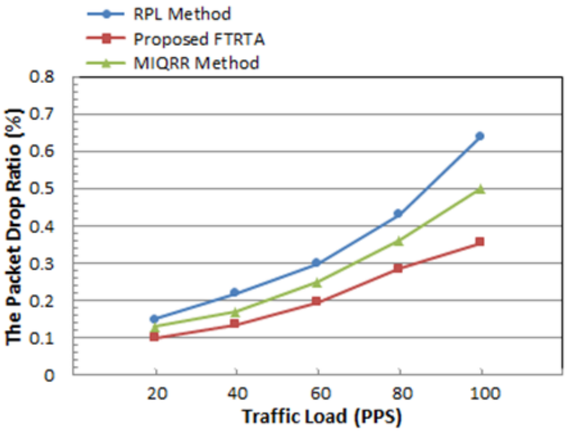

**Figure. 9 Packet drop ratio by changing the traffic load**

- Network energy consumption

Figure (10) shows the effects of traffic rates on the network energy consumption. With increasing traffic rates, energy consumption has also increased due to the direct effect of increasing data transmission on network energy consumption. The important point is that with increasing traffic load, energy consumption for FTRTA has been less than for the other two methods because of the effects of distributed data transmission on maintaining the balance of route traffic and reducing the negative effects of increased congestion, especially energy consumption. Data distributed transmission in FTRTA balances traffic on the network communication routes. This has been effective in reducing buffer overflow, data loss, resend and other congestion issues, which are sever in heavier traffic scenarios. These issues increase energy consumption, which FTRTA effectively prevents from occurring these. MIQRP has been successful in reducing energy consumption by considering energy in parent selection and graph formation, but the lack of capacity to balance traffic has exacerbated congestion issues that its effect on increasing the energy has been more severe, especially in scenarios with more traffic.

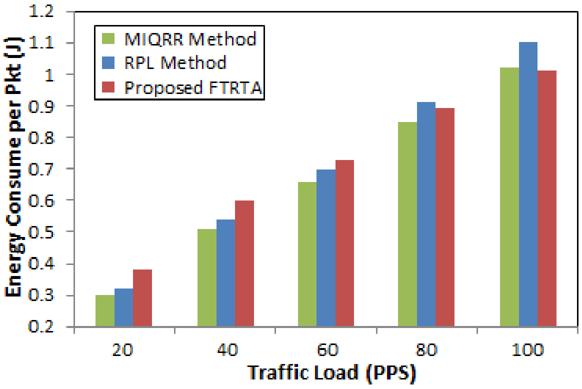

**Figure. 10 Netwotk energy consumption by changing the traffic load**

- Delay

Figure (11) shows the effects of traffic rates on the network delay. Delay is directly related to increased traffic load. Accordingly, with increasing traffic, delay has also increased, which has intensified for scenarios with heavier traffic, because congestion has increased in these scenarios, which has a reciprocal effect on increasing delay. The RPL protocol is inefficient in terms of traffic management and congestion control, and in this regard, with increasing traffic load, delay in this protocol has increased much more. Although MIQRR has considered delay in its routing, this alone is not enough, especially for high-traffic scenarios. In addition to creating a secure graph and improving the reliability of data exchanges, FTRTA operates during exchanges in such a way that it maintains the traffic balance of the routes by sending distributed data. This has resulted in improved delay for MIQRR.

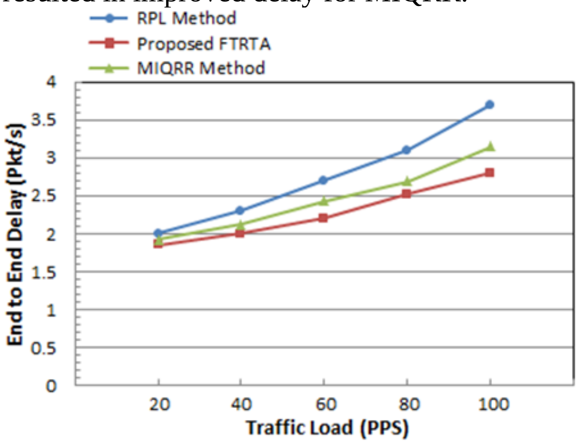

**Figure. 11 End to end delay by changing the traffic load**

- Network throughput

Figure (12) shows the effects of traffic load on network throughput. With the increase of traffic load up to 80 PPS, the throughput for FTRTA and MIQRP has an increasing trend, but after that it has had a decreasing trend. According to what was presented, FTRTA, in addition to forming a reliable graph and supporting data reliability, has also been successful in maintaining traffic balance and controlling congestion of intermediate routes. The result of this successful performance has been improved data exchange and increased network throughput for FTRTA. MIQRP has been successful in supporting the reliability of data exchanges, but this method has not managed traffic and congestion, which has led to decrease throughput, especially in scenarios with heavy traffic. RPL is a simple routing protocol and is primarily designed for use in low density networks.

This has led to a drop in throughput, especially in scenarios with heavy traffic.

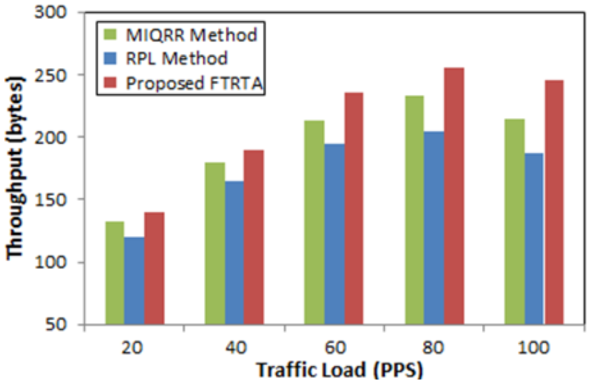

**Figure. 12 Network throughput by changing the traffic load**

## 6. Conclusion and future works

In this paper, a different method called FTRTA is introduced to enhance the reliability and fault tolerance of data exchange in IoT. FTRTA is based on optimizing the RPL protocol and utilizing data distribution techniques, aiming to improve the reliability of routing support with fault tolerance. The FTRTA is implemented to evaluate software based on the Cooja simulator software, and the results of its performance indicate an increase in successful network receipts, improved stability of intermediate routes, and increased network throughput compared to similar methods. However, while FTRTA has performed remarkably well in improving fault tolerance, it is ineffective in supporting the quality of routing and data exchange. Therefore, in future research, we will attempt to address this issue by evaluating qualitative criteria in addition to fault tolerance.

## References

[1] H. HaddadPajouh, A. Dehghantanha, R. M. Parizi, M. Aledhari, and H. Karimipour, A survey on internet of things security: Requirements, challenges, and solutions, *Internet of Things*, Vol. 14, pp. 100129, Jun, 2021.

[2] R. Hassan, F. Qamar, M. K. Hasan, A. H. M. Aman, and A. S. Ahmed, Internet of Things and its applications: A comprehensive survey, *Symmetry*, Vol. 12, no. 10, pp. 1674, Oct, 2020.

[3] Huang, Haiping, et al. An efficient signature scheme based on mobile edge computing in the NDN-IoT environment, *IEEE Transactions on Computational Social Systems*, Vol. 8, no. 5, pp. 1108-1120, May, 2021.

[4] M. Al-Emran, S. I. Malik, and M. N. Al-Kabi, A survey of internet of things (IoT) in education: opportunities and challenges, *Toward social internet of things (SIoT): Enabling technologies, architectures and applications*, pp. 197-209, 2020.

[5] L. Xing, Reliability in Internet of Things: Current status and future perspectives, *IEEE Internet of Things Journal*, Vol. 7, no. 8, pp. 6704-6721, May, 2020.

[6] K. Gulati, R. S. K. Boddu, D. Kapila, S. L. Bangare, N. Chandnani, and G. Saravanan, A review paper on wireless sensor network techniques in Internet of Things (IoT), *Materials Today: Proceedings*, 2021.

[7] B. Pourghebleh, V. Hayyolalam, and A. A. Anvigh, Service discovery in the Internet of Things: review of current trends and research challenges, *Wireless Networks*, Vol. 26, no. 7, pp. 5371-5391, May, 2020.

[8] Y. B. Zikria, M. K. Afzal, F. Ishmanov, S. W. Kim, and H. Yu, A survey on routing protocols supported by the Contiki Internet of things operating system, *Future Generation Computer Systems*, Vol. 82, pp. 200-219, May, 2018.

[9] A. J. Dey, and H. K. D. Sarma, Routing Techniques in Internet of Things: A Review, *Trends in Communication, Cloud, and Big Data*, PP. 41-50, 2020.

[10] J. Nassar, M. Berthomé, J. Dubrulle, N. Gouvy, N. Mitton, and B. Quoitin, Multiple instances QoS routing in RPL: Application to smart grids, *Sensors*, Vol. 18, no. 8, pp. 2472, Jul, 2018.

[11] A. A. Kadhim, and S. A. Rafea, Routing with Energy Threshold for WSN-IoT Based on RPL Protocol, *Iraqi J. Comput. Commun. Control Syst. Eng*, Vol. 19, no. 1, pp. 71-81, Mar, 2019.

[12] T. L. Jenschke, R. A. Koutsiamanis, G. Z. Papadopoulos, and N. Montavont, Multi-path selection in RPL based on replication and elimination, *International Conference on Ad-Hoc Networks and Wireless*, pp. 15-26, Sep, 2018.

[13] M. Conti, P. Kaliyar, and C. Lal. A robust multicast communication protocol for Low power and Lossy networks, *Journal of Network and Computer Applications*, pp. 102675, Aug, 2020.

[14] T. Muhammed, R. Mehmood, A. Albeshri, and A. Alzahrani, HCDSR: A hierarchical clustered fault tolerant routing technique for IoT-based smart societies, *Smart Infrastructure and Applications*, pp. 609-628, 2020.

[15] P. Sanmartin, A. Rojas, L. Fernandez, K. Avila, D. Jabba, and S. Valle, Sigma routing metric for RPL protocol, *Sensors*, Vol. 18, no. 4, pp. 1277, Apr, 2018.

[16] M. Lazarevska, R. Farahbakhsh, N. M. Shakya, and N. Crespi, Mobility Supported Energy Efficient Routing Protocol for IoT Based Healthcare Applications, *IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 1-5, Oct, 2018.

[17] M. Bouaziz, A. Rachedi, A. Belghith, M. Berbineau and S. Al-Ahmadi, EMA-RPL: Energy and mobility aware routing for the Internet of Mobile Things, *Future Generation Computer Systems*, Vol. 97, pp 247-258, Aug, 2019.

[18] T. Winter, et al. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, *rfc*, Vol. 6550, pp. 1-157, Mar, 2012.

[19] S. Sennan, S. Balasubramaniyam, A. K. Luhach, S. Ramasubbareddy, N. Chilamkurti, and Y. Nam, Energy and Delay Aware Data Aggregation in Routing Protocol for Internet of Things, *Sensors*, Vol. 19, no. 24, pp. 5486, Dec, 2019.

[20] N. Sousa, J. V. Sobral, J. J. Rodrigues, R. A. Rabêlo and P. Solic, ERAOF: A new RPL protocol objective function for Internet of Things applications, *2nd*

*International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, pp. 1-5, Jul, 2017.

[21] B. Vaziri, and A. T. Haghighat, Brad-OF: An Enhanced Energy-Aware Method for Parent Selection and Congestion Avoidance in RPL Protocol, *Wireless Personal Communications*, pp. 1-30, 2020.

[22] S. Hoghooghi, and R. N. Esfahani, Mobility-Aware Parent Selection for Routing Protocol in Wireless Sensor Networks using RPL, *5th International Conference on Web Research (ICWR)*, pp. 79-84, Apr, 2019.

[23] K. Jaiswal, and V. Anand, EOMR: An Energy-Efficient Optimal Multi-path Routing Protocol to Improve QoS in Wireless Sensor Network for IoT Applications, *Wireless Personal Communications*, Vol. 111, no. 4, pp. 2493-2515, Apr, 2020.

[24] S. K. Preeth, R. Dhanalakshmi, R. Kumar, and S. Si, Efficient parent selection for RPL using ACO and coverage based dynamic trickle techniques, *Journal of Ambient Intelligence and Humanized Computing*, Vol. 11, no. 11, pp. 4377-4391, Nov, 2020.

[25] S. Sankar, and P. Srinivasan. Fuzzy logic based energy aware routing protocol for Internet of Things, *International Journal of Intelligent Systems and Applications*, Vol. 10, no. 10, pp. 11, Oct, 2018.

[26] P. Singh, and Y. C. Chen, RPL Enhancement for a Parent Selection Mechanism and an Efficient Objective Function, *IEEE Sensors Journal*, Vol. 19, no. 21, pp. 10054-10066, Jul, 2019.

[27] T. D. Nguyen, J. Y. Khan, and D. T. Ngo, A distributed energy-harvesting-aware routing algorithm for heterogeneous IoT networks, *IEEE Transactions on Green Communications and Networking*, Vol. 2, no. 4, pp. 1115-1127, May, 2018.

[28] "IEEE Draft Standard for Local and Metropolitan Area Networks Part 15.4: Low Rate Wireless Personal Area Networks (LR-WPANs) Amendment to the MAC sub-layer," IEEE P802.15.4e/D 6.0 (Revision of IEEE Std 802.15.4-2006), pp. 1–200, Aug, 2011.

[29] L. Wallgren, R. Shahid, and V. Thiemo, Routing attacks and countermeasures in the RPL-based internet of things, *International Journal of Distributed Sensor Networks*, Vol. 9, no. 8, pp. 794326, Aug, 2013.

[30] J. Ko, J. Eriksson, N. Tsiftes, S. Dawson-Haggerty, A. Terzis, A. Dunkels and D. Culler, ContikiRPL and TinyRPL: Happy Together, *Proceedings of the workshop on Extending the Internet to Low power and Lossy Networks (IPSN)*, April 12-14, 2011.

[31] A. Dunkels, J. Eriksson, N. Finne and N. Tsiftes, Powertrace: Network-level power profiling for low-power wireless networks, 2011.

[32] R. M. Srinivasa, and D. N. Rao. "Faulty Nodes Detection for Reliable Data Transmission in Intelligent Wireless Sensor Networks." *International Journal of Intelligent Engineering & Systems* 17, no. 2024.

[33] V. Mohammadi, A. M. Rahmani, A. Darwesh, and A. Sahafi. "Fault tolerance in fog-based Social Internet of Things." Knowledge-Based Systems vol. 265 pp. 110376, 2023.

[34] Z. Qinbin, T. Zhao, X. Chen, Y. Zhong, and H. Luo. "A fault-tolerant transmission scheme in SDN-based industrial IoT (IIoT) over fiber-wireless networks." *Entropy* 24, no. 2, pp. 157, 2022.

**Mohsen Mozafari Vanani** received the B.Eng. degree from Islamic Azad University, Shahrekord branch, Iran, in 2020 and the M.S. degree from Islamic Azad University, Shahrekord branch, Iran, in 2021. Also since 2022, he is a Ph.D. student at Islamic Azad University, Central Tehran branch, Iran. His Master's thesis with subject Improving Loading communications and exchanges with an optimization approach and the use of ant colony algorithm and with an excellent score It has been judged.

**Pouya Khosravian Dehkordi** received the B.Eng. degree from Islamic Azad University, Najafabad branch, Iran, in 2005 and the M.S. degree from Islamic Azad University, Arak branch, Iran, in 2008. Since 2009, he is a faculty member of Islamic Azad University, Shahrekord branch, Iran. Also since 2014, he is a Ph.D. student at Islamic Azad University, Yazd branch, Iran. His Ph.D. thesis deals with Service Function Chaining. His current research interests include Software Defined Networks, Service Function Chaining, Natural Language Processing, and Automata Theory.