



# Designing a context-aware model for RPL load balancing of low power and lossy networks in the internet of things

Zohreh Royaei<sup>1</sup> · Hamid Mirvaziri<sup>2</sup> · Amid Khatibi Bardsiri<sup>3</sup>

Received: 30 September 2019 / Accepted: 21 July 2020 / Published online: 16 August 2020  
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

## Abstract

The IPv6 routing protocol (RPL) for low power and lossy Networks (LLNs) was accepted as the standard routing protocol for the IoT by IETF in March 2012. Since then, it has been used for different IoT applications. Although the RPL deals considerably with IoT network requirements, there are still some open-ended problems to solve, for it was not initially designed for IoT applications. This paper addresses the RPL problems including load imbalance, which causes congestion in some nodes, significantly reduces the network performance, and decreases node energy and network lifetime. This paper proposes the automata-ant colony based multiple recursive RPL (AMRRPL), which is a modified version of the RPL for IoT networks, and uses a balancing model to avoid congestion. As a result, it will reduce network energy consumption, prolong the network lifetime, and reduce packet loss. The AMRRPL is evaluated in three steps. First, a multi-hop return objective function is presented based on the ant colony and computes the rank according to node context. The second step develops a new parent selection mechanism dynamically selected by stochastic automata and dynamic metrics for an optimal parent. General evaluation results show that this algorithm can make better decisions with regard to the optimal parent instead of making decisions simply based on the parent's rank. The third step resolve bottlenecks and swarm problems by managing the moving nodes through the heuristic flabellum algorithm inspired by physical and biological behaviour of flabella in the sea. Finally, the proposed algorithm performance is evaluated through the Cooja simulator. The proposed algorithm shows significant improvements in packet delivery and network lifetime, energy and convergence.

**Keywords** Load balancing · Context-aware · Routing protocol · Internet of things · Flabellum algorithm

## Abbreviations

IoT	Internet of things
AI Algorithm	Artificial intelligence algorithm
RPL	Routing protocol for LLNs
LLN	Low-power and lossy networks
DAG	Directed acyclic graph
DODAG	Destination oriented directed acyclic graph

## 1 Introduction

In recent years, the Internet of things (IoT) has become growingly popular with researchers worldwide. It has made its way into different scopes such as transportation, agriculture, industry, and healthcare due to specific features such as working in an IP-based network and being able to hold thousands or millions of nodes (Atzori et al. 2010; Kim 2014). These nodes are able to communicate and cooperate with each other to achieve a goal. Regarding the inherent attributes of IoT such as being IP-based, large-scale, and universally addressable, the Internet Engineering Task Force (IETF) has made some standardization efforts for the IoT (Kushalnagar et al. 2007). A pertinent IETF working group is the Routing Over Low power and Lossy networks (ROLL). As the name suggests, the main objective of this group is to focus on the routing of low power and lossy networks (LLNs) such as the IoT. In an LLN, there are some power-constrained nodes and typically one border router. In some cases, there are a couple of border routers (Kim

✉ Hamid Mirvaziri  
hamidmirvaziri@yahoo.com

<sup>1</sup> Department of Computer Engineering, Kerman Branch, Islamic Azad University, Kerman, Iran

<sup>2</sup> Department of Computer Engineering, Shahid Bahonar University, Kerman, Iran

<sup>3</sup> Department of Computer Engineering, Bardsir Branch, Islamic Azad University, Bardsir, Iran

2015). The border router is also known as the gateway. If a node is unable to communicate directly with the border router, it uses other nodes as intermediate nodes towards the border router. This process is handled by routing protocols in the network. Therefore, routing protocols play critical roles in delivering data from a node to the border router of the IoT (Yang 2017). In this regard, the IETF has standardized a routing protocol for the IoT, known as the RPL (Winter et al. 2012). The RPL enables users to define routing strategies according to their preferences about network requirements and metrics (Mayzaud et al. 2017). This facility is provided by the objective function (OF) concept, which is also one of the focus points in this paper. The OF defines how to decide on the suitability of a node in other to use it as a mean of achieving the network goal (Hassan 2016).

This paper aimed to analyse the problems with the RPL, one of the most important of which is lack of load balancing. As a result, some nodes suffer from congestion, something which severely reduces the node energy as well as the network performance and lifetime.

Energy is one of the limited resources in sensor networks. Therefore, it is one of the most critical issues in the IoT. In applications such as data gathering, environmental monitoring, and tracking, using a fixed power source and/or recharging a battery manually may not be both economically and technically feasible. To solve this problem, it is necessary to manage energy in IoT devices. Energy management can help prolong lifetime in these devices (Mittal et al. 2019).

This paper proposes the automata-ant colony based multiple recursive RPL (AMRRPL), which is the modified version of the RPL for IoT networks using a balancing model and avoiding congestion. As a result, it reduces network energy consumption and the packet loss rate consequently and prolongs the network lifetime.

Bottleneck, congestion, the effect of upstream parents, and ineffective parameters are the setbacks that prevent load balancing. Therefore, solutions have been proposed to solve these problems and perform load balancing in the RPL. Figure 1 shows the structure of the proposed method.

The AMRRPL was developed in three steps. First, a multi-hop return objective function is developed based on the ACO to compute the rank according to node context. The second step provides a new parent selection mechanism, dynamically selected by stochastic automata and dynamic metrics for the optimal parent. General evaluation results show that this algorithm makes better decisions with regard to the optimal parent instead of making decisions simply based on the parent’s rank. The third step resolve bottlenecks and swarm problems by managing the moving nodes through the heuristic flabellum algorithm inspired by physical and biological behaviour of flabella in the sea.

This paper consists of the following sections. Section 2 addresses the RPL briefly, and Sect. 3 reviews the research literature on the RPL and its objective function. Section 4 addresses the problem that is going to be solved. Contributions are then made to solve the problems stated in Problem

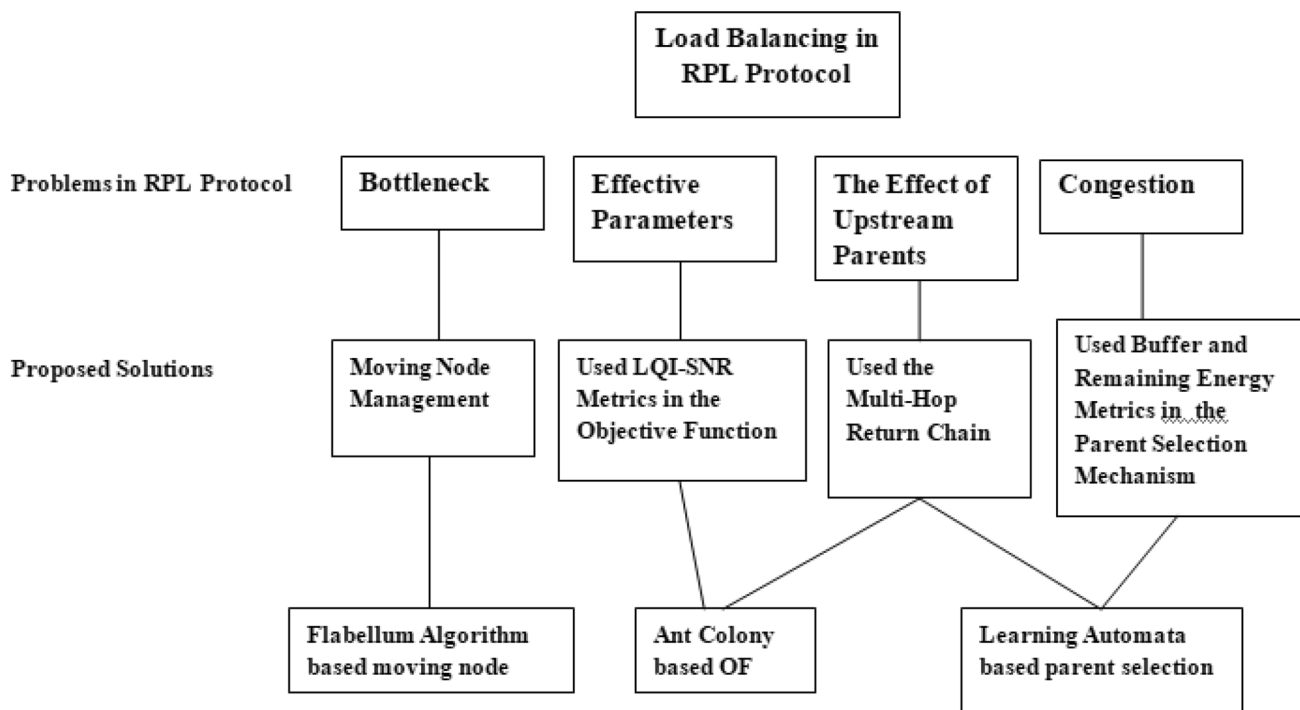


Fig. 1 Proposed methodology

Statement. Section 5 shows the experimental evaluation of the proposed protocol in different scenarios, whereas Sect. 6 draws a conclusion.

## 2 RPL: IPV6 Routing protocol for low-power and lossy networks

This section briefly introduces the RPL features and components. These definitions have been extracted from the IETF drafts.

Low-power and lossy networks face specific limitations on computation, communication, and available energy resources. The IETF formed the ROLL working group to determine the most suitable routing protocol for such networks. The ROLL group designed and standardized the RPL, based on the IPv6. This protocol uses an objective function to determine the line quality provided by each node to the gateway.

### 2.1 DODAG (destination oriented directed acyclic graph)

The DODAG is a set of vertices connected to a few edges with no distance. The RPL uses the DODAG concept to construct paths in the network. The RPL constructs a circular unidirectional graph forming certain paths from each leaf to the root (boundary router) (Winter et al. 2012).

### 2.2 Various RPL messages

There are three types of control messages in the RPL:

1. DODAG Object Information (DIO): this message is issued by the root and contains information about the DAG sample, including configuration parameters. It is similar to the one which the IPv6 uses for route advertisement (Dohler et al. 2009).

2. DODAG Information (DIS): this message is issued by a node for the DIO request and is useful for investigating neighbouring nodes.

3. Object Advertisement Destination (DAO): this message is used for sending the route information from nodes to the root. This message is sent universally to the selected parent (save mode) or the DODAG root (no-save mode). Reaching the root generates a complete route.

In the RPL, the gateway or the border router first begins issuing DIO messages by using the network. According to the objective function, the receiving node decides whether to select the gateway or not. Any node that selects the gateway as a parent starts to distribute the DIO message by using the network. This process is repeated until all nodes in the network are connected to the tree. After the tree is formed, leaf nodes issue and move the DAO message towards the

root through parents to determine the route of the traffic sent from the root to nodes and to generate a routing table (Dohler et al. 2009).

The objective function (OF) is a core concept in the RPL. The OF defines how differently metrics should be combined and translated into a rank so that the protocol will be able to use ranking to construct efficient routes. There are many routing metrics such as delay, packet loss, energy consumption, and link quality. The IETF has issued some drafts in this regard (Gnawali and Levis 2010; Thubert 2012). Currently, there are two standard OFs for the RPL. The first one is OF0 (Thubert 2012). OF0 works based on the hop count metric. In this OF, a rank is calculated by adding a value to the rank of the preferred parent. It does not consider link layer metrics, i.e. ETX (Expected Transmission Count) is the expected number of transmissions for a successful delivery of packets to the destination), and its main goal is to bring connectivity to the network. The second standard OF is the MRHOF (Minimum Rank with Hysteresis objective function) (Gnawali and Levis 2012). The MRHOF uses such metrics as ETX or latency as the rank computation basis. It also avoids the instability caused by small metric changes.

## 3 Related works

In the RPL design process, load balancing and congestion avoidance are not considered. The traffic passing through parent nodes and the size of their sub-trees are not considered in the parent selection process. This causes an unbalanced tree. There have been extensive studies of RPL load balancing, some of which are reviewed in the following section.

Imbalanced Tree Algorithm: Tripathi proposed a greedy algorithm in order to solve load balancing problem. He calculated the load imbalance factor for each routing level. In this way, the nodes that are prone to congestion are identified. This method aims to balance the routing tree and minimize the load imbalance factor. The algorithm selects a parent of a node from three nominated parents by itself. The root node executes the algorithm and tends to select the parents which minimize the load imbalance factor. This is performed periodically to keep the tree as balanced as possible during the network lifetime. This greedy algorithm aims to keep the load among same-level balanced nodes. The simulation results show that this algorithm significantly increases the average packet delivery ratio and network lifetime. This algorithm needs only a partial knowledge of the network which is the main algorithm characteristic (Tripathi et al. 2013). The algorithm proposed by Tripathi is only efficient for networks in which nodes generate the same amount of traffic. This method is completely centralized and is implemented by the root node. In addition, each period has

significant delays. The difficulty of the balanced problem is an NP type (Nabaei et al. 2018; Parsaei et al. 2016; Gao et al. 2017) and the time complexity of the proposed algorithm is  $O(N^2)$ . This algorithm involves large computational complexity; therefore, resources and time are highly consumed. The balanced tree construction is a complex mathematical problem. It needs a significant number of control messages. According to the specific conditions of low-power and lossy networks, especially poor communication links, the topology of these networks is completely dynamic and changes constantly. Balancing such topologies needs a great deal of effort to be repeated frequently.

**TREEB Algorithm:** Kulkarni proposed a method for load balancing. In this method, the DODAG root knows the number of nodes in each sub-tree. Each node that wants to join the DODAG becomes aware of the DODAG's node count and joins a DODAG with the lowest number of nodes (Kulkarni et al. 2012). In this method, nodes that want to join a DODAG can be aware of the DODAG size. This method has no effects on load balancing of each tree individually and just tries to keep the same size of trees. For instance, this method can generate completely unbalanced trees of the same size. When there is only one root, this algorithm is completely ineffective, only increases overhead, and slows down the tree generation process.

In the RPL, the topology construction and route selection are performed according to the objective functions (OF) and the routing factor. The objective function defines how to compute a node's rank and how to combine various factors in the rank calculation process. There is no guarantee in the RPL standard to use a specific objective function or a collection of criteria (Atzori et al. 2010). Therefore, it is possible to modify the RPL's OF and parameters by default. Some other RPL implementations have used other parameters such as hop count, ETC, or a combination of both. For instance, TinyRPL is a version of RPL and TinyOS, which combines OF0 with hop counts. The Contiki system, known as ContikiRPL in the RPL execution, uses the MRHOF as its default objective function, although it also includes the OF0 implementation. Its designer selects parameters and configures them in an OF; hence, the suitable definition of OF is still an interesting and open-ended topic in the RPL domain according to network requirements and preferences.

While using a hop count as a primary routing metric, the RPL significantly reduces the number of routers involved in the route, resulting in energy conservation. Considering the hop count per se disregards the energy-exhausted nodes and necessary retransmissions. Naturally, the hop-count-based RPL tends to select routes from the nodes that prematurely deplete battery's energy. However, the battery's energy of other nodes remains under-utilized (Karkazis 2013; Zhang and Li 2014). The authors in (Mamdouh et al. 2016) proposed Minimum Degree RPL (MD-RPL), which would build

a minimum degree spanning tree to enable load balancing in the RPL. The MD-RPL modifies the original tree formed by the RPL to decrease its degree. The MD-RPL improved the maximum consumed power, implying an improvement in the network lifetime.

An energy efficient routing technique is proposed in (Barbato et al. 2013) to address the importance of the energy efficient RPL in the IoT environment. This dynamic RPL decision considers the remaining energy of nodes and the required energy to route data traffic. Regarding the RPL, the node which is closer to the DODAG root node is involved in high traffic and completely exhausts its energy. Instead of router selection based on the traffic class, a restrictive approach is proposed to allow only the restricted nodes to forward the traffic. The paper reviewed by (Yang et al. 2014) analyses the RPL instability and estimates the control traffic for the RPL. This scheme mandates that the wireless links have to be bidirectional and symmetric. However, this assumption is unrealistic.

The author proposed a greedy approach in (Iova et al. 2013) to select the parent and make the network more stable through network dynamics. However, this approach needs frequent parent changes that leads to a great deal of network overhead.

A routing metric based on transmission delay and remaining energy was proposed in (Mohamed and Feham 2015). referred to as the QoS RPL (Quality of Service RPL). This algorithm benefits from the ant colony optimization (ACO) seeking to better fulfil the requirements of energy efficiency and QoS in LLNs. In the routing protocol functioning, the information about energy and delay is piggybacked on the control messages. This information is computed and updated by each node that receives a packet. This approach also uses the pheromone as a metric in the route selection process. The pheromone information is updated whenever a route is employed to forward a data packet, realizing the path reinforcement process. Thus, the most widely-used paths tend to be better evaluated by the proposed approach. However, the authors also implemented negative reinforcement to prevent the use of suboptimal routes and allow the adoption of other possibly better paths. The QoS RPL shows the ability to reduce delay and consume energy. Nonetheless, the packet delivery ratio shows a slight reduction compared with the RPL through the ETX metric.

A novel approach was proposed in (Iova et al. 2015) to improve network energy balancing and to maximize the lifetime of nodes. It is an RPL-based approach, and the authors focused on the generation of a routing solution considering an estimation of energy consumption. Using a mechanism to measure the expected lifetime (ELT) of nodes and exploring multipath, this approach tries to avoid using bottlenecks (nodes with less energy) and equalizes the power consumption. Each node should compute its ELT based on the traffic

expectation generated by itself and its children, the possible necessary retransmissions, transmissions time, and the transmission power of its radio. The authors also proposed a mechanism to limit the parental exchange. Thus, the quantity of a control message was reduced, contributing to fewer transmissions and energy depletion.

In (Capone et al. 2014), the authors proposed an additive composite metric called the lifetime and latency aggregatable metric (L2AM). The L2AM aims to provide balanced energy consumption, considering the reliability of data transmission along the path. For this purpose, the proposed routing metric merges a link reliability metric (i.e. ETX) with a new energy consumption metric referred to as the fully simplified exponential lifetime cost (FSELC). The FSELC represents the power cost that each node needs to pay for sending a message. During the RPL functioning, the nodes use DIO messages both for transporting information about the metrics and for informing the L2AM value computed for each neighbour. The L2AM value must be summed along the path to obtain the overall route cost. When a node needs to send a packet to the root, it should select the path of the lowest summation of L2AM. Thus, the proposed metric allows the node to use routes that are reliable and energy efficient.

Gozuacik in (2015) proposed the parent-aware objective function (PAOF). This is an objective function that aims to offer network load balancing for LLNs. The PAOF benefits from both ETX and parent count to perform the rank computing and preferred parent selection. The parent count metric represents the number of potentially preferred parents of the node. These two metrics are combined lexically. Comparing two candidate nodes, the PAOF first verifies the ETX modular difference between them. Whether it is smaller than the Min-Hop-Rank-Increase or not, the node should select the best parent to be the candidate with the lowest parent count. The Min-Hop-Rank-Increase is a default variable value of RPL that defines the minimum value increased in the ranking for each parent of the node. Thus, although PAOF considers two routing metrics, the primary decision is based on the ETX while the second metric being used just in case of a significant difference between ETX values of candidate nodes.

In a study on an energy efficient objective function targeted towards smart metering and industrial applications (Shakya et al. 2017), the authors used residual energy and expected energy consumption in the objective function named the smart energy efficient objective function (SEEOF). The results show 22–27% improvement in the network lifetime while compared with nodes by using the MRHOF as the objective function. The authors in (Sebastian and Sivagurunathan 2018) proposed load balancing optimization for the RPL-based emergency response by Q-learning (LBO-QL). This method faces the limitation of

communication among multiple DODAGs. As of now, the BR cannot communicate with another BR. A virtual framework needs to be generated to establish connections among the BRs. Therefore, there is a challenge to load balancing at multiple DODAG levels. Mobility plays an important role in the emergency response scenario. However, sufficient research on mobility model is desired. When there are mobile nodes, Q-learning computation will increase the control traffic overhead, and energy, as the reconstruction of DODAG, is frequent. Hence, the method is efficient for the single BR. The emergency scenario will have many BRs which interact with other IPV6-based networks. New optimization techniques for interoperability and load balancing in such environments in RPL need to be researched. The network requirement for emergency response is heterogeneous; thus, load balancing optimization for heterogeneous environment is a challenge. In (Kamgueu et al. 2013), residual energy is used as the only metric in the objective function. The results show that it improved the distribution of energy consumption and prolonged the network lifetime; however, it did not consider other important metrics such as packet loss, latency, or throughput.

In (Khan et al. 2016), control messages of the RPL are utilized to adjust the sub-network size relative to other sink nodes. Simulation results show an improvement in both throughput and energy distribution of the network nodes, leading to an improved lifetime. In a study of energy balancing, the authors proposed an RDC-based method of energy consumption estimation (Banh et al. 2016). They used this estimation as a metric for routing and achieved better distribution of energy as well as higher PDRs. However, improvement in energy consumption is marginal compared to the MRHOF used as the objective function. In addition, this provides no additional advantages other than marginal energy saving.

Other studies related to minimizing energy consumption have employed different approaches such as improving failure detection to improve energy efficiency in the RPL (Khelifi et al. 2015). This approach uses a suffering index that reflects the cost network failures and aims to improve energy consumption by pro-actively detecting failures. Some studies have proposed energy harvesting techniques for the efficient transmission of data. A routing and aggregation for minimum energy (RAME) technique (Riker et al. 2017) uses the information of the node with the lowest energy to regulate traffic. This approach limits throughput but is very effective in critical energy applications.

This paper focuses on load balancing and congestion problems as well as the DAG method under traffic and dynamic loads.

## 4 Proposed method

The RPL is unable to operate efficiently when network traffic is heavy and the network encounters numerous problems, including high loss rate, high energy consumption and load imbalance. In this section, RPL problems will be first categorized under heavy and dynamic loads then solutions for solving these problems are presented. Finally, the proposed method will be explained in details.

### 4.1 Problem statement

#### 4.1.1 Problem 1: inattention to effective parameters in the RPL

According to the network dynamics to select the best parameters in previous researches, the following results were obtained:

The ETX is used at lower rates; however, it is unusable at high transmission rates. At the same time, link quality is used for higher reliability rather than the number of steps. The number of steps has no effect on packet loss.

By defining the IEEE 802.15.4 protocol, it is clear that link quality instruction (LQI) is employed to demonstrate power and quality of receiving packets. Moreover, the signal-to-noise ratio (SNR) is usually used instead of noise power in wireless networks, for it is a more accurate parameter showing the difference between signal and noise. As a result, the proposed model will use LQI-SNR metrics instead of hop and ETX to affect load balancing.

#### 4.1.2 Problem 2: Effect of upstream parents

Any node may appear to meet the conditions to become a parent; however, the parent node may suffer from buffering or energy shortage. This is caused by selecting an incompatible parent in a heavy traffic network that leads node congestion because a good parent does not always yield optimal results, and the results of upstream parents are also important. Therefore, the multi-hop return chain was employed in proposed method.

#### 4.1.3 Problem 3: congestion

The best of the two parents are selected by the low stream nodes as the parent. This selection ignores the parent being selected by multiple nodes. Thus, the best parent becomes the cross point for a great deal of traffic, which will significantly reduce its efficiency due to limitations (bandwidth, buffer size and remaining energy). Accordingly, the parent buffer and parent remaining energy are considered in the

proposed method dynamically while using stochastic automata in an effort to solve this problem.

#### 4.1.4 Problem 4: load balancing

In the RPL design, no attempt has been made to balance the load and prevent congestion. The process of parent selection also does not consider the burden of the parent and its sub-tree size, resulting in the formation of a completely unbalanced tree.

The RPL also uses the OF to calculate the node rank and select parent; however, this causes the congestion to be transmitted from one node to another node because of being static at the execution time.

The load balancing issue requires a completely dynamic approach, one which is aware of the network load distribution and makes dynamic and knowledge-based decisions for load distribution and balancing.

Considering the fact that a node buffer is an appropriate approach to load balancing, although it may seem that buffer size can be involved to make the rank indicating the node's load, this approach will not be useful due to shifting congestion from the fully loaded node to a completely unloaded one from the previous period. This method only shifts congestion from one node to another in each period.

Since the aforementioned problems and the objective function are unable to solve the unbalancing issue alone and only change the congestion's location in each period, the mechanism for selecting the suitable parent for load balancing and reducing energy consumption is presented in the third phase. Training can be involved in this approach as follows: The node dynamically learns which parent provides the best route according to load balancing and remaining energy and node buffer.

## 4.2 Solutions to stated problems

This paper aims to present a modified version of RPL for the IoT networks, which uses a balancing model and prevents congestion to save network energy and increase lifetime.

Therefore, in the first step, parameters are prioritized, using test methods and available algorithms and the best ones are presented to the objective function. Dynamic and variable factors are investigated and tested during the routing phase and the best ones are selected for the training algorithm in parent selection mechanism.

The link quality indicator (LQI), signal-to-noise ratio, buffer size and remaining energy are the most effective factors extracted which are related to the node, link and channel used for balancing.

The OF is then applied by using the ACO and the three hop chain according to the resultant factors. With the

execution of the ACO algorithm, parent nodes are evaluated to introduce the five best parents for each node.

The final parent of each node is selected according to the stochastic automata learning algorithm and by using dynamic routing factors including buffer size and remaining node energy.

Finally, resolve bottlenecks and swarm problems by managing the moving nodes through the heuristic flabellum algorithm inspired by physical and biological behaviour of flabella in the sea.

The proposed method is evaluated through universal simulation in the Cooja simulator, and results are then compared with those of the previous algorithms.

### 4.3 Explaining the proposed method

The fully distributed nature of such networks, similar to the issues related to inter-node transfers, is caused by performance issues (route calculations). With regards to packet routing efficiency, ACO yields better results in most cases. This paper proposed an ACO based protocol for calculating dynamic routes, as well as a cooperation mechanism which provides better quality of service management in LLNs. The idea is designing an algorithm according to the decentralized actions of ants, which use their instinctive ability to find the shortest path from the origin to the destination. The focus has been on implementing an objective function which is specifically optimized for IoT networks. We have presented an approach that allows for using the link quality indicator (LQI) and the signal-to-noise ratio (SNR) as node routing factors in the optimal parent selection process for the RPL and for constructing a DODAG structure.

#### 4.3.1 Objective function

After receiving a DOI from a neighbour, each non-root node computes the cost of the route through this neighbourhood. The proposed objective function to find a route from the source to the destination is used through the parent that has a high transfer (cross) probability. The transition probability from Source  $i$  to Destination  $d$  through Parent  $j$  of Node  $i$  is computed as follows:

$$P_{ij,d} = \frac{[\tau_{ij,d}]^\alpha [LQI_{ij,d}]^\beta [SNR_{ij,d}]^\delta}{\sum_{i \in N_i, [\tau_{id}]^\alpha [LQI_{ij,d}]^\beta [SNR_{ij,d}]^\delta} \quad (1)$$

where  $\alpha$ ,  $\beta$  and  $\delta$  are pheromone indicating parameters, link quality index and the signal-to-noise ratio (SNR) respectively and all of them are greater than or equal to zero ( $\geq 0$ ).

- (A)  $\tau_{ij,d}$  is the relative weight of the pheromone trail.
- (B)  $LQI_{ij,d}$  is the heuristic value related to LQI.
- (C)  $SNR_{ij,d}$  is the heuristic value related to SNR.

Also,  $N_i$  is a collection of  $i$  parents and  $l$  is a parent of  $i$ , which provides a route to the  $d$  destination. The DIO message gathers the transfer quality of each link and the SNR of each node while moving on the network. The content updates its factors (by computing its path to the parent) and starts to send its own DIOs after a node computes the route's cost for all its neighbours and selects the best parent with regards to the related ranking for the selected factor.

Finally, the child node orders its list for the available parents through the highest degree of probability and connects to the root node through the highest  $P_{ij,d}$  value.

#### 4.3.2 Computing relative criteria

The link quality indicator (LQI) and the signal-to-noise ratio (SNR) are considered for calculating relative factors (criteria). The link quality indicator is the multi-hop average (path link quality is equal to the average link quality in all hops), whereas the signal-to-noise ratio is a concave function (the signal-to-noise ratio for one route is limited by the link that has access to the highest signal-to-noise ratio). While increasing factors should be minimized for the shortest paths, the convex function is used for maximizing the signal-to-noise ratio. In order to increase network lifetime, it is better to avoid selecting a node with low signal-to-noise ratio; since selecting the unsuitable link and a high noise rate in the link will increase network packet loss rate and waste resources (energy and time). Reciprocal values were not used for computing the aforementioned parameters, since they had the same ratios.

**4.3.2.1 Link Quality indicator** The link quality indicator (LQI) is the current parents' average LQI for the DIO message from the source to the destination through parents and while constructing the DODAG structure.

The three-hop LQI chain:

$$LQ(n) = Max \left\{ LQ(n) + \left( \frac{(LQ(n-1) + LQ(n-2) + LQ(n-3))}{3} * \theta \right) \right\} \quad (2)$$

The best  $\theta$  value was 0.20.

LQI was computed using the following formula, which is modelled based on the CC2430 microchip's real world hardware specifications.

$$LQI = (CORR - a) \cdot b \quad (3)$$

**4.3.2.2 Signal-to-noise ratio** The model proposed by (Halder and Kim 2012; Aljarrah 2017) is used for approximating the signal-to-noise ratio. This model is based on simple

calculations on low memory to match the sensor node while maintaining the original model’s accuracy. The model was implemented according to the following approximations. Maximum signal-to-noise ratio between the origin and the destination (Yang et al. 2018) is as follows:

$$SNR_{ijd} = \max \left\{ \frac{Signal}{Noise(t)} \right\} \tag{4}$$

$$\forall l \in route_j(i, d)$$

where  $Route_j(i, d)$  is the route from  $i$  to  $d$  through  $j$ .

The three-hop SNR chain:

$$SNR(n) = \max \left\{ SNR(n) + \left( \frac{(SNR(n-1) + SNR(n-2) + SNR(n-3))}{3} * \theta \right) \right\} \tag{5}$$

The best  $\theta$  value was 0.20.

**4.3.2.3 pheromone indicator** First, pheromone was applied to the  $(i, j)$  link as  $\tau_j = 0.0$  when there was no parent relation between  $i$  child and  $j$  parent node. When  $j$  was detected by using the DIO message as an  $i$  parent, an initial pheromone will be saved as  $\tau_{ij} = 0.1$ .

**4.3.2.4 route strengthening** When the data transfer begins, routes that have been strengthened will certainly be more suitable for future selections. The proposed objective function is incomplete without the pheromone trail evaporation process. In fact, it is necessary for the system to "forget" bad solutions in order to prevent suboptimal solutions. In addition, this is a negative reinforcement for the pheromone, the link quality indicator (LQI), and the signal-to-noise ratio (SNR). In other words, keeping bad solutions has a negative impact on reinforcement. In this paper the updated rules proposed in (Zhi and Hui 2015) are used and explained in details.

**4.3.2.5 Pheromone updating** The updated pheromone amount is obtained as follows:

$$\tau_{ij} = \begin{cases} (1 - \rho)\tau_{ij}, & \text{if } (1 - \rho)\tau_{ij} > 0.1, \\ 0, & \text{if } (1 - \rho)\tau_{ij} \geq 1, \\ 0.1 & \text{otherwise} \end{cases} \tag{6}$$

when node  $i$  loses its connection to the  $j$  parent, the pheromone on the  $i$  to  $j$  link will be set to zero. Pheromones, use the  $\rho$  parameter to control the link quality indicator (LQI) and the signal-to-noise ratio (SNR) at each cycle.  $(1 - \rho)$  is the pheromone evaporation factor, the link quality indicator (LQI) and the signal-to-noise ratio (SNR) on routes.

**Table 1** ACO Parameter Setting

Parameters	Values
$\alpha$	1.0
$\beta$	1.0
$\delta$	1.0
$\rho$	0.05

**4.3.2.6 Link quality indicator (lqi) update** The network link quality indicator will be subject to change continuously according to the quality of intermediary links between nodes. The link quality update rate is obtained as follows:

$$LQ_{ijd} = \begin{cases} \frac{\rho}{LQI(l)} + (1 - \rho).LQ_{ijd}, & \text{if } l \in route_j(i, d), \\ (1 - \rho).LQ_{ijd} & \text{otherwise.} \end{cases} \tag{7}$$

**4.3.2.7 Updating the signal-to-noise ratio (SNR)** The SNR is updated as follows:

$$\begin{cases} \rho.SNR(l) + (1 - \rho).SNR_{ijd}, & \text{if } l \in route_j(i, d), \\ (1 - \rho).SNR_{ijd} & \text{otherwise.} \end{cases} \tag{8}$$

**4.4 Rank calculation**

A node’s rank is a number that shows its status in a DODAG version and has defining factors. As a feature of the objective function, rank calculation is taken into account. These calculations depend on several factors such as all parents, link measurements, and node configurations.

Ranking is strictly (monotonically) increasing and can be used for evaluating the progress from the root. It can also be used for detecting and preventing routing loops. Rank calculation supports features which will be presented next. The rank value decreases in the upward direction (toward root node) and increases in the opposite direction. In addition, ranks presented by one node are required to have a higher value than destinations related to all its parents. In this case, there is no diameter to create a loop. A network can create a loop with nodes that have the same rank values when it selects a route node with the same rank value as itself. As soon as the  $i$  node (non-root) selects its  $j$  parent (highest), the node computes its rank according to the selected parent’s rank as follows:



$$\text{Rank}(i) = \text{Rank}(j) + \frac{1}{\text{Rank}(j) + P_{ij}} \tag{9}$$

where  $P_{ij}$  is the highest probability and  $i$  has selected the  $j$  parent.

The simulation's ACO parameter settings are defined in Table 1, which were obtained experimentally by the authors of (Zhi and Hui 2015; Tong et al. 2015) in the optimal state. Figure 2 is a Diagram of The Proposed objective function.

#### 4.4.1 Queue management and congestion control using learning automata

The approach in the second phase of the proposed method is focused on queuing problems in routing, which was implemented in the RPL in the IoT network. The IoT routing commonly takes place in hops where each protocol has designed an objective function based on its application. This specific objective function is in command of each node's policy and behaviour as regards to its application, as well as its neighbours. Each objective function is derived from a number of metrics affecting routing in the IoT networks. The manner and type of packet transfer can pose a challenge to the network, including routing metrics, which are commonly

known as the network control packets. The main issue is that transferring the control messages is necessary for generating, maintaining and variable topology, whereas increasing control requires a higher number of transfer packets, which is a trade-off. Some RPL-based protocols use the drop scheduler to better manage control messages. This paper proposes a load balancing method with a predictive approach to managing RPL network exchanges. This method is a latency-based approach that tries to prevent queue buffer overflow and implement a preventative mechanism for managing future network traffic.

Each parent (non-leaf) node in the RPL mechanism has two important roles: First, transferring its own packets to the root and transferring packets received from other children leading to itself, known as the sink. The important point with regard to wireless networks, however, is that there is no guarantee to have access to wireless media in the network. In order to exchange information to the parent, the network nodes have to spend time obtaining the media and perform the transfer. Each node may receive a number of packets in the stated waiting time, while not being able to transfer them to its parent.

Some network queuing rules such as FIFO belong to this group. Setting policies for filling and emptying the queue

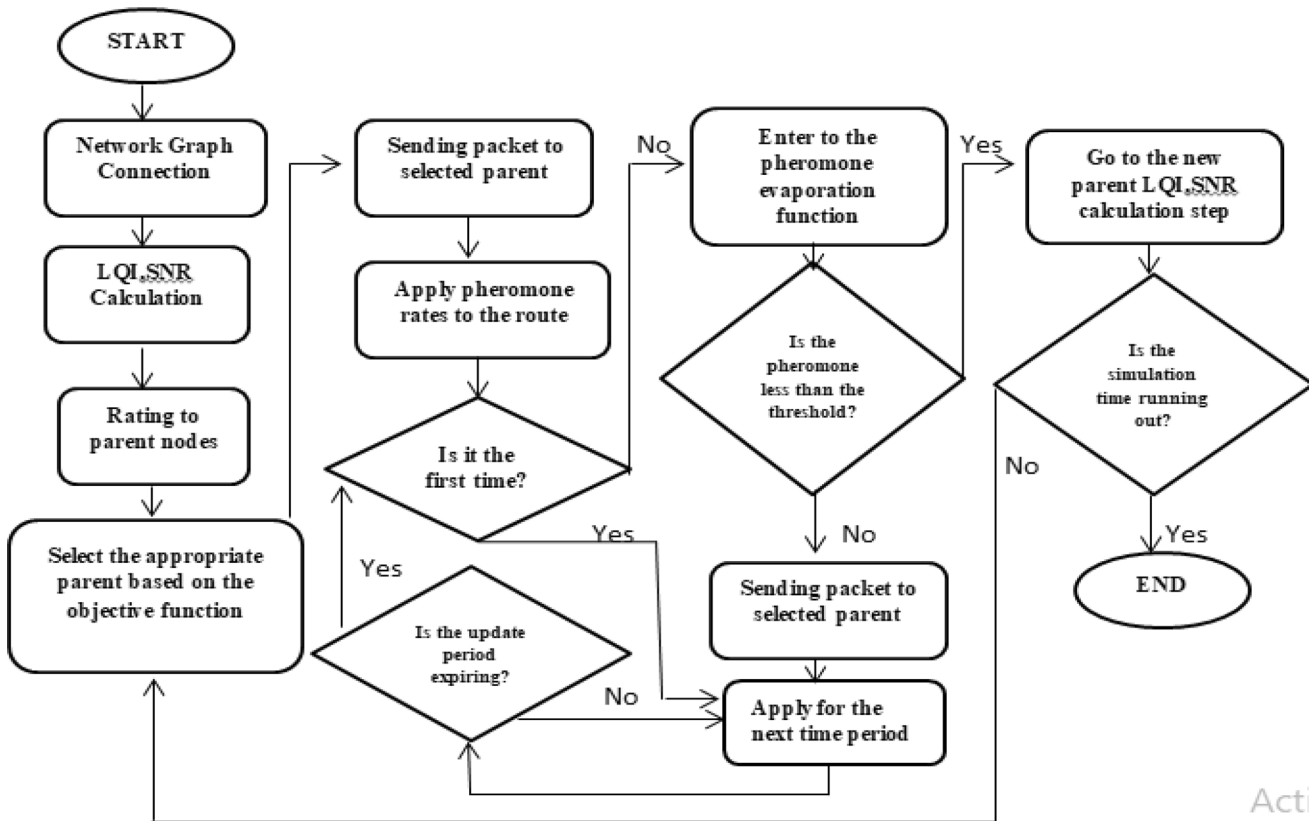


Fig. 2 Diagram of the proposed objective function

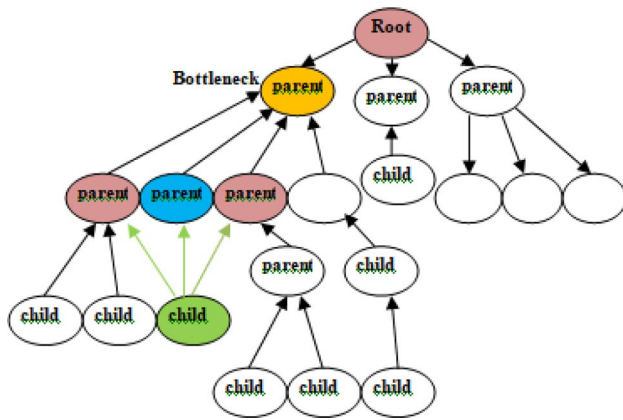


Fig. 3 The structural graph of the proposed ant colony method

in each node is one of this research main goals as well as an indicator for evaluating the packet acceptance or packet transfer rate. This metric often prevents network queue overflow. Figure 3 shows a representation of the proposed method’s structural graph and the reason for congestion occurring in the network.

In the proposed method, for each node, each generated or forwarded packet is queued and will be transmitted according the first-in-first-out (FIFO) policy. For each node, whenever a data packet is queued for transmission, the time is recorded by using the node local clock. When the same packet is de-queued, the time is also recorded. The difference between the de-queued and queued instants is the packet queuing delay.

The average value of the queuing delay is calculated over the last ten packets (ten packets is a value obtained through heuristics showing that smaller values generate oscillations and that bigger values prevent nodes from accurately updating its delay). It is called the node delay (d). If the node has not de-queued ten packets yet, d is the average value of already de-queued packets. To make the queuing delay average more representative of the recent traffic conditions, a weighing factor of 2 is used for the 5 most recently de-queued packets. The formula of node delay calculation is presented in Eq. (10):

$$NodeDelay = \frac{\sum_{i=1}^5 queueingdelay(i) + \sum_{i=6}^{10} 2 \times queueingdelay(i)}{15} \tag{10}$$

In this equation, the queueing delay (i) is the difference between de-queuing instant and queueing instant of packet (i). Once a node has already de-queued ten packets, it uses a sliding interval where the oldest queueing delay is deleted and the new one is inserted. This helps to ensure that each node always records the time of the most recent ten de-queued packets. Node delay (d) is employed to calculate the path delay (D). For each node, D is the average time in

which packets are estimated to pass through this node to the sink. Sink 1-hop neighbours have the same value of node and path delays.

The results of the deduction above that are higher than 0.7 indicate node congestion. In this case, the node queue’s entry to exit rate should be evaluated to prevent congestion.

#### 4.4.2 Parent change mechanism through learning automata

The routing metric publishes and selects the next hop according to energy rate and node queue. Each node will also issue a beacon message to notify its neighbours of reaching the congestion threshold. Therefore, neighbours connected to the congested node will not select it as the next hop until the next routing schedule. After congestion, a beacon is sent to neighbours to bring the node’s chances of taking part in routing to the state before congestion. This method compares queues according to entry and exit rates per second.

Condition One:  $\left(\frac{\lambda\Delta t}{\mu\Delta t}\right) \leq 1$

Condition Two:  $\left(\frac{\lambda\Delta t}{\mu\Delta t}\right) > 1$

The congestion probability is low in Condition 1 and high in Condition 2. In Condition 2, the difference between the queue input and output rate will be the factor used for comparison. If the  $\lambda\Delta t - \mu\Delta t$  difference is smaller than the value of slots remaining in the queue, the node will not send any beacons. If, however, the difference is bigger than the slots remaining in the queue, the node will send a beacon to attract half the traffic. If they encounter a queue buffer overflow again, a beacon will be sent to neighbours and they will be asked to reduce their traffic by half. The total reduction rate will usually be 75%. After the network congestion problem is resolved, the node changes its reception rate to the previous value and sends beacons to its available children to notify them of the situation.

**4.4.2.1 Learning automata** A learning automaton can be considered a single object that has a finite number of actions. The learning automata work by selecting and applying an action from the collection to the environment. The action is evaluated through a stochastic environment, and the automata use the environment’s response to select its next action. The automata learn to select the optimal action through this process. Using the environment’s response to the action selected by the automata for selecting the next action is specified using automata’s learning algorithm. A learning automata consists of two main parts:

1. A stochastic automaton with a limited number of actions and a stochastic environment which the automata is connected to.
2. The learning algorithm, which the automata use to learn the optimal action.

*Stochastic automata* A stochastic automata is defined as the  $SA \equiv \{\alpha, \beta, F, G, \varphi\}$  quintuplet, including the  $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  collection ( $r$  number) of automata actions, the  $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$  automata input collection, the  $F \equiv \varphi \times \beta \rightarrow \varphi$  new condition generation function, the  $G \equiv \varphi \rightarrow \alpha$  output function which writes the current condition to the next output and the  $\varphi(n) \equiv \{\varphi_1, \varphi_2, \dots, \varphi_k\}$  collection of internal automata states in  $n$  moment.

The  $F$  and  $G$  functions write the current input condition to the automata's next output (action). The automata is a stochastic one if  $F$  and  $G$ 's writings are random.

This collection, along with the learning algorithm, is known as the stochastic learning automata. The stochastic learning automata can therefore be shown with the  $LA \equiv \{\alpha, \beta, p, T\}$  quadruplet, where  $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  is the collection of automata actions ( $r$  is the number of automata actions),  $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_r\}$  is the automata input collection,  $p \equiv \{p_1, p_2, \dots, p_r\}$  is the automata action probability vector and  $T \equiv p(n+1) = T[\alpha(n), \beta(n), p(n)]$  is the learning algorithm.

After making the parent collection accessible to each child, the node selects the highest ranking from its list, which is the node with the maximum pheromone and starts to send. Each node compares the amount of energy ratio and their new queue, which is the value for the  $D_n$  node, with  $D_o$  as its previous rate value after the  $t$  time period. According to the automata formula, if change is higher than the threshold; it indicates the possibility of an increase, stability, or reduction. Automata will be entered if the condition is met. One action is selected randomly from the collection.

The return node's multi-hop remaining energy ( $\mathcal{E}(n)$ ) is proposed to approximate remaining energy, which investigates the connection between the receiver of the DIO message and the route energy of the DODAG's three return parent nodes.

$$\epsilon(n) =$$

$$\begin{cases} \frac{E_{init}(n) - E_{cur}(n)}{E_{init}(n)}, n = \text{root} \\ \text{Max}\left(\left(\frac{\epsilon_{n-1} + \epsilon_n - 2 + \epsilon_{n-3}}{3}\right) * \theta\right), \frac{E_{init}(n) - E_{cur}(n)}{E_{init}(n)} n \neq \text{root} \end{cases} \quad (11)$$

Where  $n$  is the current node,  $E_{init}(n)$  is the initial energy level and  $E_{cur}(n)$  is the current energy level for the  $n$  node.  $E_{init}(n) - E_{cur}(n) / E_{init}(n)$  indicates the remaining energy for the node  $n$ .  $\mathcal{E}(n)$  is the condition of the node chain's remaining energy in the route. In other words, the remaining

energy of nodes is considered in return, while the effect of the parent's remaining energy is reduced as its route goes lower.  $\Theta$  was assumed to be 0.20.

In our computation method, the significant factor for the proposed method is  $\mathcal{E}(n)$ , which distinguishes our ranking method from others. This factor gives the multi-hop return information to the ranking calculation equation. If the protocol doesn't consider the previous parent's conditions, it may select a parent that has congestion problems on its route to root, even if the selected parent is in good condition in terms of remaining power and buffer. This paper has used multi-hop parent information to shift the focus from just one parent condition to the parent's multi-hop chain condition leading to a general view of the node's conditions for conversion to a parent.

The next important metric for selecting the efficient parent is the buffer proposed as follows. The buffer size is evaluated through the following formula.

$$Q(n) = \begin{cases} Q(n), n = \text{root} \\ \text{Max}\left(\left(\frac{Q_{(n-1)} + Q_{(n-2)} + Q_{(n-3)}}{3}\right) * \theta\right), Q(n) n \neq \text{root} \end{cases} \quad (12)$$

The same node's buffer size, as well as the one for its three previous parents, is obtained (through the DIO message). Then, the average of the three previous parents will be multiplied by  $\Theta$  to reduce the upstream parent's impact. Then, the maximum value between that and the node's buffer is obtained.  $\Theta$  is 0.20 actions in the automata: 1-decreasing probability, 2-increasing probability and 3-fixed probability.

A) Optimal answer form the environment

$$\begin{cases} p_i(n+1) = p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) = (1 - a)p_j(n) \forall j, j \neq i \end{cases} \quad (13)$$

B) Suboptimal answer form the environment

$$\begin{cases} p_i(n+1) = (1 - b)p_i(n) \\ p_j(n+1) = \frac{b}{r-1} + (1 - b)p_j(n) \forall j, j \neq i \end{cases} \quad (14)$$

The automaton randomly selects an action from the collection and applies it to the environment. According to the reaction received, that action will be rewarded, while the other two will be penalized.

#### 4.4.3 Load balancing and solving the bottlenecking problems through a moving node

According to the random structure of the position of nodes in the proposed RPL, network congestion is undeniable since differing number of nodes request from other network nodes. In other words, each parent node may receive an unpredictable number of requests from its children. This will lead to

increased traffic and as a result, generate congestion in the network's nodes. In this regard, an approach for using moving nodes has been proposed in another section of this study, in which a number of nodes will be GW or gate nodes. These nodes have a higher radio range and energy than other nodes in the network and will use a moving unit in the network. These nodes enter the network through the working nodes process and if necessary, assume the role of parent nodes for high congestion children.

In order for moving and normal nodes to cooperate in the network, changes should occur in the network's graph structure, which will be explained.

**4.4.3.1 Prioritizing and managing the moving nodes using the flabellum algorithm** Being inspired by a phenomenon and exploiting special knowledge provided by the problem, heuristic algorithms explore the complicated optimization problem space and offer a sufficiently good (optimum) solution. In the present research, given the biological and physical behaviour of flabellum's movement in an artificial system, optimization algorithm in continuous space is presented. The physics of movement, group subset behaviour and flabella's death indicate that these organisms are intelligent and attempt to hunt for a prey and survive while interacting with each other. Intelligence enables them to efficiently use the water flow and wind flow to reach their target (formation of group subset behaviour). In our proposed optimization algorithm, the ocean surface is considered as the problem space and sensing tentacles as the information exchange tool. According to the organism's behavioural approach in biology, wind power, water flow and swimming are three factors affecting the searching process in the problem space.

Typically, flabellum optimization algorithm is implemented in two general steps:

- Forming an artificial system with continuous time in the problem space; initial positioning of agents; determining the fit of toxins; and specifying the strategy of moving with wind power and water flow.
- Updating the movement and parameters during algorithm implementation phases

**4.4.3.2 Forming the system** At the beginning of any heuristic optimization algorithm, the problem space is defined. The problem space is a multidimensional coordinate system in which searching for an optimum solution occurs. In flabellum optimization algorithm, the ocean surface is considered as the problem space on which search agents (a group of flabella) are placed. Each agent in the problem space has the following features:

- The position of each flabellum with the sensing radius
- Movement with wind flow and water flow
- Flabella's amount of toxins (the rate of fit)

The position of each agent in the search space indicates a solution of the optimization problem. All positions in the problem space have the neighborhood sensing radius communication medium. The competency of each agent depends on its location on the target function.

Global optimization strategy in the problem space is such that the best location found by the search agent on the target function is regarded as the global optimum. The purpose of this strategy is to describe the ocean shore so as to direct the search agents towards that path by the wind power.

While search agents are directed towards the global optimum in all states, the agents' movement during the occurrence of three states to form a group behavior is considered as the local optimum controlled by water flow and the organism's swim.

If we consider the system as a group of flabella, in which a position is indicative of a point in the optimization problem space, then  $d$  denotes the position of the dimension and  $x_i^d$  the agents.

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \quad (15)$$

Once the position of each agent ( $x(t)$ ) is randomly determined on the problem space, the agents' rate of fit (the concentration of toxins)  $fit_i(t)$  is evaluated based on their location on the problem space. In order for an agent to change its current location to a new one  $x(t+1)$ , it requires a velocity vector. The velocity vector of an agent changes from the position  $V(t)$  to the next position  $V(t+1)$  by wind and water powers.

$$F(t+1)_i^d = \alpha_i(t) \times V(t) \times rand(G_{best} - X_i(t)) \quad (16)$$

where  $V$  is wind power whose constant value is 2 ( $v=2$ ) and  $\alpha$  is the effective coefficient of the agent  $i$  by wind power, which can be adjusted in the range  $[0.1 < \alpha < 0.9]$  proportional to the amount of water power  $V$  but is always considered constant.  $Rand$  is a random number with uniform distribution in the range  $[0,1]$ .  $G_{best}$  is the best location found by an agent.

In response to the three occurrences, the local optimum, combined with the global optimum, completes the movement strategy of the next step. In this state of the system, the water power is imposed on the agent  $i$  as  $\overline{F}(wind)_i^d(t)$  at time  $t$  to the dimension  $d$  towards the local optimum (formation of subset behavior) in three states.

**4.4.3.3 The first state** If there is a neighborhood in the sensing radius, whose fit is the best compared to the current agent  $i$ , it moves one step towards that neighborhood and this power is calculated as follows.

$$F_i^d(t+1) = \beta_i(t) \times U(t) \times \text{Sin}(\text{rand}) \left( L_{\text{neighborhood}_{\text{best}_i}} - X_i(t) \right) \tag{17}$$

where  $U$  is water power with constant value 2 ( $U = 2$ ) and  $\beta$  is the effective coefficient of the agent  $i$  by the water power, which can be adjusted in the range  $[0.1 < \beta < 0.9]$  proportional to the amount of water power  $U$  but is always considered constant.  $\text{Rand}$  is a random number with uniform distribution in range  $[0, 1]$ .  $L_{\text{neighborhood}_{\text{best}_i}}$  is the best agent in terms of competency in the agent's neighborhood.

**4.4.3.4 The second state** If there is no better neighbourhood for the agent  $i$  in the sensing radius, the agent moves one step towards its own personal memory, whose value in this state is calculated as follows

$$F_i^d(t+1) = \beta_i(t) \times U(t) \times \text{Sin}(\text{rand}) \left( L_{\text{neighborhood}_{\text{best}_{\text{mem}_i}}} - X_i(t) \right) \tag{18}$$

$L_{\text{neighborhood}_{\text{best}_{\text{mem}_i}}}$  is the best personal memory of the agent  $i$ .

**4.4.3.5 The third state** If the agent  $i$  has no personal memory for the second state, then it randomly moves one step, whose value is calculated as follows.

$$F_i^d(t+1) = \beta_i(t) \times U(t) \times \text{rand}() \tag{19}$$

The agent  $x(t)$  with a sensing radius has evaluated the competency of its surrounding agents. If no agent in the neighborhood of agent  $x(t)$  has better competency, or if the neighborhood of agent  $x(t)$  is vacant and agent  $x(t)$  has no personal memory on the other hand, then agent  $x(t)$  randomly moves one step. The next step of agent  $x(t)$  is towards the agent  $G_{\text{best}}$ . The power that is generally imposed on the agent  $i$  is the result of wind and water powers expressed as follows.

$$F_i^d(t) = \alpha_i(t) \times V(t) \times \text{rand}(G_{\text{best}}) + \beta_i(t) \times U(t) \times \text{rand}(L_{\text{best}}, M_{\text{best}}, \text{rand}) \tag{20}$$

According to Eq. 16, wind power is generated in all three states of occurrence (wind power directs one step towards the shore at each stage). In contrast to each three states of occurrence, agents show resistance against wind power in order to form a group subset behavior.

The best neighborhood in the three states of occurrence (local optimum) is selected as the best according to the competency of the neighborhood. Competency can be changed depending on the type of the optimization problem (minimization, maximization) for the best neighborhood. The best competency for the global optimum in this algorithm is considered as the highest competency among the community members. In other words, the agent with more competency, in the community distributed in the problem space,

is selected as the global optimum. The global optimization strategy of this algorithm is to direct the agent towards the shore and its death. Therefore, an agent with the highest competency is a global optimum directed towards the shore.

**4.4.3.6 Updating** As the system forms, all flabella are randomly spread over the problem space. The fitness of their location is evaluated every moment and for each flabellum displacement, Eqs. 16 to 19 are calculated, then it is placed in the next location.

Flabellum algorithm parameters include the competency of each agent's toxin, cross-sectional coefficient of sail ( $\alpha$ ) and the impact factor of water power ( $\beta$ ). It should be noted that the wind power is a constant value; hence, increasing it affects the movement velocity of the agent. Of course, how it is influenced can be controlled by  $\alpha$ . The lesser the amount of  $\alpha$  is, the more the movement velocity of the agent with the wind power is reduced. If we consider a higher value for water power impact factor  $\beta$ , it increases the movement velocity of the agent since the water flow rate is constant. The lesser the amount of  $\beta$  is, the more the movement velocity of the agent is reduced. Wind power reinforces the algorithm exploration strength, while water power guarantees the algorithm efficiency.

Algorithm 1: Pseudocode of the proposed optimization algorithm.

*Pseudocode of the flabellum algorithm*

---

1. Determining the system space and initial valuation
2. Determining the initial position of the flabellum
3. Evaluating the position
4. Updating  $\alpha$  and  $\beta$  parameters
5. Calculating the wind and water powers for each flabellum
6. Computing the effect of wind and water powers velocity of each flabellum
7. Updating the positions of flabella
8. In case the algorithm termination condition was not met, it returns to Step 3; otherwise, the best solution found in the output is returned and the algorithm terminates.
9. End.

Algorithm 1: Pseudocode of the proposed

First, changes will be made to the RPL network's CC message structure, for designing a new a message could generate extra control overhead on the network. For this purpose, each network node will send the CC message to everyone in case of congestion in their queues or an increase in the energy depletion rate, which will lead to the node's early death. Normal network nodes can use this message to reduce the send rate to the node; however, its main goal is to inform the existing moving nodes of the node's radio range for cooperation. In other words, the network's congested nodes request for help from the moving nodes available in their proximity and send periodic messages to inform them. The network's moving nodes lack GPS and are moved randomly in the network.

Each of the network's moving nodes move randomly after receiving the CC message from the sender and detect whether it is getting close or moving away. This process will be rewarded if the signal strength is increased by moving from point  $x_1$  to point  $x_2$  and it will get closer to another hop. A DAO message is sent to it after it is placed in the node's full radio range and the list of current children will be received from the node in the DAP-Ack response. After message reception is verified by the congested node, the moving node will assume the role of the congested parent node and receive messages from the other node's children and the node itself and then transfer their messages to the node's parents.

## 5 Simulating and evaluating methods

Table 2 shows the proposed network simulation conditions. According to Table 2, there are 180 sensor nodes distributed randomly in the simulation environment and evaluated in these tests, and the extent to which the proposed and the similar base method are successful in energy efficiency per successful send in the network, which leads to dynamic routing, will be evaluated.

This paper proposes the automata-ant-colony multiple recursive RPL, abbreviated to AMRRPL, which is a modified version of RPL for the IoT networks providing a balancing model through the automata and ant colony algorithm as well as the multi-step recursive model. It prevents congestion by managing the moving nodes. As a result of load balancing and congestion prevention, it will reduce network energy consumption, prolong network lifetime, and reduce packet loss.

HECRPL (Zhaoa et al. 2017) is a distributed, reliable and energy-efficient routing protocol. In HECRPL, both wireless links lossy rates and energy consumption are taken into consideration to estimate the routing cost. In HECRPL, the routing decision is made with the purpose of minimizing energy consumption. HECRPL selects the optimal cluster-parent-set

(CPS) through a top-down approach in conjunction with transmission power-level selection.

HECRPL incorporates the following five major features to effectively prolong the lifetime of the network, while achieving a high reliability and fairness for the network: (1) a top-down approach for optimal cluster-parent-set (CPS) selection to minimize the energy depletion of the network and leverage path diversity, while achieving the global goal, (2) an overhearing-based coordination among nodes in a CPS to avoid duplicate transmissions, (3) the priority in a CPS is based on a hybrid of residual energy and the lossy conditions of wireless channels, (4) an efficient loss recovery scheme for the detection and retransmission of the lost data packets and (5) the transmission power is refined to increase the network capacity (increase spatial reuse) and further reserve energy. Simulation results demonstrate that HECRPL can significantly prolong the lifetime of the network and provide much more robust network connectivity than the benchmark. However, the spatial reuse feature cannot be exploited because of the timer-based priority scheduling for CPS coordination. In addition, congestion occurs easily in a resource-constrained network. An efficient load sharing scheme is necessary to alleviate the congestion issues.

E-RPL (Preeth et al. 2019), another paper that compared with the proposed method, has presented an energy efficient routing in the IoT by using the ACO. The proposed E-RPL considers multiple routing factors and selects an efficient parent node to build an optimal DODAG structure. The E-RPL consists of the ACO based multi-factor optimization for parent selection and coverage-based dynamic trickle algorithm for energy efficient DODAG construction without compromising network coverage and reliable data routing. The ACO considers the expected transmission count (ETX) and rank value as pheromone factors, whereas the residual energy and children count as heuristic factors. The E-RPL exploits the parent-child relationship factor as a pheromone evaporation factor to balance the conflicting factors of ETX, rank, delay, and energy consumption. Moreover, the weight-based algorithm is utilized to combine pheromone, heuristic, and pheromone evaporation factors towards

**Table 2** Simulation conditions

Value	Parameter
180 node	Number of nodes
240 m*240 m	Network environment
30 to 50 streams per minute	Network traffic rates
30 m	Node radio board
5 J	Primary energy of the node
CBR	Traffic type
200 s	Simulation time

a single-objective function. To develop an optimal DODAG structure with a reduced routing overhead, the E-RPL introduces concentric corona-based network partition and determines the value of broadcast count dynamically concerning the node density and coverage. The use of ETX as well as rank value effectively balances the routing performance and energy consumption. Moreover, the consideration of connected children and the remaining energy in the parent selection reduces the collision impact and ensures the reliable packet delivery; however, it needs to be evaluated under various network size to show the routing scalability.

### 5.1 Probability of reaching $I_{max}$

The ultimate goal of generating the drop scheduler for the RPL is to reach maximum network stability. This test investigates the probability of achieving this goal in simulation time. One of the most important reasons for resetting the drop scheduler is the network graph's instability and bottlenecks or un-parented nodes in the network. In terms of delivery rate, such un-parented or unstable nodes periodically send a DIS message to find a better parent. These parent change requests will reduce the node's probability of reaching  $I_{max}$  in the network and will repeatedly reset network nodes. Effective parameters were considered in the proposed AMRRPL method for forming the intended link in order to prevent this problem. Since the proposed network considered the LQI and SNR metrics the main factors for forming the link, the results of this test present significant improvements compared to other methods, according to Fig. 4. Avoiding greedy parent node selection and considering a combination of desired parents in multiple hops are among other features that reduce network bottleneck and increase graph node stability.

In the proposed model's scenario, the network was considered to have an average range (number of nodes, radio range and scheduler). Simulation results show that the multi-hop link management learning system and congestion management, the proposed ant colony method for the network graph dynamics was able to achieve a higher probability of network stability, or a better convergence time, than other methods (Fig. 4). The effect of selecting suitable children in the AMRRPL method should not be ignored either, which had better performance compared to the HECRPL and ERPL methods and sped up network stability.

### 5.2 Network lifetime test

The network lifetime test is performed to evaluate the network's effectiveness in saving energy for active nodes. Many studies have considered the time of death for the first and the middle nodes of the network as the main factor for network assessment. The more unbalanced the network's energy

consumption, the faster this event will occur. Energy efficiency was the goal of the network proposed in this study and an attempt was made to consider link quality and the parent's condition to prevent the premature death of the network's nodes as much as possible and delay the death of the network's first node. Figure 5 shows the test results for two traffic criteria, 30 and 50 packets in unit of time for the base and proposed methods.

Generating a high quality link, knowledge-based parent selection and taking node energy parameters into account in the decision system's computations have reduced the probability of selecting a low energy node in the network in the proposed method; which delays the death of the network's first node. With the traffic flow of 30 packets, the rate of lost nodes in the proposed network is approximately halved after 100 s, while increasing to 75% in the base method. After 200 s, due to loss of connection between the network's root and farther away leaf node, the energy consumption rate is only for sending the DIS message. Increasing the network's packet production traffic rate to 50 units in the proposed method increases the number of packets lost in the network to 100. Increasing the network traffic rate or reducing the link's success probability in the network during experiments will sharply reduce the number of live nodes in the network. At the same time, energy efficient routing and queue management will reduce the node's energy cost. According to the results, the proposed method has outperformed the base method by approximately 9.34 percent in the 30 traffic and 11.16% in the 50 traffic.

### 5.3 Testing energy consumption rate for variable traffic

The average network energy consumption test was proposed for measuring the routing pattern's impact on network

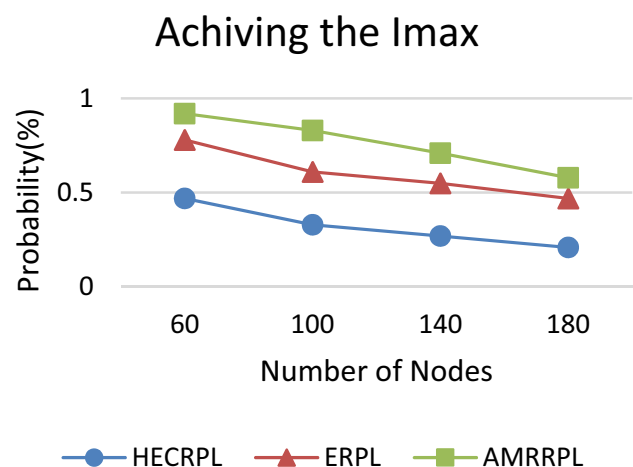


Fig. 4 Test results of the probability of reaching  $I_{max}$

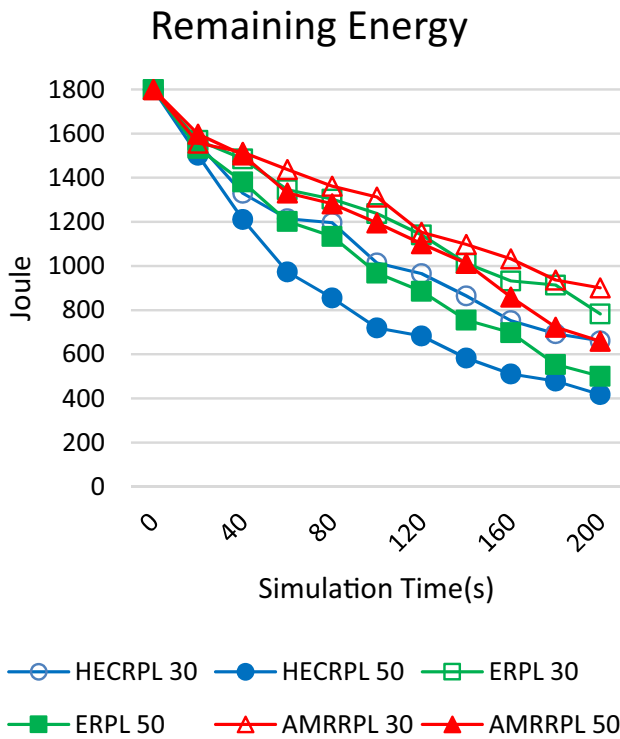


Fig. 5 Network lifetime for 30 and 50 current packets in unit of time

energy consumption. Figure 6 shows the results of this test, in which the proposed AMRRPL algorithm was able to outperform the HECRPL and ERPL base methods due to considering the ant colony algorithm for generating the optimal route and involving node energy and queue parameters causing processing, queuing, and publishing delays as well as the upstream parent’s accessibility.

Figure 6 has taken the average node energy consumption in the network’s traffic ranges into consideration, which indicates the proposed AMRRPL method’s awareness of network energy and condition. Overtime, simulating this process will reduce the proposed network’s energy consumption. The average network energy consumption improvement compared to the base method is approximately 15 percent for 20 traffic and 20% for 30 traffic. The difference was somewhat reduced in 40 traffic, reaching 12%, whereas it increased once again to 15% for 50 traffic. Finally, both graphs of the proposed and base methods had a minimum 8% difference in this test with 60 traffic. The average improvement obtained was 14 percent. The proposed network’s improvement in variable traffic is often the result of the dynamic radio range of network nodes preventing the generation of lossy links and fixing or completely disconnecting links whose energy consumption rates are higher than the obtained value. The simulation results show that the decision does not hinder the network’s data transactions and could provide more optimal

distribution. In other words, generating safe and stable links reduces noise and crosstalk and prevents packet losses in the network. Finally, the aforementioned points increase the RPL network’s energy efficiency.

### 5.4 Control overhead rate test

Since the major part of the working time of RPL-based protocols is devoted to exchanging controlling messages including DODAG Information Object (DIO), Destination Advertisement Object (DAO), DODAG Information Solicitation (DIS), DAO-Ack, Beacon, Clear Channel Assessment (CCA) and CC, the more these exchanges are decreased while maintaining the network stability, the higher the network efficiency and network working time will be. In the proposed method, the exchange rate of controlling messages is decreased due to the formation of stable and multi-path graphs. Moreover, by decreasing the swarm rate and the overflow of network node queues, the number of beacons sent to reduce and adjust the packet delivery rate from child to parent is reduced. In this method, by taking into account the rate of network end-to-end delay, nodes encounter lesser bottlenecks compared to the basic method and are able to know the traffic rate and continuous flow in the network. Also, optimum use of moving nodes’ mechanism contributes in generating load balance in the network and the results are improved by 20% in this test compared to other methods. Figure 7 has taken the Control overhead rate into consideration,

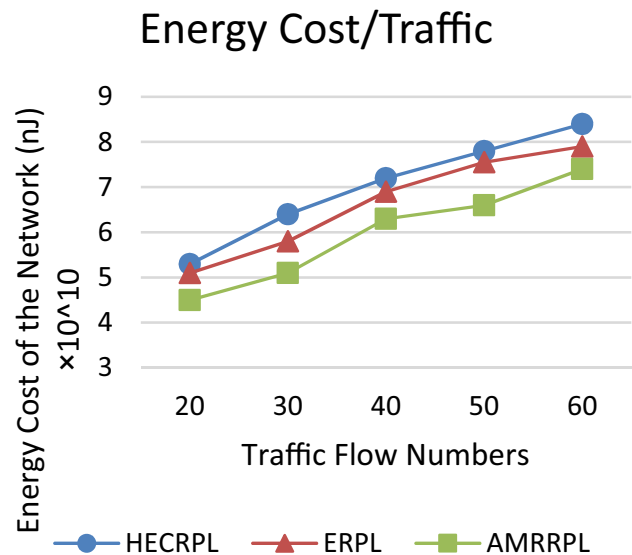


Fig. 6 Energy consumption in variable current traffic



### 5.5 Delivery rate test with various movement patterns

However, the main challenge of delivery rate in the RPL is in heavy traffics which can cause bottlenecks in the network and disturb its delivery rate. In this test, three movement models are selected for moving nodes in the network. In the first model, moving nodes move in the network based on a random pattern and according to the CC message received from nodes involved in the swarm, they try to help them. In the second model, the movement pattern of moving nodes in the network is determined by using Tabu search algorithm, and the final pattern is to use the flabellum algorithm. The results indicate that the proposed flabellum method could yield a higher, acceptable delivery rate compared to the other two methods.

Figure 8 has taken the results of packet delivery rate test with various movement patterns into consideration.

### 5.6 Scheduler reset probability test in heterogeneous network

This parameter was included in order to study the feasibility of the proposed method to retain the current drop scheduler. The lesser the event occurs in network nodes, the lesser DIO messages are sent in the network and the more the network efficiency is increased. The use of moving nodes in stabilizing network graph and generating load balance can reduce unwanted resets in drop scheduler in the network. According to the test results, the flabellum movement management method has outperformed random movement and Tabu search methods. Figure 9 has taken the test results of drop

scheduler reset rate with various movement patterns into consideration. At the end Figs. 10 and 11 have taken The test results of packet delivery rate and The test results of the total energy consumption of the network in different stages of the proposed algorithm (3 phases: ant colony based OF, automata based parent selection, flabellum based moving node management) into consideration.

## 6 Conclusion

This paper evaluated the RPL problems under heavy and dynamic load by focusing on packet loss and network lifetime. It is discovered that the RPL standard cannot effectively manage and balance heavy loads and dynamic loads.

This paper proposed the automata-ant colony multiple recursive RPL, abbreviated to AMRRPL, which is a modified version of the RPL for the IoT networks to develop a balancing model by using the automata and ant colony algorithm as well as the multi-step recursive model. It prevents congestion by managing the moving nodes. As a result of load balancing and congestion prevention, it will reduce network energy consumption, prolong the network lifetime, and reduce packet loss.

This protocol was presented in three steps. The first step was to evaluate the condition of a multi-hop parent chain before selecting the last one as the node's selected parent. Therefore, there was an attempt to balance the network's load and to improve the network lifetime and energy important factors, including LQI, SNR, and buffer as well as remaining energy level taken into consideration. The ant

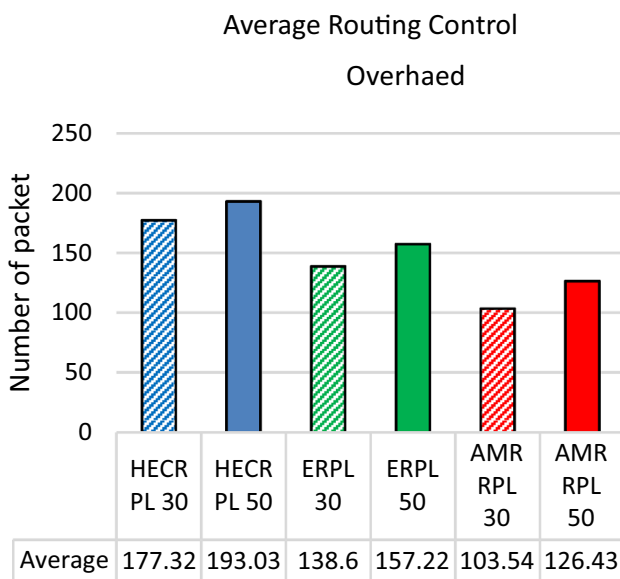


Fig. 7 Network control overhead rate test

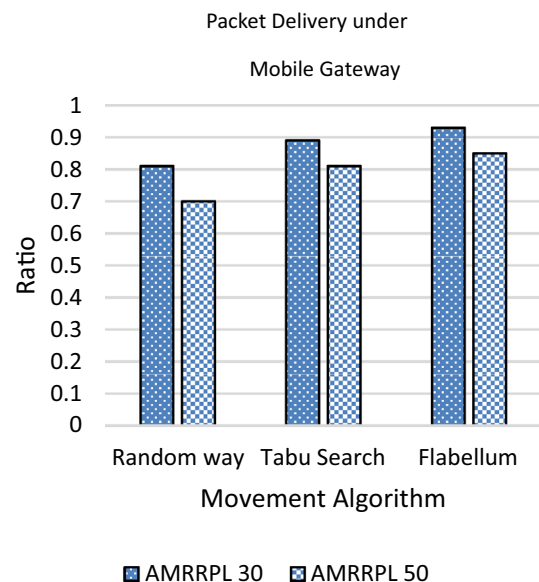


Fig. 8 The results of packet delivery rate test with various movement patterns

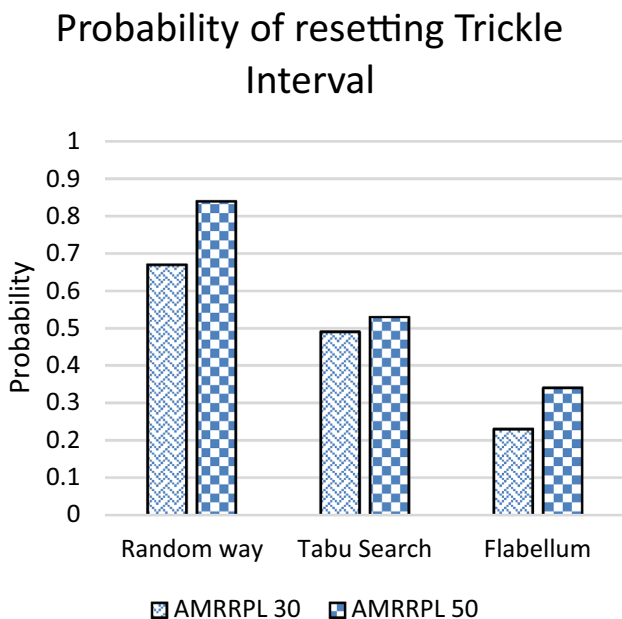


Fig. 9 The test results of drop scheduler reset rate with various movement patterns

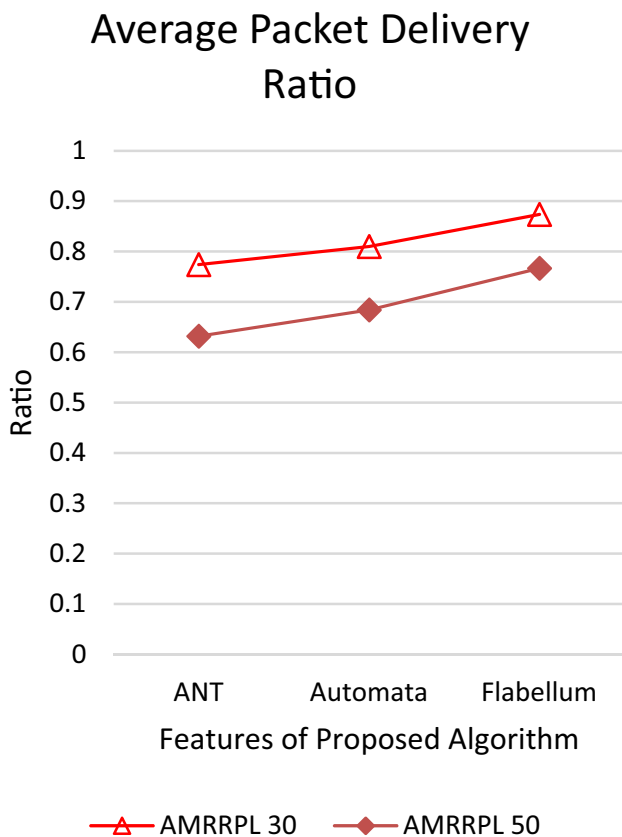


Fig. 10 The test results of packet delivery rate in different stages of the proposed algorithm

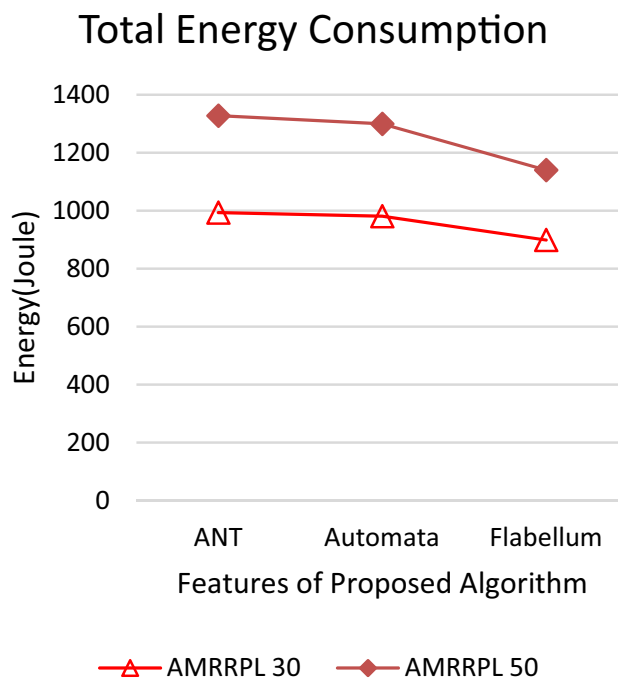


Fig. 11 The test results of the total energy consumption of the network in different stages of the proposed algorithm

colony algorithm was used for rank calculation. The second step proposed an algorithm to select the parent dynamically and with awareness of node conditions by using stochastic automata.

The comprehensive evaluation showed that this algorithm made better decisions on the proper parent selection in a network with high traffic dynamism rather than making decisions merely based on the parent’s rank. The third step resolve bottlenecks and swarm problems by managing the moving nodes through the heuristic flabellum algorithm inspired by physical and biological behaviour of flabella in the sea.

The proposed method was evaluated under different scenarios in Cooja, proving that the AMRRPL outperformed existing algorithms with regard to packet delivery, energy consumption rates, and network lifetime and stability through the load balancing model and congestion prevention.

**Author contributions** ZR have idea and ZR and HM and AKB analysis and simulate the paper. All authors read and approved the final manuscript.

**Funding** No sources of funding for the research.

**Availability of data and materials** Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

## Compliance with ethical standards

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

## References

- Aljarrah E (2017) Deployment of multi-fuzzy model based routing in RPL to support efficient IoT. *Int J Commun Netw Inf Secur* 9(3):457–465
- Atzori L, Iera A, Morabito G (2010) The internet of things: a survey. *Comput Netw* 54(15):2787–2805
- Banh M, Nguyen N, Phung T-K.-H, Nguyen Thanh L, Thanh N. H, Steenhaut K (2016) Energy balancing RPL-based routing for Internet of Things. In: Sixth international conference on communications and electronics (ICCE), Ha Long, Vietnam, July 27–29, pp 125–130
- Barbato A, Barrano M, Capone A, Figiani N (2013) Resource oriented and energy efficient routing protocol for ipv6 wireless sensor networks. In: Proceedings of the IEEE online conference on green communications, Piscataway, NJ, USA, 29–31 Oct, pp 163–168
- Capone S, Brama R, Accettura N, Striccoli D, Boggia G (2014) An energy efficient and reliable composite metric for RPL organized networks. In: Proceedings of the 2014 12th IEEE international conference on embedded and ubiquitous computing (EUC); Milano, Italy, 26–28 August, pp 178–184
- Dohler M, Watteyne T, Winter T, Barthel D (2009) Routing requirements for urban lowpower and lossy networks. IETF Secretariat; Fremont, CA, USA, RFC 5548
- Gao W, Sarlak V, Parsaei M, Ferdosi M (2017) Combination of fuzzy based on a metaheuristic algorithm to predict electricity price in an electricity markets. *Chem Eng Res Design* 131:333–345
- Gnawali O, Levis P (2010) The ETX objective function for RPL document draft-gnawali-roll-etxof-00, Working Draft, IETF Secretariat, InternetDraft
- Gnawali O, Levis P (2012) The minimum rank with hysteresis objective function. RFC 6719. <https://doi.org/10.17487/RFC6719>
- Gozuacik N, Oktug S (2015) Parent-aware routing for IoT networks. Internet of things, smart spaces, and next generation networks and systems. Springer, Cham, pp 23–33
- Halder S, Kim W (2012) A fusion approach of RSSI and LQI for indoor localization system using adaptive smoothers. *J Comput Netw Commun*. <https://doi.org/10.1155/2012/790374>
- Hassan A (2016) Improved routing metrics for energy constrained interconnected devices in lowpower and lossy networks. *J Commun Netw* 18(3):327–332
- Iova O, Theoleyre F, Noel T (2013) Stability and efficiency of RPL under realistic conditions in wireless sensor networks. IEEE Int Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), London
- Iova O, Theoleyre F, Noel T (2015) Using multiparent routing in RPL to increase the stability and the lifetime of the network. *Ad Hoc Netw* 29:45–62
- Kamgueu PO, Nataf E, Ndié TD, Festor O (2013) Energy-based routing metric for RPL [Research Report] RR-8208, INRIA:14. <https://doi.org/10.1109/lcnw.2015.7365929>
- Karkazis P (2013) Evaluating routing metric composition approaches for QoS differentiation in low power and lossy networks. *Wirel Netw* 19(6):1269–1284
- Khan MM, Lodhi MA, Rehman A, Khan A, Hussain FB (2016) Sink-to-sink coordination framework using RPL: routing protocol for low power and lossy networks. *J Sens* <https://doi.org/10.1155/2016/2635429>
- Khelifi N, Oteafy S, Hassanein H, Youssef H (2015) Proactive maintenance in RPL for 6LoWPAN. In: Wireless communications and mobile computing conference (IWCMC), International, IEEE, pp 993–999
- Kim H (2015) A measurement study of TCP over RPL in low-power and lossy networks. *J Commun Netw* 17(6):647–655
- Kim H, Bang J, Lee Y (2014) Distributed network configuration in large-scale low power wireless networks. *Comput Netw* 70:288–301
- Kulkarni P, Gormus S, Fan Z (2012) Tree balancing in smart grid advanced metering infrastructure mesh networks. In: IEEE international conference on green computing and communications (GreenCom):109–115.
- Kushalnagar N, Montenegro G, Schumacher C (2007) IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement and goals. IETF RFC 4919. <http://www.ietf.org/rfc/rfc4919.txt>
- Mamdouh M, Elsayed K, Khattab A (2016) RPL load balancing via minimum degree spanning tree. In: 12th international conference on wireless and mobile computing, networking and communications (WiMob), IEEE, New York, pp 1–8
- Mayzaud A, Badonnel R, Chrismet I (2017) A distributed monitoring strategy for detecting version number attacks in RPL-based networks. *IEEE Trans Netw Serv Manage* 14(2):472–486
- Mittal M, Tanwar S, Agarwal B, Mohan Goyal L (2019) Energy conservation for IoT devices: concepts. Paradigms and solutions. Springer, Singapore, p 206
- Mohamed B, Feham M (2015) QoS routing RPL for low power and lossy networks. *Int J Distrib Sens Netw* 2:1–10. <https://doi.org/10.1155/2015/971545>
- Nabaei A, Hamian M, Parsaei M, Safdari R, Samad-Soltani T, Zarrabi H, Ghassemi A (2018) Topologies and performance of intelligent algorithms: a comprehensive review. *Artif Intell Rev* 49:79–103
- Parsaei M, Javidan R, Sobouti M (2016) Optimization of fuzzy rules for online fraud detection with the use of developed genetic algorithm and fuzzy operators. *Asian J Inform Technol* 15(11):1856–1864
- Preeth SKSL, Dhanalakshmi R, Kumar R, Si S (2019) Efficient parent selection for RPL using ACO and coverage based dynamic trickle techniques. *J Ambient Intell Human Comput*. <https://doi.org/10.1007/s12652-019-01181-w>
- Riker A, Curado M, Monteiro E (2017) Neutral operation of the minimum energy node in energy-harvesting environments. In: 2017 IEEE symposium on computers and communications (ISCC), pp 477–482
- Sebastian A, Sivagurunathan S (2018) Load balancing optimization for RPL based emergency response using Q-learning. *MATTER Int J Sci Technol* 4(2):74–92
- Shakya N, Mani M, Crespi N (2017) SEEOF: smart energy efficient objective function: adapting RPL objective function to enable an IPv6 meshed topology solution for battery operated smart meters. In: Proceedings of the 2017 global internet of things summit (GloTS), Geneva, Switzerland, pp 6–9
- Thubert P (2012) Objective function zero for the routing protocol for low-power and lossy networks (RPL). IETF RFC 6552, Mar IETF RFC 6552, March
- Tong M, Chen Y, Chen F, Wu X, Shou G (2015) An energy-efficient multipath routing algorithm based on ant colony optimization for wireless sensor networks. *Int J Distrib Sens Netw* 11(6):64–79
- Tripathi J, Oliveira J (2013) Quantifying load imbalance: a practical implementation for data collection in low power lossy networks. In: IEEE 47th Annual Conference on Information Sciences and Systems (CISS), pp 1–6

- Winter T, Thubert P, Brandt A, Clausen T, Hui J, Kelsey R, Levis P, Pister K, Struik R, Vasseur J (2012) RPL: IPv6 routing protocol for low-power and lossy networks. IETF RFC 6550
- Yang Z (2017) CRB-RPL: a receiver-based routing protocol for communications in cognitive radio enabled smart grid. *IEEE Trans Veh Technol* 66(7):5985–5994
- Yang X, Guo J, Orlik P, Parsons K, Ishibashi K (2014) Stability metric based routing protocol for lo4-power and lossy networks. *IEEE ICC*, pp 3688–3693
- Yang Z, Han R, Chen Y, Wang X (2018) Green-RPL: An energy-efficient protocol for cognitive radio enabled AMI network in smart grid. *IEEE Access* 6:18335–18344
- Zhang T, Li X (2014) Evaluating and analyzing the performance of RPL in Contiki. *ACM proceedings of the first international workshop on mobile sensing, computing and communication*:19–24
- Zhaoa M, Han Joo P, Chanc H (2017) An energy-efficient and cluster-parent based RPL with power-level refinement for low-power and lossy networks. *Comput Commun* 104:17–33
- Zhi T, Hui Z (2015) An improved ant colony routing algorithm for WSNs. *J Sens Article ID* 438290.

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.