IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 11, 2022, accepted February 7, 2022, date of publication February 10, 2022, date of current version February 18, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3150928

RPL Based Emergency Routing Protocol for Smart Buildings

RONG-GUEI TSAI[1], PEI-HSUAN TSAI[2], (Member, IEEE), GUAN-RONG SHIH[2], AND JINGXUAN TU[1]

[1]New Engineering Industry College, Putian University, Fujian 351100, China
[2]Institute of Manufacturing Information and Systems, National Cheng Kung University, Tainan City 701, Taiwan

Corresponding author: Pei-Hsuan Tsai (phtsai@mail.ncku.edu.tw)

This work was supported in part by the Ministry of Science and Technology of Taiwan; in part by the Industrial Technology Research Institute of Taiwan under Grant MOST 108-2221-E-006-095-MY2 and Grant MOST 110-2221-E-006-008; and in part by the Higher Education Sprout Project, Ministry of Education to the Headquarters of University Advancement, National Cheng Kung University (NCKU). The work of Rong-Guei Tsai was supported in part by the Education Department of the Fujian Province Project, China, under Grant JAT190580; and in part by the Science Foundation of the Fujian Province Project, China, under Grant 2020J01924.

ABSTRACT The IPv6 routing protocol for low-power and lossy networks (RPL) is a routing protocol widely used for internet of things (IoT). However, there is no RPL-based routing protocol designed for real-time effective routing in a fire incident scenario. The characteristics of sensor networks in a fire incident are that sensors or IoT devices are continuously broken, which varies the graph in RPL continuously. Therefore, this study proposes an emergency RPL (EMRPL), which can predict the trajectory of fire effectively and transmit sensing data during a fire incident in real-time. Compared with RPL, EMRPL can effectively increase the packet delivery ratio. This shows that EMRPL has higher efficiency and can be effectively applied to fire incidents.

INDEX TERMS Internet of Things, routing protocol, RPL.

I. INTRODUCTION

A. BACKGROUND

ESPITE the development of various sensors and alarms for real-time detection of fires and warnings, building fires are a lethal indoor disaster, in which injuries and deaths are still inevitable. One major reason is that people are warned of the occurrence of a fire without real-time fire information, which may result in people making incorrect decisions, such as running into dangerous areas during evacuation or firemen wasting time in the wrong area to search and rescue people. Fortunately, with the development of Internet of Things (IoT), the sensor network of building automation systems can transmit environmental sensing data to help with fire analysis and decisions to provide real-time information for fire rescue.

IoT devices have the characteristics of multi-source heterogeneity, and most devices have certain limitations in their computing capabilities, storage capabilities, communication capabilities, and energy reserves. Hence, the Internet Engineering Task Force (IETF) working group formulated a routing protocol suitable for IoT, called the IPv6 routing protocol for low-power and lossy networks (RPL). RPL is a distance-vector routing protocol that is developed based on IPv6. It constructs a destination oriented directed acyclic graph (DODAG) oriented by the destination node (root node). Each node within the network has an assigned rank value, which increases as the teams move away from the root node, thereby indicating the distance between itself and the root node. The nodes send packets using the lowest rank value as the route selection criteria.

However, there is no RPL based routing protocol designed for real-time effective routing in a fire incident [1], [3]. The object functions of most RPL-based routing protocols focus on the power consumption of sensors and bandwidth, security, which is different from the model and needs of the network in a fire incident. The characteristic of the sensor network in a fire incident is that sensors or IoT devices are continuously broken. This makes the graph in RPL vary continuously. Therefore, this study develops an emergency RPL (EMRPL), which can predict the trajectory of the fire effectively and transmit sensing data in a fire incident in real-time. For real-time transmission, the definition of ''real-time'' is

The associate editor coordinating the review of this manuscript and approving it for publication was Marco Martalo.

that the packet transmission meets its deadline instead of a fast transmission. In this paper, we use the packet delivery ratio to represent the performance of real-time because the peer-to-peer transmission time of an indoor sensor network is short when it is successfully transmitted. The missed deadline would occur when the transmission fails, and the packet is lost. The delay time is to present the length of the routing path of different algorithms. Although that our approach, EMRPL, determines a longer path, it guarantees a higher packet delivery ratio that meets the definition of real-time.

Because the damage to the sensor during a fire incident is significant and continuous, if the spread information of the fire is added, the time and location of the sensor damage can be predicted to improve routing efficiency. Therefore, the primary idea of this study is to utilize the time and location of sensor damage. In the packet transmitting process, the sensor selects a neighbor that is far away from the fire and closer to the destination root node to transmit the packet. Experiments have confirmed that this will improve the overall packet delivery ratio so that information on the fire location can be used. Furthermore, it provides relief personnel to use or guide personnel to escape from the fire.

To improve the packet delivery ratio at a fire scene, we provided three modes in our routing method. The first one is the initialization mode, whose main job is localization and initialization of neighbor information and root node information. Second is the normal mode, which focuses on effective transmission, and the third is the emergency mode, which uses fire prediction to establish routing paths to avoid frequent rebuilding of graphs to improve the packet delivery ratio.

B. CONTRIBUTIONS

To allow the information collected by the sensors to be used effectively, we designed a routing protocol based on RPL that is suitable for a continuous and massively damaged dynamic network environment considering the aftermath of disasters. The main contributions of this study are summarized as follows.

Presently, there is no RPL-based routing protocol designed for real-time effective routing in a fire incident. This study proposes EMRPL, which can increase the packet delivery ratio. Furthermore, EMRPL is suitable in a fire incident and helps to improve the efficiency of disaster relief.

We proposed adopting the idea of orthogonal projection, which decides the neighbor node for packet forwarding, thereby reducing the time taken to rebuild the record.

We analyzed the influence of EMRPL and RPL on the packet delivery ratio for the communication radius, number of fire sources, fire spreading speed, and number of root nodes.

According to the experimental results, when the communication radius was greater than 80 m, EMRPL was better than RPL by 2% – 9% in packet delivery ratio. Moreover, when the number of root nodes was greater than 12, the packet delivery ratio increased by 2% – 7%.

The remainder of this paper is organized as follows: In the second section, we explain the background of RPL and provide an overview of the current development of RPL. The third section describes the EMRPL method, and we explain the operation details of the three modes in EMRPL. The fourth section discusses the simulation experiment. We compared the performance of EMRPL and RPL for packet delivery ratio and delay time. The final section presents the conclusions of this study.

TABLE 1. RPL-based protocols.

Motivation	RPL-based protocols
P2P	P2P RPL [4], Geo Rank [5], ER-RPL [6], AODV-RPL [7]
Multicast	MPL [8], SMRF [9]
Mobility	Co-RPL [10, 11], mRPL [12], mod-RPL [13]
Traffic	LOADng-CTP [14], DT-RPL [15]
Power saving	RPLca+ [16], QoS RPL [17], multiELT-RPL [18], ERGID [19]
Object function	EAOF [20], L2AM [21], OF-FL [22], FUZZY OF [23], SCAOF [24], ETEN-RPL [29], MRHOF [30]

II. RELATED WORKS

RPL, which is based on IPv6, is a routing protocol for low-power and lossy networks (LLNs) [3]. IoT devices have specific requirements for routing owing to resource constraints and other factors. Based on these routing requirements, the RoLL working group designed an RPL routing protocol, which realized the vision of IoT through thousands of interconnected devices and multi-hop communication of messages. RPL is widely used in various fields, such as healthcare, smart cities, smart buildings, industry, military, etc. However, routing protocols have many considerations to meet application requirements. The routing protocol currently proposed for IoT/LLN was implemented while improving the existing basic protocols. However, the main and latest methods are based on RPL.

A. RPL OVERVIEW

RPL establishes DODAG using an objective function and routing cost. Each node in the DODAG (except the root node) chooses a parent node as the DODAG upward route. The objective function selects an optimal path based on the routing cost. However, there are various objective functions for a node. According to the requirement of the environment, the expected transmission times or delays can be used as the routing cost. Each RPL instance has a unique ID. Moreover, each RPL instance can have multiple DODAGs, and each DODAG has a unique ID. Therefore, one RPL Instance ID and one DODAG ID can determine a unique DODAG. RPL Instance ID 0 is composed of three DODAGs (ID from 0 to 2), as shown in Fig. 1. All nodes in the DODAG with the same RPL Instance ID will adopt the same objective function.

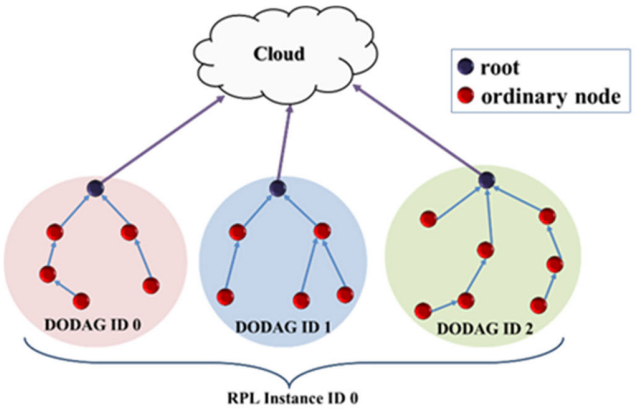

FIGURE 1. Example of RPL instance architecture.

In the RPL routing protocol, nodes establish routes by exchanging DODAG information object (DIO), DODAG information solicitation (DIS), and DODAG advertisement object (DAO) messages. The entire process is divided into two procedures: (1) the establishment of upward routing, and (2) the establishment of downward routing. After deployment, the root node designated as DODAG sends DIO messages to the neighboring nodes. The DIO message includes RPL Instance ID, DODAG version, DODAG ID, rank value of the root node, working mode of RPL routing, DODAG setting information, and routing cost [3].

An RPL instance contains multiple root nodes. The ordinary node selects the parent node based on the routing cost in the DIO and uses the objective function to calculate the rank value. After the node modifies the routing cost and rank value in the DIO, it sends DIO messages to the neighboring nodes. Similarly, other nodes find their parent node in the same manner and continue to send DIO messages to their neighbors. Ordinary nodes use the parent node for packet forwarding, so ordinary nodes send the sensing data upwards and then send the sensing data to the parent node until it reaches the root node. At this time, the root node cannot send data directly to ordinary nodes because the downward route has not yet been established. Among them, the DIS message provides nodes that have not joined the network to request DIO messages and seek to join the network.

In RPL, downward routing is established through DAO messages. When a node receives a DIO message, it forwards the DAO message through the parent node until it is sent to the root node. When the root node receives the DAO messages sent by all the nodes, it builds a routing table for all the nodes. When the root node wants to send packets to lower nodes, the root node constructs a route according to the routing table.

B. ENHANCED ROUTING PROTOCOLS IN IOTS

The routing protocol proposed for IoT/LLN was implemented while improving the existing and well-known basic protocols. However, the main and latest methods are based on RPL. Peer-to-peer (P2P) is an important transmission mode that is mainly used in IoTs. RoLL WG proposed a new solution for improving the performance of P2P transmission. Therefore, the IETF standard proposes a point-to-point routing protocol called P2P-RPL [4]. In [5], the author proposed a geographic routing method called GeoRank, which combines RPL and greedy other adaptive face routing to reduce the number of control messages. Considering that majority of P2P routing protocols create routes by controlling packets, which results in high cost and energy consumption, Zhao *et al.* proposed an energy-efficient region-based routing protocol that aims to not compromise network reliability. In the case of realizing energy-saving P2P communications [6]. In an actual environment, the routing requirements of various IoT applications must be met. Considering symmetric and asymmetric links in the path discovery process, RPL based on ad hoc on-demand distance-vector routing [7] aims to improve the P2P traffic pattern of RPL.

Multicast communication is the basic transmission type of IoT. A multicast protocol for LLN (MPL) was proposed in 2010 [8]. MPL was defined as the IETF standard by RFC 7731 in 2016. MPL uses a flooding mechanism controlled by a trick stream timer to perform multicast message transmission without maintaining a routing table. Although MPL can provide high reliability, it may also result in high end-to-end delays and communication costs. Therefore, Oikonomou *et al.* proposed stateless multicast RPL forwarding to mitigate these drawbacks [9].

In addition to multicast communication, support for mobile nodes is required for majority of applications in IoT environments. Therefore, an effective mobility protocol should support fast, continuous, and reliable communication between dynamic and static nodes. Therefore, Gaddour *et al.* proposed a method based on the corona mechanism to ensure the quality of service in LLNs using mobile nodes [10], [11]. Fotouhi *et al.* proposed mRPL as a solution to enhance mobility support in RPL [12]. Considering the necessity of mobility support for healthcare and medical applications, Gara *et al.* proposed an improved version of RPL called mod-RPL [13]. This method considers applications executed on a network, including mobile nodes and static nodes.

IoT/LLN applications may require different data traffic patterns during execution. For example, two nodes create P2P traffic to exchange messages, and then both nodes need to generate MP2P traffic and send data to the central node. The central node can use multicast communication to send data messages to specific nodes. Based on the requirements of using different data traffic modes, routing protocols should provide transmission modes for multiple traffic [14], [15].

The reliability of data transmission and acceptable delay are the basic requirements of all IoT applications. Owing to the power limitations of various IoT devices, it is necessary to perform packet transmission with less energy consumption. For all routing protocols of LLNs, feasible quality of service (QoS) and reasonable energy consumption are regarded as the most critical functions [16]–[19]. RPL uses the objective function (OF) to establish the network topology and the process of parent node selection. In RPL,

there is no mandatory use of a specific OF. This selection should be made according to application requirements. IETF defines two initial objective functions, OF0 and MRHOF. Although they can meet simple routing requirements, these OFs may still have limitations, such as not considering energy information during route selection. Therefore, considering different routing metrics, some studies [20]–[24] proposed different objective functions to meet the application requirements.

III. EMERGENCY RPL

A. DATA STRUCTURES OF EMRPL

To support the three modes in EMRPL, various data must be stored in the nodes and packets. There are several attributes of a node in EMRPL, as shown in Table 2. Usually, roots are nodes with complicated computations and large power capacity, such as a gateway. They are responsible for collecting data from ordinary nodes or sending commands to ordinary nodes. Ordinary nodes are usually sensors with a lower computing capacity and limited power capacity. Their main task is to collect information in the environment and transmit data back to the root node via multi-hop communication. In this paper, we only considered non-isolated nodes and left the problem of isolated nodes as a future work.

TABLE 2. Attributes of a node in EMRPL.

Node ID	Each node has an unique ID
Role	Root node/ordinary node
Absolute location	x, y, and z coordinates
Hop number	Root node: 0 / ordinary node: ∞
Rank value	Root node: 0 / ordinary node: ∞
Root info	Root ID, location, hop number
Neighbor table	Node ID: Null Rank value: Null State: Null Location: Null

B. MODES IN EMRPL

The entire network system is divided into three modes: initialization, normal, and emergency.

1) INITIALIZATION MODE

All the attribute values of a node listed in Table 2 must be decided in the initial mode and contain ID, role, absolute location, status, hops, rank, root info, and neighbors. The ID is unique to the sensor network. The roles of the node, which are defined by the users, are classified as root and ordinary. These nodes know their absolute positions during deployment or via localization algorithms [25]–[28]. Furthermore, hops record the minimum number of hops required from the node to the root node. The node with the lowest rank value is responsible for packet transmission. The attribute values of root info record the ID, location, and hops of the root nodes.

In the initialization mode, a neighbor table is constructed for each node that needs to be established using request packets. The neighbor table records information about its neighbor nodes, including neighbor ID, rank value, state, and location. The initial value of the hop number of the root node is zero, whereas the initial value of the hop number of the ordinary node is infinite. The state of neighbor nodes in the neighbor table indicates whether the neighbor node is damaged or not. However, the initial state of the neighbor node is null.

The attributes of the request packet shown in Table 3. contain the root ID, root location, number of hops, sender ID, and sender location, where the root ID represents the destination (root node) where the environmental data collected by the sensor can be delivered, root location is the absolute location of the destination, and the number of hops record the number of transfer times of the request packet from the root node to the receiver node. The sender ID and sender location record the ID and location of the neighbor node.

TABLE 3. Attributes of the request packet.

attributes	description
Root ID	indicate the ID of the root node R and request the sensor S to join the GODAG established by R.
Root location	indicate the location of the root node R
Number of hops	the initial value is 0, when the neighbor node S of the root node R receives plus 1 and forwards the request packet to its neighbor nodes
Sender ID	record the ID of the forwarder of the requested packet
Sender location	record the location of the forwarder of the requested packet

In the following example, we illustrate the creation of a graph using a request packet. Consider Fig. 2(a), we assume that a request packet is sent from the root node R and received by the neighbor node A. Node A records node R in the neighbor table. Because the root ID and sender ID are both R, the neighbor node R is a root node. The information of the neighbor node R is recorded in the neighbor table of node A (neighbor ID, rank value, state, location). Then, node A analyzes the hop number of the request packet. Node A finds that the hop number in the request packet ($hop(R) = 0$) is less than the hop number ($hop(A) = \infty$) recorded by itself, so it pluses one and updates its hop number to become a new hop number. In the initial mode, if the sensor receives request packets from other root nodes, the sensor records the information of the root node in its neighbor table. When a node receives the request packets from other root nodes with hop number equal to its hop number, it is directly discarded (ignored).

In Fig. 2(b), node A broadcasts a request packet with hop number = 1 to its neighbor nodes R, B, and C. The neighbor nodes R, B, and C record the information of node A in their neighbor table. The neighbor node R finds that the hop number in the request packet is higher than its hop number ($hop(R) = 0$) and discards the request packet directly.

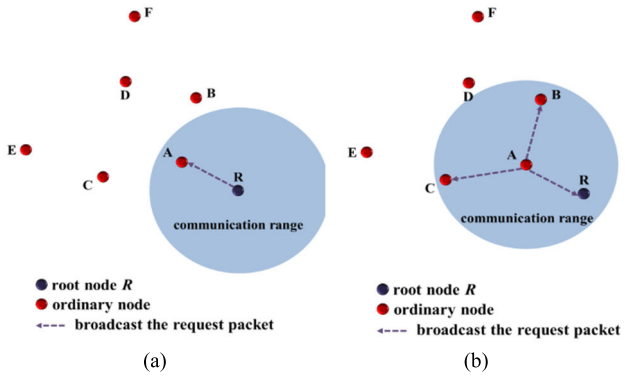

FIGURE 2. Example of (a) the root node R broadcasts request packet ($hop(R) = 0$) to its neighbor node A; (b) the node A broadcasts request packet ($hop(A) = 1$) to its neighbor nodes R, B, and C.

In the initial mode, the rank value of each node is defined as the number of hops; for node R, the rank value of node R is 0, the rank value of node A is one, and so on. The rank value is the result of the calculation obtained by using the objective function, as shown in (1). When none of the nodes receive any request packets from their neighbor nodes, the initialization mode ends. Then, sensor nodes begin to collect data in the environment, thereby entering the normal mode.

$$f(N) = hop(N) \tag{1}$$

However, in an actual network environment, there may be more than one root node. Therefore, when ordinary nodes receive the request packet sent from the root node, ordinary nodes check whether their root info already exists for the root node. If it already exists in the root info, the request packet will be ignored; otherwise, it will be recorded in their own root info. The root info of hop number is recorded as the minimum hop number from the ordinary node to the root node.

2) NORMAL MODE

When the initial mode was completed, the ordinary nodes began to collect and transmit data to the root node. Each ordinary node calculates the rank value of their neighbor nodes using the objective function $f(N)$, which is defined as the number of hops, as shown in (1).

Consider Fig. 3. Assuming that an ordinary node D wants to send data to the root node R, it selects the neighbor node with the smallest rank value as the forwarding node. In this case, D has two neighbors, F and B, where $f(F) = 4$ and $f(\mathrm{B}) = 2$. Therefore, D selects node B as the forwarding node. When the neighbor node B receives the data from node D, node B selects the neighbor node with the smallest rank value (A and D). Because $f(A) < f(D)$, node A is selected as the forwarding node and backhauled to the root node R layer-by-layer. The path represents the packet transmission path from node D to the root node R. In the network, any non-isolated ordinary node can establish a path from itself to the root node R in the normal mode. Therefore, after the initial mode is completed, the environmental information collected by all the ordinary nodes can be transmitted to the root node in this manner.

In many cases, the adjustment and expansion of the network scale are basic features. In the normal mode, there are two operations: adding a new node or removing a node. Fig. 3 and 4 illustrate the situation in which a new node is added. A new node F that wants to join the network executes a three-way handshake. Node F sends an *add message* to inform the neighboring nodes that node F wants to join the network. Assuming that node F has two neighbor nodes (nodes B and D). After receiving the add message from node F, they will compare their neighbor table and check whether node F already exists in their neighbor table. If node F already exists in their neighbor table, the add message will be ignored. Otherwise, nodes D and B return a request packet to node F individually.

When node F receives request packets from B and D, it compares the number of hops between nodes B and D. At this time, if it is found that the number of hops of node B is smaller than the number of hops of D, node F sets the number of hops to $f(B) + 1$, and the information of nodes B and D is recorded in the neighbor table. Node F transmits back the request packet to nodes B and D, who add the information of node F to their neighbor table.

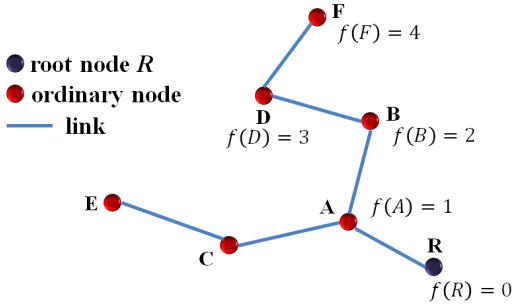

FIGURE 3. Example of routing of EMRPL in the normal mode.

Consider Fig. 4. Assuming that node F wants to leave the network (DODAG). It sends a *delete message* request to neighbors B and D. When nodes B and D receive the delete packet of F, node F removes from its neighbor table of nodes B and D.

3) EMERGENCY MODE

When a fire occurs, the sensor detects an abnormal sensing value and broadcasts an emergency packet to inform the neighboring nodes to enter the emergency mode. In the emergency mode, the emergency packet notifies the neighboring nodes that a fire has occurred. Fig. 5 shows the emergency

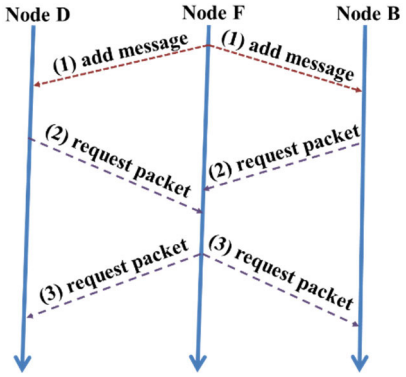

FIGURE 4. Adding a new node F to a DODAG.

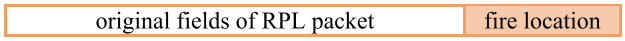

FIGURE 5. Emergency packet format.

packet format that includes the original fields of the RPL packet and the *fire location*, i.e., the fire location is the location of the sensor that detected the high-temperature event. The sensor node regards the relay node that broadcasts an emergency packet as a fire point, so the node should not be selected as a relay node.

In the normal mode, we only consider the root location, i.e., the number of hops of a packet to be transmitted from the source to the destination (root node). In the emergency mode, we consider two factors, i.e., the location of the *root node* and *the location of the fire source*. Suppose a sensor S_i wants to transmit sensing data to the root node R, we consider the vector $\overrightarrow{v_{l,r}}$ transmitted from S_i to the root node R. On the other hand, we also consider the impact of the fire source location. S_i chooses the nearest coordinate from the emergency packet as a new fire source location. To avoid selecting nodes closer to the fire source location as the transfer nodes, the inverse vector $\overrightarrow{v_{f,l}}$ is calculated.

In the emergency mode, the sensor continuously collects data in a specific period T and sends the data to the root node. However, the data transmission of the sensor fails because of damage to the root node or other relay nodes. The sensor nodes select the nearest root node as the new root node from among the remaining root nodes. The sensor nodes do not re-enter the initial mode. When a root is damaged, its neighbors would broadcast packets to announce the damage of root. For all the nodes with the same damaged root, they would change root by selecting a new one that is nearest it. The forwarding path is therefore rebuilt by using $\overrightarrow{v_{sum}}$ as described in the paper. When a relayed node is damaged, its successors reselect a new relayed node or a new root node. It is possible that the data transmission fails. However, according to our simulations, the EMRPL outperforms the RPL methods in terms of the packet delivery ratio.

In the root info, all the root node information is recorded. Therefore, the sensor selects the root node with the smallest hop distance (excluding the previous root node) from other root nodes as a new root node for data transmission.

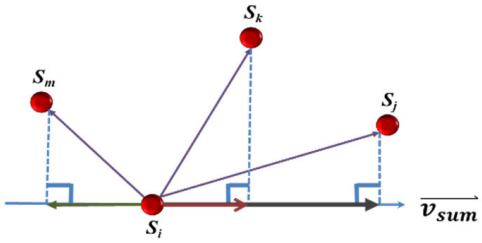

FIGURE 6. Example of selecting forwarding nodes in EMRPL.

Meanwhile, each node continuously updates the fire source location. When the sensor receives a new emergency packet, it is compared with the previous fire location. If the previous fire location is farther than the new fire location, the previous fire location will be replaced by a new location.

The sum vector $\overrightarrow{v_{sum}}$ of $\overrightarrow{v_{l,r}}$, and $\overrightarrow{v_{f,l}}$ is calculated, as shown in (2). The $\overrightarrow{v_{sum}}$ determines the direction in which the sensing data of sensor S_i should be transmitted. S_i finds a suitable forwarding node from the neighbor table and considers the orthogonal projection of the vector formed by each neighbor on $\overrightarrow{v_{sum}}$. Based on the scenario in this study, the nodes would be disable due to damages resulting in routing failures. To ensure the packet transmission ratio, we consider not only choosing the shortest path to root but also avoiding the nodes which would be highly possible disable soon. A sum vector $\overrightarrow{v_{sum}}$ combining the two objectives is designed to determine the transmission direction. Its value is set to the rank value of each neighbor node. Equation (3) shows the objective function $f(n)$ in the emergency mode, where $\overrightarrow{v_{sum}}$ is the sum vector of $\overrightarrow{v_{l,r}}$, and $\overrightarrow{v_{f,l}}$. $\overrightarrow{v_{l,n}}$ is the vector from sensor i to neighbor node n. Consider Fig. 6. Assuming that sensor S_i has three neighbor nodes, namely S_j, S_k, and S_m, we obtain three vectors, $\rightharpoonup S_lS_j$, $\overrightarrow{S_lS_k}$, and $\overrightarrow{S_lS_m}$. We calculated the orthogonal projection (rank value) of these vectors on the sum vectors and found that the rank value of S_j is larger. This means that $\overrightarrow{S_lS_j}$ is similar to $\overrightarrow{v_{sum}}$. Therefore, S_j was selected as the forwarding node of S_i. If there are two neighbor nodes with the same rank value, the sender node adopts FIFO to select the neighbor. In other words, the neighbor node that sent the rank packet earlier is selected as a forward node.

$$\overrightarrow{v_{sum}} = \overrightarrow{v_{l,r}} + \overrightarrow{v_{f,l}} \tag{2}$$

$$f(n) = \frac{\overrightarrow{v_{l,n}} \cdot \overrightarrow{v_{sum}}}{|\overrightarrow{v_{sum}}|} \tag{3}$$

4) EXAMPLE OF ROUTING IN EMERGENCY MODE

Consider Fig. 7, where a fire occurred near the sensor S_6. When node S_6 detects an abnormally high-temperature event, node S_6 broadcasts emergency packets to notify all the nodes to enter the emergency mode. The absolute position of S_6 is regarded as the fire location in emergency packets. The neighbor table of each ordinary node is updated and the new rank value of the neighbor is calculated using (3). Each ordinary node can select the node with the largest rank value

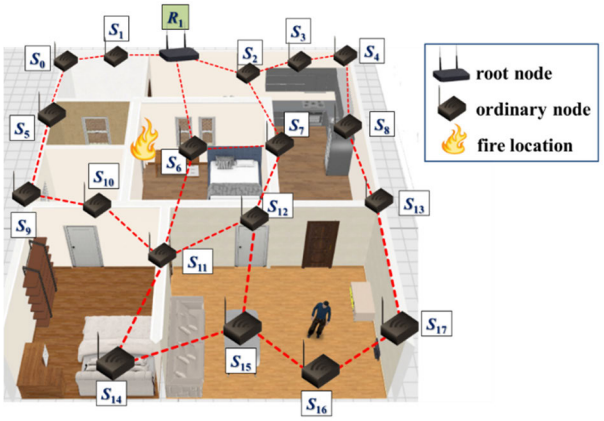

FIGURE 7. Fire incident to demonstrate emergency mode in EMRPL.

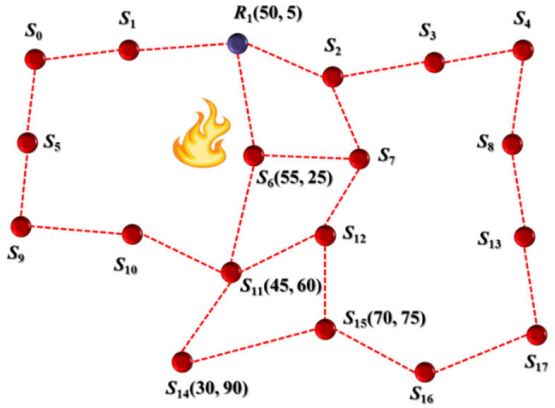

FIGURE 8. Example of routing in emergency mode in EMRPL.

from the neighbor table as the node for data forwarding. As the fire spreads, the fire area expands gradually. Each ordinary node continues to receive emergency packets from other nodes, selects a fire location closest to itself, and periodically updates the neighbor table (rank value of neighbor node). This determines the node for packet forwarding.

The data transmission of a sensor fails because of damage to the root node or other relay nodes. The sensor selects the root node with the smallest hop distance (excluding the previous root node) from other root nodes as a new root node for data transmission. In the emergency mode, similar to the normal mode, there are two fundamental operations for adding or removing nodes. The newly added node exchanges information with neighboring nodes through a three-way handshake method. The operation of adding and deleting nodes is the same as that in the normal mode.

To illustrate the operating process of EMRPL, we converted Fig. 7 into a two-dimensional plane, as shown in Fig. 8. When sensor S_6 detects an abnormally high temperature, it broadcasts an emergency packet to all sensor nodes. The emergency packet that is sent by node S_6 records the location of node S_6. Therefore, the location of S_6 was regarded as the fire location. Assuming that sensor S_{14} wants to send the sensing data to root node R_1. S_{14} selects the node with the largest rank value from its neighbor table as the forwarding node.

First, we calculate vector $\overrightarrow{v_1}$ from node S_{14} to root node R_1, as shown in (4).

$$\overrightarrow{v_1} = \overrightarrow{S_{14}R_1} = (50 - 30, 5 - 90) = (20, -85) \tag{4}$$

Because we expect the selected forwarding node to be far away from the fire location, the inverse vector $\overrightarrow{v_2}$ from node S_{14} to the fire location is calculated, as shown in (5).

$$\overrightarrow{v_2} = -\overrightarrow{S_{14}S_6} = -(55 - 30, 25 - 90) = (-25, 65) \tag{5}$$

Therefore, the vector of $\overrightarrow{v_1}$ and $\overrightarrow{v_2}$ is:

$$\overrightarrow{v_{sum}} = \overrightarrow{v_1} + \overrightarrow{v_2} = (20 - 25, -85 + 65) = (-5, -25) \tag{6}$$

In this case, S_{14} has two forwarding nodes, i.e., nodes S_{11} and S_{15}. Thus, we calculate the orthogonal projection of vector $\overrightarrow{b}$ on the sum vector $\overrightarrow{v_{sum}}$.

$$\begin{aligned}
\overrightarrow{b_{11}} &= \overrightarrow{S_{14}S_{11}} \\
proj_v b_{11} &= \frac{\overrightarrow{v_{sum}} \cdot \overrightarrow{b_{11}}}{|\overrightarrow{v_{sum}}|} = \frac{(-5 \times 45) + (-25 \times 60)}{-5^2 + (-25^2)} \\
&= 2.654 \\
\overrightarrow{b_{15}} &= \overrightarrow{S_{14}S_{15}} \\
proj_v b_{15} &= \frac{\overrightarrow{v_{sum}} \cdot \overrightarrow{b_{15}}}{|\overrightarrow{v_{sum}}|} = \frac{(-5 \times 70) + (-25 \times 75)}{-5^2 + (-25^2)}
\end{aligned} \tag{7}$$

In EMRPL, we define the rank value of each node as the orthogonal projection of vector $\overrightarrow{b}$ on the sum vector $\overrightarrow{v_{sum}}$ ode S_{14} selects a neighbor node with a larger rank value for data transmission. In this case, node S_{14} selects node S_{15} as the forwarding node. Next, node S_{15} continues to select a node from its neighbor nodes S_{12} or S_{16} as the forwarding node. As the data are transmitted from S_{14}, the child node S_{14} is not included to avoid the cycle. After forwarding several data, the data sent from S_{14} reaches the root node R_1.

IV. PERFORMANCE EVALUATION

A. SIMULATION CONFIGURATION PARAMETERS

In this study, we used 16 GB of DDR4-3000 memory. The operating system used Windows 10, according to different software and needs. We developed simulations using Java with MATLAB and Matplotlib package as auxiliary tools for drawing and complex calculations.

We conducted four sets of experiments to compare the EMRPL, RPL, ETEN-RPL [29], and MRHOF [30] in terms of packet delivery ratio and average delay time. Each set of experiments was conducted as follows: (A) change the communication radius of each node. (B) Change the spreading speed of fire. (C) Change the number of fire origins. (D) Change the number of root nodes. The parameters of the experimental settings are listed in Table 4. Each ordinary node sends out a packet every unit time. The entire experiment was executed under 120 unit time, and each set of experiments

TABLE 4. Parameter settings.

Items	Parameters
Network topology	Peer-to-peer (P2P)
Number of ordinary nodes	100
Number of roots	2 – 20
Area size	500 m × 500 m
Perform time	120 epochs
Communication radius of ordinary	30 m – 130 m
Rate of fire spread	1 m/s – 25 m/s
Number of origin of fires	1 – 19
Execution times	Perform 500 times and take the average
Sending frequency of packets	1 s/epoch
Transmission delay (one hop)	0.05 s

is the average value with 500 times. Assuming that it takes 0.05 seconds for a packet to pass through a hop.

The packet delivery ratio (PDR) is defined as the ratio of the total number of packets successfully received by the root node to the total number of packets sent by the sensor nodes (refer to (8)). The average delay time (ADT) is defined as the time required for the packet to be successfully transmitted from the sensor to the root node (refer to (9)). $Suc_{received}$ is the total number of packets successfully received by the root node, $packet_{total}$ is the total number of packets sent by all sensor nodes, and HC is the total hop counts of all successfully received packets to the root node.

$$PDR = \frac{Suc_{received}}{packet_{total}} \tag{8}$$

$$ADT = \frac{HC \times delay}{Suc_{received}} \tag{9}$$

B. INFLUENCE OF COMMUNICATION RADIUS

The node gets damaged from the fire. The density affects the number of neighbors that EMRPL can forward packets to. Fig. 9 shows the relationship between the node communication radius and the average density. The average density is defined as the average number of neighbors of a node. When the communication radius is 30 m, the average node density is approximately 2. Similarly, when the communication radius is 40 m, the average node density is approximately 2.8. Therefore, when the communication radius is 130 m, the average node density is 17.7. In other words, as the communication radius increases, the average density of nodes also increases, which means that the number of neighbors that can forward packets also increases, as shown in Fig. 10.

Fig. 10 shows the influence of the communication radius on the packet delivery ratio and average time delay, respectively. The communication radius is increased from 30m to 130m, and each set of experiments increases by 10m. The fire spread speed is 5 m/s and the fire point is randomly generated. There are five root nodes and 100 ordinary nodes which were randomly deployed in a 500 * 500 m environment.

As the communication radius increases, the number of neighbor nodes for a node also increases. In other words, there are more transmission paths available for packets, so the packet delivery ratio gradually increases, and the average delay time decreases. When the communication radius is less

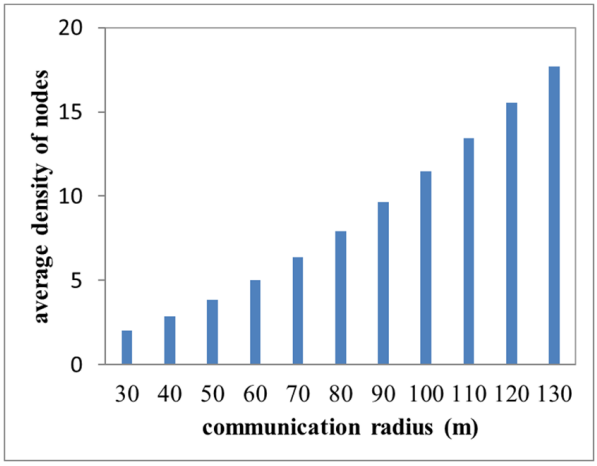

FIGURE 9. Relationship between communication radius and average density.

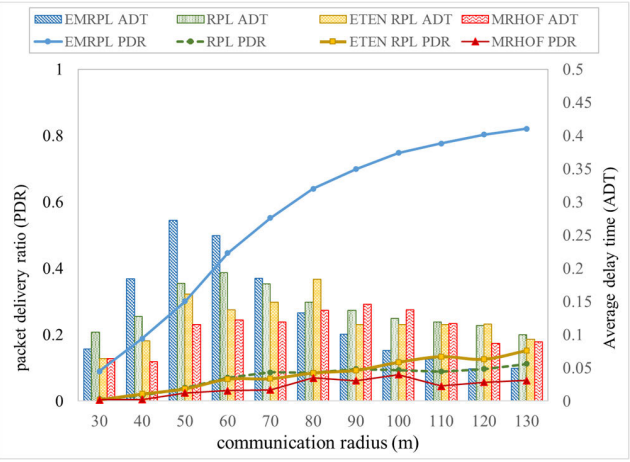

FIGURE 10. Influence of communication radius for packet delivery ratio and average delay time.

than 80, the average delay time of EMRPL is still larger than the other three methods. This is because EMRPL considers the position of fire, so it chooses a longer path resulting in longer average delay time and a higher packet delivery ratio compared to other methods. When the communication radius is greater than 80 m, the average time delay of EMRPL is gradually smaller than that of other methods because the number of available transmission paths increases. Therefore, the increasing in the communication radius helps to reduce the average delay time in EMRPL. According to the experimental results, with the increasing of communication radius, the increasing of packet delivery ratio for other methods is not significant. It is because those other methods do not consider the position of fire to select the path.

C. SPREADING SPEED OF FIRE ORIGIN

The spreading speed of fire origin increases from 1 m/s to 25 m/s, and each set of experiments increases by 3 m/s. The communication radius is 50 m, a randomly generated fire point is generated, and the number of root nodes is 5.

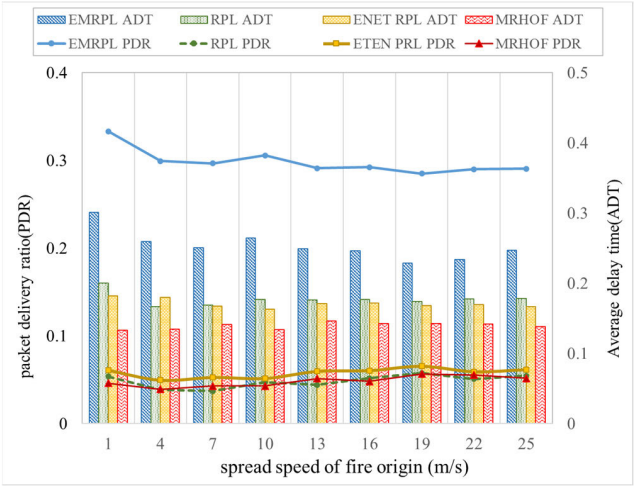

FIGURE 11. Impact of different spreading speeds of fire origin for packet delivery ratio and average delay time.

As Fig. 11 illustrates that with the increasing of fire spread, the variation of packet delivery ratio is not significant for all the methods. However, EMRPL performs better than other methods. In terms of average delay time, other methods are lower than EMRPL because nodes choose neighbors farther from the fire source as transmission nodes, resulting in a longer average delay time for EMRPL. In conclusion, the diffusion speed of fire insignificant impacts on the packet delivery ratio and the average delay time.

D. NUMBER OF FIRE ORIGIN

In this set of experiments, we discuss the influence of the number of fire sources on the packet delivery ratio and average delay time. The location of the fire was randomly generated with a spread rate of 5 m/s. There were 5 root nodes and 100 ordinary nodes were randomly deployed in an environment of 500 m $\times$ 500 m. The number of fire sources increased from 1 to 19. The communication radius of each node was 50 m.

Fig. 12 illustrates that the more the number of fire sources, the faster the increasing of damaged nodes. Experimental results demonstrates that EMPRL outperforms other methods in harsh network environments. As the number of fire sources increases, the packet delivery ratio decreases slightly, and the EMRPL is about 0.25 higher than other methods. From this experimental result, it is shown that the location of the fire point is an important factor in improving the packet delivery ratio.

E. NUMBER OF ROOT NODES

In the actual situation, each node selects the nearest root node as the destination root node. However, the root node may be damaged, causing the node to re-establish routing. Therefore, a better routing protocol must comply with real-time in a harsh environment where a large number of nodes are damaged.

FIGURE 12. Influence of number of fire origin for packet delivery ratio and average delay time.

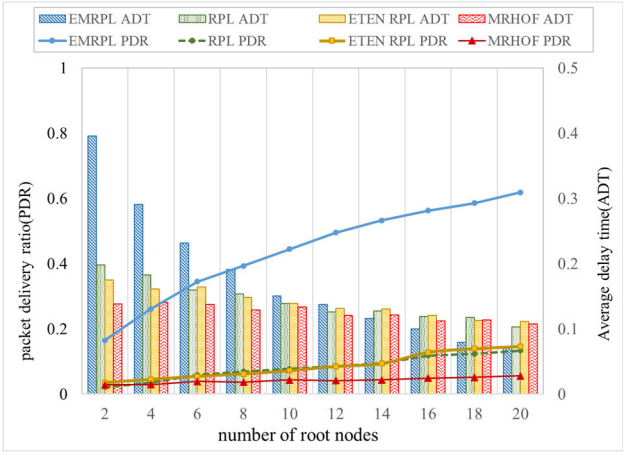

FIGURE 13. Influence of number of root nodes for packet delivery ratio and average delay time.

Fig. 13 illustrates that with the number of root nodes increases, the packet delivery ratio and the average delay time of packets are improved because there are more paths available. Although the average delay time of EMRPL is higher than other methods when the number of roots is less than 14, the difference between EMRPL and other methods is decreasing. When the number of root nodes is 14, the average delay time of EMRPL in the packet is lower than other methods. It is because that when there are sufficient roots to build sufficient paths, EMRPL would select the short and safe path that avoids fire sources.

F. LESSONS LEARNED AND IMITATIONS DISCUSSION

Although RPL performs better than EMRPL in delay time, EMRPL performs better than RPL in packet delivery ratio (PDR) which is more important for real-time. It is because the concept of real-time here is that the packet transmitted to the root node before its deadline. In fire applications, we hope that the fire information can be safely transmitted to the root node. In terms of packet delivery ratio, the packet delivery ratio of the RPL protocol is low, which means

that there are more packets that miss deadlines and most of the packets cannot be transmitted to the root node. Results indicate EMRPL is more in line with real-time. To achieve a safe transmission path in EMRPL, packets are transmitted over a longer hop distance. This is because EMRPL considers the location of the fire but does not consider the spread of the fire, resulting in EMRPL having a relatively long hop distance. In this paper, we compare the performance of EMRPL and other methods in terms of packet delivery ratio and average delay time through four influencing factors. Experiments show that the changes in the communication radius and the number of root nodes have greater influence than the spreading speed of fire origins and the number of fire origins.

V. CONCLUSION AND FUTURE WORK

Building fires are lethal indoor disasters in which injuries and deaths are inevitable. Generally, this is because people are warned of the occurrence of a fire without real-time fire information, which results in people making incorrect decisions. With the development of IoT, the sensor network of building automation systems can transmit environmental sensing data to help in fire analysis and make correct decisions to provide real-time information for fire rescue. RPL is a distance-vector routing protocol that was developed based on IPv6. There is no RPL-based routing protocol designed for real-time effective routing in case of a fire. The characteristics of sensor networks in fire are that sensors or IoT devices are continuously broken, which varies the graph in RPL continuously. Therefore, this study developed EMRPL, which is based on predicting the trajectory of fire effectively and transmit the sensing data in real time. Compared to RPL, EMRPL is more suitable for applications such as fire incidents, where sensors are damaged continuously. It is a highly efficient and feasible solution for providing real-time information for fire rescue in disaster relief. The future works of optimizing EMRPL includes shortest paths and isolated nodes. The paths selected by EMRPL is safe but not the shortest because the spread of fire is unpredictable. With the help of fire predication, EMRPL can be improved to select a shortest safe path to reduce the average delay time of EMRPL. On the other hand, the damaged sensor may cause isolated nodes in sensor network. In this paper, we assume that there are no isolated nodes in the sensor network. We consider using multiple communication radiuses to overcome the problem of isolated nodes to improve the packet delivery ratio.

REFERENCES

[1] N. Nasser, L. Karim, A. Ali, and M. Anan, "Routing in the Internet of Things," in *Proc. Global Commun. Conf.*, 2017, pp. 1–6.
[2] H.-S. Kim, J. Ko, D. E. Culler, and J. Paek, "Challenging the IPv6 routing protocol for low-power and lossy networks (RPL): A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2502–2525, 4th Quart., 2017.
[3] H. Kharrufa, H. A. A. Al-Kashoash, and A. H. Kemp, "RPL-based routing protocols in IoT applications: A review," *IEEE Sensors J.*, vol. 19, no. 15, pp. 5952–5967, Aug. 2019.
[4] M. Goyal, E. Baccelli, M. Philipp, A. Brandt, and J. Martocci, *Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks*. document RFC 6997, IETF Secretariat, Fremont, CA, USA, 2013.
[5] C. H. Barriquello, G. W. Denardin, and A. Campos, "A geographic routing approach for IPv6 in large-scale low-power and lossy networks," *Comput. Elect. Eng.*, vol. 45, pp. 182–191, Jul. 2015.
[6] M. Zhao, I. Wang-Hei Ho, and P. H. J. Chong, "An energy-efficient region-based RPL routing protocol for low-power and lossy networks," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1319–1333, Dec. 2016.
[7] S. Anamalamudi, M. Zhang, A. R. Sangi, C. E. Perkins, and S. Anand, *Internet-Draft DRAFT-IETF-ROLL-AODV-RPL-03*, Internet Engineering Task Force, Fremont, CA, USA, 2018.
[8] J. Hui and R. Kelsey, *Multicast Protocol for Low-Power and Lossy Networks (MPL) IETF Secretariat*. document RFC 7731, Fremont, CA, USA: 2016.
[9] G. Oikonomou, I. Phillips, and T. Tryfonas, "IPV6 multicast forwarding in RPL-based wireless sensor networks," *Wireless Pers. Commun.*, vol. 73, no. 3, pp. 1089–1116, 2013.
[10] O. Gaddour, A. Koubaa, R. Rangarajan, O. Cheikhrouhou, E. Tovar, and M. Abid, "Co-RPL: RPL routing for mobile low power wireless sensor networks using corona mechanism," in *Proc. 9th IEEE Int. Symp. Ind. Embedded Syst. (SIES)*, Jun. 2014, pp. 200–209.
[11] O. Gaddour, A. Koubáa, and M. Abid, "Quality-of-service aware routing for static and mobile IPv6-based low-power and lossy sensor networks using RPL," *Ad Hoc Netw.*, vol. 33, pp. 233–256, 2015.
[12] H. Fotouhi, D. Moreira, and M. Alves, "MRPL: Boosting mobility in the Internet of Things," *Ad Hoc Netw.*, vol. 26, pp. 17–35, Mar. 2015.
[13] F. Gara, L. Ben Saad, R. Ben Ayed, and B. Tourancheau, "RPL protocol adapted for healthcare and medical applications," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Aug. 2015, pp. 690–695.
[14] J. Yi and T. Clausen, "Collection tree extension of reactive routing protocol for low-power and lossy networks," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 3, Mar. 2014, Art. no. 352421.
[15] H.-S. Kim, H. Cho, H. Kim, and S. Bahk, "DT-RPL: Diverse bidirectional traffic delivery through RPL routing protocol in low power and lossy networks," *Comput. Netw.*, vol. 126, pp. 150–161, Oct. 2017.
[16] E. Ancillotti, R. Bruno, and M. Conti, "Reliable data delivery with the IETF routing protocol for low-power and lossy networks," *IEEE Trans. Ind. Informat.*, vol. 10, no. 3, pp. 1864–1877, Aug. 2014.
[17] B. Mohamed and F. Mohamed, "QoS routing RPL for low power and lossy networks," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 11, Nov. 2015, Art. no. 971545.
[18] O. Iova, F. Theoleyre, and T. Noel, "Using multiparent routing in RPL to increase the stability and the lifetime of the network," *Ad Hoc Netw.*, vol. 29, pp. 45–62, Jun. 2015.
[19] T. Qiu, Y. Lv, F. Xia, N. Chen, J. Wan, and A. Tolba, "ERGID: An efficient routing protocol for emergency response Internet of Things," *J. Netw. Comput. Appl.*, vol. 72, pp. 104–112, Sep. 2016.
[20] C. Abreu, M. Ricardo, and P. M. Mendes, "Energy-aware routing for biomedical wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 40, no. 1, pp. 270–278, 2014.
[21] S. Capone, R. Brama, N. Accettura, D. Striccoli, and G. Boggia, "An energy efficient and reliable composite metric for RPL organized networks," in *Proc. 12th IEEE Int. Conf. Embedded Ubiquitous Comput.*, Aug. 2014, pp. 178–184.
[22] O. Gaddour, A. Koubaa, N. Baccour, and M. Abid, "OF-FL: QoS-aware fuzzy logic objective function for the RPL routing protocol," in *Proc. 12th Int. Symp. Model. Optim. Mobile, Ad Hoc, Wireless Netw. (WiOpt)*, May 2014, pp. 365–372.
[23] P.-O. Kamgueu, E. Nataf, and T. Ndie Djotio, "On design and deployment of fuzzy-based metric for routing in low-power and lossy networks," in *Proc. IEEE 40th Local Comput. Netw. Conf. Workshops (LCN Workshops)*, Oct. 2015, pp. 789–795.
[24] H. Araújo, R. Filho, J. Rodrigues, R. Rabelo, N. Sousa, J. Filho, and J. Sobral, "A proposal for IoT dynamic routes selection based on contextual information," *Sensors*, vol. 18, no. 2, p. 353, Jan. 2018.
[25] P.-H. Tsai, "Building coordinate system of sensor nodes using self-configurable grid-based approach," *J. Inf. Sci. Eng.*, vol. 34, no. 2, pp. 451–468, 2018.
[26] P.-H. Tsai, R.-G. Tsai, and S.-S. Wang, "Hybrid localization approach for underwater sensor networks," *J. Sensors*, vol. 2017, pp. 1–13, Oct. 2017.
[27] P.-H. Tsai, G.-R. Shih, W.-D. Cheng, and R.-G. Tsai, "Σ-Scan: A mobile beacon-assisted localization path-planning algorithm for wireless sensor networks," *IEEE Sensors J.*, vol. 19, no. 23, pp. 11492–11502, Oct. 2019.
[28] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2019.

[29] L. Gao, Z. Zheng, and M. Huo, "Improvement of RPL protocol algorithm for smart grid," in *Proc. IEEE 18th Int. Conf. Commun. Technol. (ICCT)*, Oct. 2018, pp. 927–930.
[30] B. Sharma, J. Gajrani, and V. Jain, "Performance measurement of RPL protocol using modified MRHOF in IoT network," in *Proc. Int. Conf. Deep Learn., Artif. Intell. Robot.*, 2021, pp. 235–245.

RONG-GUEI TSAI received the M.Eng. degree in computer science from the National Chiayi University, Chiayi, Taiwan, in 2010, and the Ph.D. degree from the Institute of Manufacturing Information and Systems, National Cheng Kung University, Tainan, Taiwan, in 2019. In 2019, he joined the New Engineering Industry College, Putian University, as an Assistant Professor. His research interests include sensor networks, the Internet of Things, localization algorithms, and path planning algorithms.

PEI-HSUAN TSAI (Member, IEEE) received the M.Eng. degree in computer science from Cornell University, Ithaca, NY, USA, in 2004, and the Ph.D. degree in computer science from the National Tsing Hua University, Hsinchu, Taiwan, in 2010. In 2011, she joined the Institute of Manufacturing Information and Systems, National Chen Kung University, as an Assistant Professor, where she is currently an Associate Professor. Her research interests include sensor networks, data fusion, embedded systems, hospital automation, real-time scheduling algorithms, and user interface designs.

GUAN-RONG SHIH received the B.S. degree in computer science and information engineering from the National Cheng Kung University, Tainan, Taiwan, in 2017, where he is currently pursuing the Ph.D. degree with the Institute of Manufacturing Information and Systems. His research interest includes path planning algorithms for IoT.

JINGXUAN TU is currently pursuing the B.S. degree with the New Engineering Industry College, Putian University, Fujian, China. Her major is big data science and technology at Putian University. She has participated in the National Internet+ Contest and the National Internet of Things Competition for College Students in China. Her current research interests include the Internet of Things and path planning algorithms for wireless sensor networks.