Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 9s (2025)

AM-RPL in a Hybrid IoT Network with Static and Dynamic Devices in Cooja Simulator

[1]Leelavathi R, [2]Vidya A

[1,2] Computer Science and Engineering Department, Vivekananda Institute of Technology, Bengaluru, India
[1]rajleelavathi@gmail.com, [2]Vidyaananth16@gmail.com

Article History:

Received: 12-01-2025

Revised: 15-02-2025

Accepted: 01-03-2025

Abstract:
Introduction: For instance, in a static IoT network where all sensor nodes are installed in precisely determined spots, it is possible that just one of the nodes will serve as a central hub for all data flowing through the network. As all traffic is redirected to this node, congestion at this node dramatically increases, quickly consuming the node's energy, creating a hole in the network. It's possible that the network will fail as a result of the failure of this node and the subsequent breakdown in communications. In order to overcome this problem, we proposed an algorithm Articulation Node Based Mobile Node Routing protocol for LLNs AM-RPL to avoid network failure. This algorithm first locates the articulation node in a given network, and it then gives the dynamic node instructions to move in that direction. A communication link is established as soon as a new node enters the communication range, updating the new DAG to include the dynamic node. This connects two components and shares articulation nodes load, improving network connectivity and performance. The proposed algorithm is implemented in the Cooja simulator to test the protocol's performance. The experiment's findings indicate that, when compared to a typical scenario, the PDR, Radio Duty Cycle, Charge Consumption, and Parent Switches of nodes all improve as load is distributed using dynamic nodes.

Objectives: This research aims to develop an efficient routing mechanism for Low-Power and Lossy Networks (LLNs) to address congestion-induced failures in static IoT networks. The **Articulation Node Based Mobile Node Routing Protocol (AM-RPL)** is proposed to dynamically identify articulation nodes and direct mobile nodes toward them, ensuring balanced network load distribution. By mitigating excessive energy depletion at central nodes, AM-RPL enhances network resilience, stability, and longevity. The study further evaluates the protocol’s effectiveness in optimizing key network parameters and explores its scalability across diverse IoT topologies for large-scale deployment.

Methods: To achieve the research objectives, AM-RPL is designed to **detect articulation nodes** and guide mobile nodes to strategically reposition themselves within the network. The protocol is **implemented and tested in the Cooja simulator**, where various IoT network scenarios are created to assess its performance. Metrics such as **Packet Delivery Ratio (PDR), Radio Duty Cycle, Charge Consumption, and Parent Switches** are recorded and analyzed to determine improvements over conventional static routing protocols. A comparative analysis is conducted to evaluate AM-RPL’s effectiveness in reducing congestion, improving energy efficiency, and maintaining stable communication links.

Results: The simulation results indicate that AM-RPL significantly enhances **network stability and performance** by reducing congestion at critical nodes. The introduction of mobile nodes enables **dynamic load balancing**, leading to improvements in **PDR, energy consumption, and parent selection efficiency**. Additionally, the protocol demonstrates better adaptability to varying network conditions, making it a viable solution for **large-scale IoT deployments**. While percentage improvements vary based on network scenarios, the overall trend consistently shows enhanced **reliability, energy efficiency, and sustained connectivity**, validating AM-RPL's effectiveness in overcoming the limitations of traditional static routing methods.

Conclusions: The deployment of the Articulation Node Based Mobile Node Routing protocol (AM-RPL) adeptly addresses the paramount challenge of network failure precipitated by congestion and energy depletion at pivotal central nodes in static IoT networks. By dynamically identifying articulation nodes and strategically directing mobile nodes to these loci, AM-RPL proficiently redistributes the load, thereby fortifying network resilience. The simulation outcomes within the Cooja environment reveal marked enhancements in Packet Delivery Ratio (PDR), Radio Duty Cycle, Charge Consumption, and Parent Switches. These results substantiate the effectiveness of the AM-RPL algorithm in preserving network stability and optimizing performance by mitigating congestion-induced failures.

Prospective endeavors may encompass the validation of the protocol across varied IoT network topologies and the pursuit of further refinements to augment its scalability and robustness. While we did not measure performance improvement in percentage terms due to varying readings under the same network conditions, AM-RPL reliably showed enhancements in reliability, energy efficiency, and network stability. This consistent improvement underscores the protocol's potential for optimizing large-scale IoT networks.

Keywords: IoT, Routing protocol, RPL, Mobile Nodes, Articulation Point, Energy Conservation, Cooja Simulator.

1. Introduction

The Internet of Things (IoT) represents a vast network of interconnected devices, encompassing not only computing devices but also objects, people, and animals. This network facilitates seamless data exchange through human-to-human, human-to-machine, and machine-to-machine communication [1][5][13][17]. The primary objective of IoT is to achieve automation across various processes with minimal human intervention. According to IDC, the number of connected devices worldwide is projected to surpass 40 billion by 2030, up from 14 billion in 2023 [2].

To realize this vision, both electronic devices and everyday objects are being integrated into the internet ecosystem. For example, in a smart home environment, appliances such as lights, fans, televisions, air conditioners, washing machines, and security systems are equipped with sensors to monitor environmental conditions. These sensors transmit data to a border router, which is connected to the internet, allowing for remote access and further processing of the collected information.

Selecting the optimal route for transmitting sensed data to the border router is crucial to ensure data integrity and avoid loss. Given the limited processing power, storage capacity, and energy resources of IoT nodes, these sensors cannot directly send data to the border router. Instead, data must traverse through intermediate nodes, making it imperative to choose routes with high-quality links and nodes to ensure reliable delivery.

The RPL (Routing Protocol for Low-Power and Lossy Networks) protocol, standardized by the IETF (Internet Engineering Task Force) in accordance with the specifications of the ROLL (Routing Over Low-Power and Lossy Networks) working group, is designed to address the requirements of low-power, memory-constrained, and processing-limited devices in IoT networks. RPL operates effectively in static networks where nodes are fixed at predetermined locations.

However, real-world scenarios often involve mobile devices, necessitating a revision of traditional routing protocols to accommodate dynamic connectivity. This adaptation is essential for maintaining robust communication in networks where nodes may frequently change their locations or connectivity statuses. A comprehensive survey in [reference] highlights the challenges and solutions associated with routing protocols for mobile IoT environments.

The literature extensively covers the integration of mobile nodes within static IoT networks and the strategies for facilitating data communication between static and dynamic nodes. One significant challenge in static networks is the risk of single points of failure, such as cut vertices or articulation nodes. These articulation nodes act as critical communication bridges between network components, and their failure can disrupt connectivity, as illustrated in Figure 1. Unlike dynamic networks, static networks cannot easily reconfigure to address such failures. To mitigate this issue, incorporating mobile nodes into static networks is proposed. The core concept involves relocating mobile nodes towards articulation nodes and distributing the load to ensure uninterrupted communication.

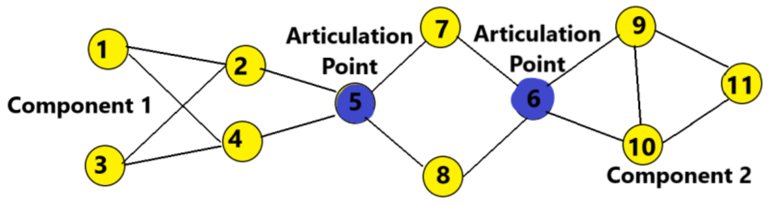


Fig 1: Random deployment of nodes in a network

In this work, our primary focus is on identifying articulation nodes within the network and determining the optimal movement direction for mobile nodes to optimize energy usage and extend the network's operational lifespan. Specifically, we aim to enhance network longevity, which is defined as the time elapsed until the first node depletes its energy. By strategically relocating mobile nodes to alleviate the load on articulation nodes, we seek to mitigate network failures and improve overall connectivity and performance.

The rest of the paper is structured as follows: Section II presents a review of the Literature, and Section III presents a Network Model followed by Mathematical Model in Section IV. The implementation of the suggested algorithm is described in Section V and VI. We analyse the simulation results in Section VII, and then in Section VIII, we conclude our work.

2. Literature Review

Paper [2] proposes Co-RPL, a Corona mechanism-based extension to RPL, to support node mobility in WSNs. A corona is a circular region with a radius r that is centred at the root of the DAG. The purpose of using corona coordinates is to localise each sensor node according to its proximity to the sink. Co-RPL maintains backward compatibility with the standard specification while reusing the RPL control messages. The authors used the Cooja simulator to show that Co-RPL reduces average energy consumption by 50%, packet loss ratio by 45%, and there is an end-to-end delay of 2.5s, compared to standard RPL.

In [3], the authors present several challenges encountered by the RPL due to the mobility of IoT devices. The authors then provided a comprehensive analysis of the available approaches to resolving RPL's problems, categorising them as "RSSI based solutions," "Trickle-timer based solutions," "Position-based solutions," "ETX based solutions," and "Miscellaneous solutions." In addition, they offer suggestions for further study of RPL's ability to accommodate a wide range of real-time applications and mobile support.

EM-RPL enhanced mobility support routing protocol reduces delay experienced during handoff and helps in extending the lifetime of LLN network [4]. The participant node selects the best data transfer parent using the two processes either mobile node parent selection, and static node parent selection. Thus, it selects the DODAG's best route and reduces the loss of packet during data transmission by mobile node.

Recent advancements in the RPL (Routing Protocol for Low-Power and Lossy Networks) have significantly enhanced the performance and adaptability of IoT (Internet of Things) networks. [5] present a thorough survey of RPL, emphasizing the protocol’s evolution and addressing critical challenges such as scalability and energy efficiency. Their work provides a comprehensive overview of recent advancements and identifies persistent issues that affect RPL’s performance in large-scale and diverse deployments [5].

In response to these challenges,[6] propose an innovative dynamic RPL-based routing protocol that improves network robustness by dynamically adjusting routing paths and parent node selections. Their approach demonstrates enhanced packet delivery ratios (PDR) and reduced end-to-end delays by actively responding to changes in network topology, making it more suitable for dynamic environments [6].

Work [7] further contribute to this field by reviewing recent enhancements to RPL, including sophisticated path selection mechanisms and energy-efficient routing strategies. Their review highlights advancements that optimize RPL for dynamic and resource-constrained environments, and outlines future research directions to address unresolved challenges and further improve protocol performance [7].

Author in [8] introduce a dynamic parent selection strategy within RPL that aims to enhance network reliability and performance, particularly in scenarios characterized by frequent topology changes. Their work focuses on optimizing parent node selection to improve connectivity and reduce network disruptions, thereby addressing some of the limitations associated with traditional RPL implementations [8].

Work [9] build on these advancements by proposing dynamic and adaptive routing strategies to enhance RPL’s performance. They focus on optimizing several key metrics, including energy consumption, latency, and throughput. Their approach aims to create a more resilient and efficient network by adapting routing decisions based on real-time network conditions and traffic pattern [9].

Author in [10] address the challenges of adapting RPL to highly dynamic IoT networks. They propose novel solutions to enhance protocol adaptability and efficiency, demonstrating significant improvements in network performance. Their work emphasizes the need for flexible and robust routing mechanisms to handle the dynamic nature of modern IoT applications [10].

Finally, [11] focus on optimizing RPL for environments with low power and high mobility requirements. They propose algorithms designed to enhance network efficiency and performance under specific operational conditions. Their work highlights the importance of tailored solutions for different network scenarios to maintain connectivity and performance [11].

The [12] reviews RPL-based routing protocols, focusing on their challenges in scalability, mobility, and energy consumption. The authors propose future directions such as enhancing security and mobility support in IoT networks, highlighting the need for adaptable and energy-efficient solutions for large-scale IoT deployments.

The [13] provide a comprehensive survey of RPL and its enhancements for Low-Power and Lossy Networks (LLNs). They emphasize issues like loop detection, congestion, and energy efficiency and discuss proposed solutions to optimize RPL's performance in resource-constrained IoT environments.

[14] this paper surveys energy-efficient strategies in RPL for IoT applications, analyzing various techniques to improve network longevity. It highlights challenges in balancing energy consumption with network performance, suggesting optimization methods like adaptive duty cycling and energy-aware routing metrics.

The authors in [15] explore dynamic routing optimization in RPL for energy-constrained IoT networks, focusing on improving energy efficiency and reducing latency. They propose an energy-aware routing scheme that dynamically adjusts the routing path based on the residual energy of nodes.

The study of [16] investigates methods to enhance RPL for low-power and high-mobility IoT networks. The authors propose a mobility-aware mechanism that addresses the issue of frequent topology changes and packet losses, optimizing RPL's performance in dynamic IoT environments.

The authors in [17] propose a dynamic parent node selection mechanism in RPL to enhance network performance in IoT. This method improves the selection process by considering link reliability and node mobility, which helps in reducing packet loss and improving routing stability.

[18] This paper reviews adaptive RPL protocols designed for dynamic IoT environments, focusing on optimizing energy consumption, latency, and reliability. The authors provide a detailed comparison of adaptive strategies and suggest future research on cross-layer optimizations to further improve RPL's adaptability.

The [19] paper reviews optimization techniques for RPL, concentrating on energy consumption and network performance in IoT networks. The authors highlight advancements in load balancing and parent selection algorithms that contribute to more efficient energy use and improved overall network lifetime.

The authors in [20] present an enhanced RPL-based routing protocol aimed at improving network performance for low-power IoT devices. Their enhancements focus on link quality estimation and congestion control, leading to better packet delivery rates and reduced energy consumption.

The paper [21] investigates optimizations in RPL for high-density IoT networks, proposing an efficient routing mechanism that addresses scalability issues. The authors introduce techniques to minimize routing overhead and enhance data transmission reliability in congested networks.

Collectively, these studies underscore the ongoing efforts to refine RPL and dynamic routing strategies, addressing the evolving needs of IoT networks. They highlight the importance of adapting routing protocols to manage energy consumption, network topology changes, and overall performance, contributing to more resilient and efficient IoT systems.

Collectively, these studies underscore the ongoing efforts to refine RPL and dynamic routing strategies, addressing the evolving needs of IoT networks. They highlight the importance of adapting routing protocols to manage energy consumption, network topology changes, and overall performance, contributing to more resilient and efficient IoT systems.

3. Network Model

A. Background:

Let N represent the number of sensor nodes randomly deployed within a specified area. These nodes, characterized by their limited processing power and storage capacity, are tasked with sensing environmental data and relaying this information to a sink node for further analysis. Due to their constrained resources, these nodes are unable to transmit sensed data directly to the sink node in a peer-to-peer fashion. Instead, the data is passed through intermediate nodes until it reaches its destination. Given the need for real-time data transmission in contemporary applications[11], delays in data communication are unacceptable, necessitating the development of efficient routing algorithms that can function effectively on low-power embedded devices. A significant challenge arises from the fact that a substantial portion of the nodes' energy is expended on communication rather than computation. Consequently, there is ongoing research aimed at improving the efficiency of routing algorithms.

The Internet Engineering Task Force (IETF) has standardized the RPL (Routing Protocol for Low-Power and Lossy Networks) [13][18][20] protocol specifically for IoT applications, where reliable data delivery in real-time is critical. Unlike conventional routing protocols that determine the shortest path based on node locations or link distances, RPL employs a node's rank to establish the shortest path. RPL constructs a Directed Acyclic Graph (DAG) known as DODAG (Destination-Oriented DAG)[12][14][15][16] using various ICMP (Internet Control Message Protocol) messages, including DIO (DODAG Information Object), DAO (Destination Advertisement Object), DAO-ACK (DAO Acknowledgment), and DIS (DODAG Information Solicitation). The exchange of these control messages is regulated by a mechanism called the Trickle Timer, which controls the frequency of message dissemination. Additionally, the construction of the DODAG is guided by an Objective Function (OF)[17]. There are two primary Objective Functions used in RPL:

1. OF0: Constructs the DODAG based on hop count.

2. MRHOF (Minimum Rank with Hysteresis Objective Function): Builds the DODAG while considering hop count with added hysteresis to improve stability [19].

To enhance the performance of RPL in terms of network metrics such as Packet Delivery Ratio (PDR), Average Power Consumption, Network Lifetime, and End-to-End Delay, various research efforts have been undertaken. This study aims to further improve RPL's performance by integrating the concepts of dynamic sensor nodes and articulation nodes. The subsequent sections will detail the methodology for identifying articulation nodes within a network and the effects of dynamic node movement on these articulation points.

B. PROBLEM STATEMENT:

To elucidate the problem statement, consider a sub-graph of the provided graph illustrated in Figure 1, where the nodes are positioned as indicated. In this graph, node 3 functions as an articulation point, implying that its removal would partition the graph into two disconnected components. Given the 2D representation of the graph, identifying this critical node is straightforward. However, in real-world scenarios where nodes are deployed randomly, a systematic approach is required to identify such articulation points.

As previously mentioned, certain nodes in a network, known as articulation nodes, are critical to maintaining network connectivity. These nodes are analogous to cut-vertices in graph theory and serve as vital connection points between two connected components of a network. Consequently, all traffic traversing from one component to another must pass through these nodes, resulting in significant congestion. This node's role as a communication bridge requires it to process every incoming packet and forward it to the appropriate destination, leading to accelerated energy depletion. Consequently, the rapid reduction in available energy at the articulation node creates a gap in the network. This fragmentation results in multiple distinct components, ultimately leading to network disconnection, as illustrated in the figure.

Articulation nodes are typically undesirable in a network due to their potential to disrupt communication between various network devices. This is particularly critical in Internet of Things (IoT) applications that require real-time data delivery. Our primary focus is on preserving the integrity of the network's connectivity as a whole, treating it as a unified component even if other parts of the network encounter issues.

If an articulation node is present within the network, it must be addressed to prevent disconnection of communication links. The first step is to accurately identify the location of any articulation points within the network. Once identified, the second step involves instructing mobile nodes to move to these locations. This ensures that mobile nodes actively participate in maintaining communication by establishing links between the various connected components.

A potential concern may arise regarding the failure of even the mobile nodes within the network. This uncertainty can be mitigated by noting the extremely low probability of multiple nodes failing simultaneously in a given network.

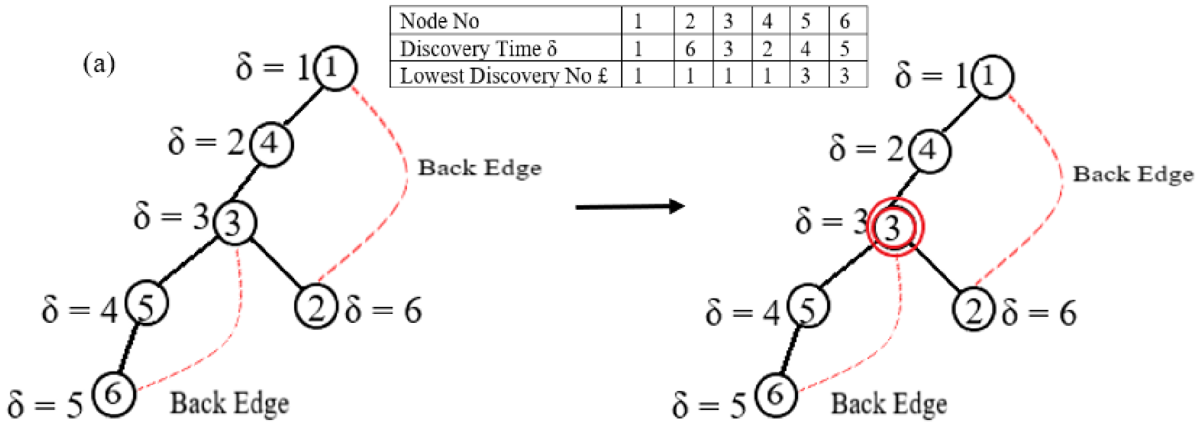
The proposed algorithm operates in two phases. In Phase I, the algorithm searches for the articulation node in the network. Once identified, the location of the articulation node is extracted from node information. In Phase II, the mobile node is directed to move towards the identified location and instructed to establish a connection by selecting an appropriate parent node, thereby distributing the load of the articulation node.

4. Mathematical Model

A. Procedure 1: Identifying the Articulation Node in a Given Network

RPL creates a Destination-Oriented Directed Acyclic Graph (DODAG) for a specific network of randomly placed nodes to facilitate communication between nodes and a sink. Let G=(V,E) represent the set of vertices and edges of the given DODAG created by RPL. To locate the articulation node in the given graph G, first, construct the Depth-First Search (DFS) tree of the graph based on the DFS algorithm. The algorithm begins at the root node and traverses the tree to represent the DFS concept. It starts by marking the root node as visited and progresses downwards to determine the Discovery time δ for each node in the network, i. e .how long it took for each node in a given network to be discovered The sequence in which nodes are visited in a given graph is known as Discovery time δ.

Following the calculation of each node's discovery time, the next step is to determine each node's lowest discovery time, denoted by £. The lowest discovery time represents the depth of a node relative to the starting node, considering the back edges added during the construction of the DFS tree. Specifically, if a graph contains two nodes u and v, an edge (u,v) is considered a back edge if it is part of the original graph but not part of the DFS tree, and node v is a descendant of node u.



Node No	1	2	3	4	5	6
Discovery Time δ	1	6	3	2	4	5
Lowest Discovery No £	1	1	1	1	3	3

Fig 2(a): DFS tree of a given graph with δ and £ Fig 2(b): Articulation Node of a given graph

After calculating the Discovery time δ and the lowest discovery time £ for the initial node, the algorithm examines its neighboring nodes. It determines whether the adjacent vertices have been visited and acts accordingly. If an adjacent vertex has not been visited, the algorithm labels it as the current vertex and determines its Discovery time δ as well as its lowest discovery time £.

Once the Discovery time δ and lowest discovery time £ for all nodes have been determined, the algorithm proceeds to identify articulation nodes. To achieve this, the algorithm arbitrarily considers any two nodes, x and y, where x is the parent node and y is its child node. Node x is considered an articulation node if the lowest discovery time of node y is greater than or equal to the Discovery time of node x.

$£ (y) \geq \delta (x)$ for (x,y Ɛ DFS tree & $x \leftarrow y$) (1)

For example Fig 2(a) represents the DFS tree of a graph which is depicted in firgure 1(a). Algorithm has calculated the node Discovery time δ and Lowest Discovery Time £ of each node and tabulated in Table I. Next in order to find the articulation node, consider any two node (5,6) where node 5 is parent of node 6 as per DFS tree. By applying euation (1) lets check whether node 5 is an articulation node or not. That is

$£ (6) \geq \delta (5) \Rightarrow 3 \geq 4 \Rightarrow$ False (2)

As the comparision resulted in false node 5 is not an articulation node. Next consider other pair of nodes for example (3, 5) and apply equation (1)

$£ (5) \geq \delta (3) \Rightarrow 3 \geq 3 \Rightarrow$ True (3)

Above comparion results in true and hence it can be concluded that there exist a articulation node in the given network and node 5 acts as a articulation node which is depicted in fig 2(b).

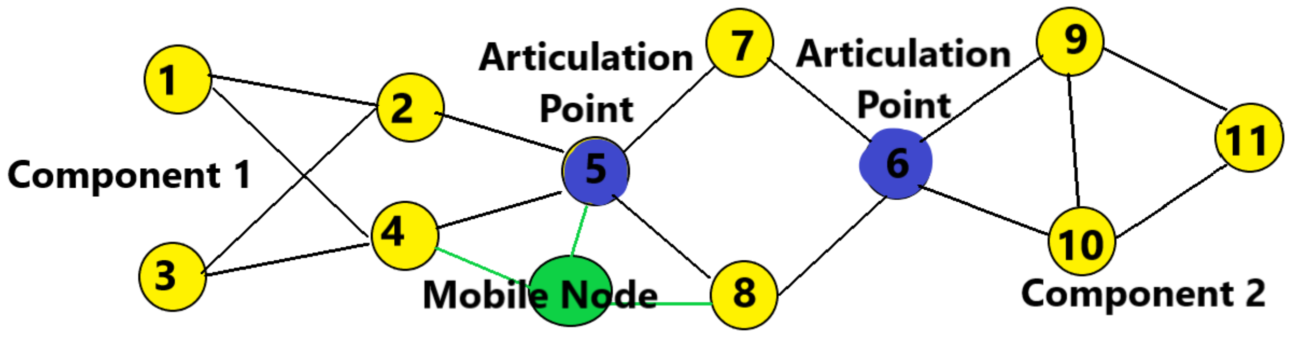


Fig 3: Mobile Node establishing an alternate connection and sharing the load of Articulation Node

Table I: Discovery time δ and Lowest Discovery Time £

Node No	1	2	3	4	5	6	7	8	9	10	11
Discovery Time (δ)	1	2	3	4	5	6	7	8	9	10	11
Lowest Discovery No (£)	1	2	1	3	4	4	4	7	7	7	7
Articulation Point	Yes	No	Yes	No	No	No	No	No	No	No	No

B. *Procedure 2: Relocating Mobile Nodes to the Articulation Node and Constructing a New DAG*

Once the articulation node is identified its location is extracted from node's info and dynamic node is instructed to move towards that location as depicted in Fig 3. Mobile node moves to the specified location in a negligibly short period of time that is determined by

$\Delta t = \eta^{ET} - \eta^{ST}$ (4)

where η^{ET} and η^{ST} represent the end time and start time of the movement, respectively.

The Euclidian Distance equation provides the distance travelled by the mobile node as

$$\eta_{dist} = \sqrt{(x_a - x_m)^2 + (y_a - y_m)^2} \quad (5)$$

The Potential Field Algorithm using Vector-Based Approaches is utilised to instruct the mobile node to move towards that quadrant where articulation node is present. The force f is required to move mobile node from one point to another. Two types of forces influence mobile sensor nodes: repulsive forces from obstacles f(rep) and attracting forces from pure coverage zones, f(attr) as per equation (6). The forces between any two nodes are obtained by attractive and repulsive patterns as shown in fig 4 and the force generated between given two nodes η and α is given by:

$$f\,(\eta.\alpha) = f(\mathrm{attr}_{(\eta.\alpha)}) + f(\mathrm{rep}_{(\eta.\alpha)}) \quad (6)$$

Where,

$$f(\mathrm{attr}_{(\eta.\alpha)}) = \left(\frac{-\lambda_{attr}}{d(\eta,\alpha)^2}\right)\left(\frac{loc(\eta)-loc(\alpha)}{d(\eta,\alpha)}\right) \quad (7)$$

$$f(\mathrm{rep}_{(\eta.\alpha)}) = \begin{cases} \left(\frac{-\lambda_{rep}}{d(\eta,\alpha)^2}\right)\left(\frac{loc(\eta)-\,loc(\alpha)}{d(\eta,\alpha)}\right), & for\ critical\ connection \\ 0, & Otherwise \end{cases} \quad (8)$$

where, loc(η) and loc(α) indicates the position of mobile and articulation nodes in a given network, d(η,α) is the distance between η and α, λ_{rep} and λ_{rep} are the constraints of force. This approach ensures the effective relocation of the mobile node to the articulation node's vicinity, enabling the establishment of new communication links and the construction of an updated Directed Acyclic Graph (DAG) to enhance network connectivity and resilience.

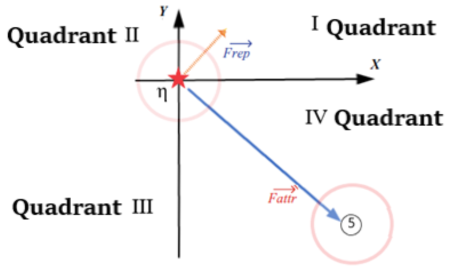


Fig 4: Virtual Attractions and Repulsions

Depending on the coordinates of the articulation nodes, the angle of movement for the mobile node is determined, finalizing the quadrant of interest, such as the IV quadrant in our example. By applying the appropriate force, the node moves towards the destination in a negligible amount of time. Upon reaching the destination, topology inconsistency arises as the mobile node transitions from one parent node to another. This transition phase is referred to as "handoff."

Given that the RPL routing protocol is inherently adaptive, a new Directed Acyclic Graph (DAG) is constructed to incorporate the mobile node. Once the mobile node receives Destination Information Object (DIO) messages from neighboring nodes, as illustrated in Figure 12, the modified algorithm selects the optimal parent from the available parent set based on the new Objective Function (OF). The proposed OF is calculated by evaluating the efficiency of each node, which is assessed using routing metrics such as Expected Transmission Count (ETX), Packet Delivery Ratio (PDR), Residual Energy, Latency, and Throughput.

C. Routing Metrics:

During the node movement, network topology may experience inconsistencies due to the switching of parent nodes, a process known as "handoff." To address this, the AM-RPL protocol dynamically adapts by building a new Directed Acyclic Graph (DAG) that includes the newly positioned mobile node. The optimal parent is selected based on the updated metrics, which are crucial for maintaining efficient routing.

The performance metrics used for selecting the optimal parent node are derived from the following calculations:

- **Expected Transmission Count (ETX):** ETX measures the number of transmissions, including retransmissions, required to successfully deliver a packet from a source node to a destination node. It is calculated by:

$$ETX = \frac{1}{Ps * Pr} \tag{9}$$

where Ps is the probability of successful packet transmission, and Pr is the probability of receiving an acknowledgment for the transmitted packet.

- **Packet Delivery Ratio (PDR):** PDR is the ratio of successfully delivered packets to the total packets sent to the destination node, expressed as a percentage:

$$PDR = \frac{T_{rec}}{T_{sent}} * 100 \tag{10}$$

where T_{rec} is the number of packets received, and T_{sent} is the total number of packets sent.

- **Residual Energy (ϑres(i)):** Residual energy is the remaining energy of the node, calculated as:

$$\vartheta_{res}(i) = \vartheta_{init}(i) - \vartheta_{cons}(i) \tag{11}$$

where $\vartheta_{init}(i)$ is the initial energy of the node and $\vartheta_{cons}(i)$ is the energy consumed by the node.

- **Energy Consumed ($\vartheta_{cons}(i)$):**

Energy depleted $\vartheta_{cons}(i)$ is measured by amount of energy a node consumes while it is sensing, processing, communicating and transmitting. $\vartheta_{cons}(i)$ is given by

$$\vartheta_{cons}(i) = \frac{(\lambda t * \zeta t + \lambda r * \zeta r + \lambda cpu * \zeta cpu + \lambda cpu\_idle * \zeta cpu\_idle)}{£} * V \tag{12}$$

where λ_t, λ_r, λ_{cpu}, λ_{cpu_idle} indicates Transmit Time, Listen Time, CPU time, and LPM time respectively. The microcontroller's active-mode current requirements are denoted by ζ_t = 19.5mA, ζ_r = 21.5mA, ζ_{cpu} = 1.8mA, ζ_{cpu_idle} = 0.0545mA. Voltage V = 3V and £ is a constant typically has a value of 32768 ticks per second.

- **Latency:** Latency is the time it takes for a packet to reach its destination after being sent from the source:

$$\text{Latency } (l) = P_{arr} - P_{sent} \tag{13}$$

where P_{arr} is the arrival time, and P_{sent} is the send time of the packet.

- **Throughput:** Throughput is the ratio of the total data transmitted to the simulation time:

$$Throughput = \frac{T_{tot}}{\tau} \tag{14}$$

Where T_{tot} is the total data traffic and τ is the simulation time.

- **Efficiency (ξ(i)):** The efficiency of a node i is calculated by:

$$\xi(i) = \frac{\vartheta avg\ (i)}{\chi} \tag{15}$$

where, ϑ_{avg} is average energy of a node which is given by

$$\vartheta_{avg}(i) = \frac{\vartheta_{res}(i)}{ETX(i) * \vartheta_{bit}} \tag{16}$$

Here $\vartheta_{bit} \rightarrow$ Residual energy of node i required to transmit each bit. And ETX(i) is expected transmission count of link from node i to its parent.

- **Current Data Traffic Flow (χ):** The current data traffic flowing through the link is given by:

$$\chi = \frac{T_{tot}(i)}{DTR} \tag{17}$$

Where DTR is the rate at which data transferred and is measured in bits per second and $T_{tot}(i)$ is the total data traffic sent from node i:

$$T_{tot}(i) = T_{tx}(i) + \sum_{j\, \varepsilon\, child\ array} T_{rx}(j) \tag{18}$$

$T_{tx}(i)$ represents the amount of data generated and transmitted by node i itself and $T_{rx}(j)$ is the amount of data traffic received by set of connected child node of node i.

- **Final Step: DAG Construction**

After evaluating these performance metrics, the new DAG is constructed by selecting the node with the highest efficiency as the parent as in fig 5. This approach ensures optimal load balancing and enhances the overall performance of the network. By integrating the newly positioned mobile node into the network, the algorithm establishes communication links between previously disconnected components, thereby distributing the load and preserving network connectivity.

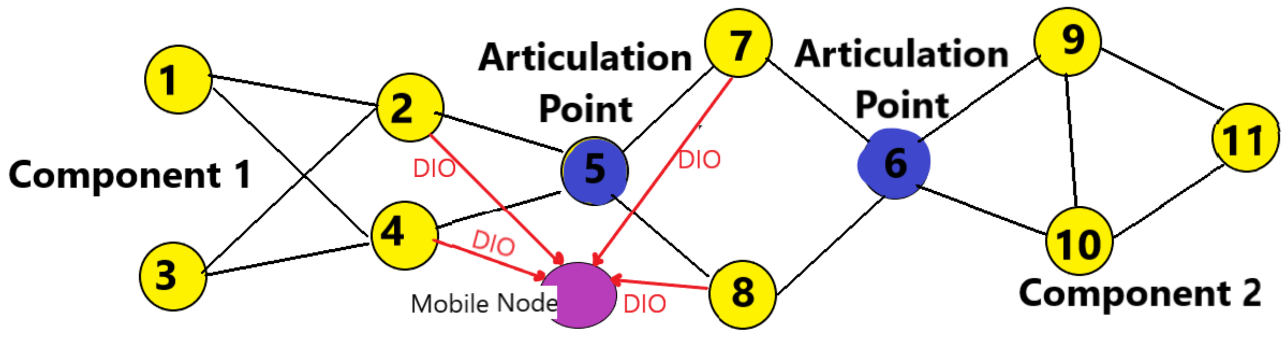


Fig 5: Mobile Node Selecting Most Effective Parent

Table II : Used Notation Summary

ETX	Expected Transmission Count	£	Node's Lowest Discovery Time
Δt	Mobile Movement Time	δ	Discovery Time
η^{ET}	Mobile Node Movement End Time	υ	Velocity of the Mobile Node
η^{ST}	Mobile Node Movement Start Time.	Δt	Time Required for the Mobile Node to Move to the Articulation Node.
η_{dist}	Distance Between the Mobile Node and the Articulation Node.	P_{arr}	Packet Arrival Time.
x_a,y_a	Coordinates of the Articulation Node.	P_{sent}	Packet Sent Time.
x_m,y_m	Coordinates of Mobile Node.	τ	Total simulation time.
$T_{tot}(i)$	Total Packets Sent in the Network	$T_{tot}(i)$	Total packets sent by node i.
f	Potential Field Force.	f_{attr}	Attractive Force.
f_{rep}	Repulsive Force.	(x'_m, y'_m)	Update Position of the Mobile Node.
Δx and Δy	The Movement Components Derived from The Force f.	P_s	Probability of Successful Packet Transmission
P_r	Probability Of Receiving an Acknowledgment	PDR	Packet Delivery Ratio
T_{rec}	The Number of Packets Received	T_{sent}	The Number of Packets Sent
ϑ_{init}	The Initial Energy	ϑ_{res}	Residual Energy
ϑ_{cons}	Consumed Energy	λ_r	Listen Time
λ_t	Transmit Time	λ_{cpu}	CPU Time
l	Latency	λ_{cpu_idle}	LPM Time
T_{tot}	Total Data Traffic	τ	Simulation Time
ϑ_{avg}	Average Energy	$\xi(i)$	Efficiency of a Node i
ϑ_{bit}	Residual Energy of Node i Required to Transmit Each Bit	χ	Current Data Traffic Flow
$T_{tx}(i)$	Amount of Data Generated	$T_{rx}(j)$	Amount of Data Traffic Received

5. Algorithm

A. Algorithm1: Relocate Mobile Nodes and Update DAG

Input:

Coordinates of the articulation node (x_a,y_a)

Coordinates of the mobile node (x_m,y_m)

Routing metrics: ETX, PDR, Residual Energy, Latency, Throughput, and Efficiency

Output:

Updated Directed Acyclic Graph (DAG)

Steps:

1. **Identify Articulation Node:**

Extract coordinates (x_a,y_a) of the articulation node from node information.

2. **Calculate Movement Parameters:**

- Compute the Euclidean distance η_{dist} between the mobile node and the articulation node:

$$\eta\text{dist} = \sqrt{(x_a - x_m)^2 + (y_a - y_m)^2}$$

- Determine the angle θ of movement towards the articulation node

$$\theta = \ arctan\left(\frac{y_a - y_m}{x_a - x_m}\right)$$

- Calculate the time Δt required for the mobile node to move to the articulation node

$$\Delta t = \frac{\eta_{\text{dist}}}{v}$$

where v is the velocity of the mobile node.

3. **Apply Potential Field Algorithm:**

- Compute the attractive force f_{attr} towards the articulation node and the repulsive force f_{rep} from obstacles.
- The resultant force f is given by:

$$f = f_{attr} - f_{rep}$$

- Update the position $(x'_m, y'_m)$of the mobile node using the force:

$$(x'_m, y'_m) = \ (x_m + \ \Delta x\,,\ y_m + \ \Delta y)$$

- where Δx and Δy are the movement components derived from the force f.

4. **Perform Handoff:**

- Address topology inconsistency due to the node's transition between parent nodes. This phase is termed "handoff."

5. **Update Network Topology:**

- Utilize the RPL routing protocol to build a new DAG incorporating the relocated mobile node.
- Receive Destination Information Object (DIO) messages from neighboring nodes.

B. *Algorithm 2: Evaluate Routing Metrics for New Parent Node:*

Input:

- Coordinates of the articulation node
- Parameters for calculating routing metrics (ETX, PDR, Residual Energy, Latency, Throughput)
- Metrics for mobile node efficiency
- Neighboring nodes' information: Discovery time, Lowest discovery time, and connectivity details

Output:

- Optimal parent node for new DAG construction

Steps:

1: Calculate ETX:

- Measure the probability of successful packet transmission P_s and acknowledgment reception P_r.
- Compute ETX:

$$ETX = \frac{1}{P_s \, . P_r}$$

2. **Calculate PDR:**

- Measure the number of successfully received packets T_{rec} and total packets sent T_{sent}.
- Compute PDR:

$$PDR = \left(\frac{T_{rec}}{T_{sent}}\right) X \; 100$$

3. **Calculate Residual Energy ($\vartheta_{res}(i)$):**

- Calculate energy consumption $\vartheta_{cons}(i)$ using:

$$\vartheta_{cons}(i) = \frac{(\lambda_t * \zeta_t + \lambda_r * \zeta r + \lambda cpu * \zeta cpu + \lambda cpu\_idle * \zeta cpu\_idle)}{£} * V$$

Where λ_t, λ_r, λ_{cpu}, λ_{cpu_idle} are the times spent in transmit, listen, CPU, and idle modes respectively, and $\zeta t, \zeta r, \zeta cpu, \zeta cpu\_idle$ are the corresponding current requirements.

- Compute residual energy:

$$\vartheta_{res}(i) = \vartheta_{init}(i) - \vartheta_{cons}(i)$$

4. Calculate Latency (l):

- Measure the packet arrival time P_{arr} and packet sent time P_{sent}.
- Compute latency

$$l = P_{arr} - P_{sent}$$

5. **Calculate Throughput:**

- Measure the total transmitted data T_{tot} and simulation time τ.
- Compute throughput:

$$\text{Throughput}= \frac{T_{tot}}{\tau}$$

6. **Calculate Efficiency (*ξ(i)*):**

- Compute average energy ϑavg(i):

$$\vartheta_{avg}(i) = \frac{\vartheta_{res}(i)}{ETX(i) * \vartheta_{bit}} \quad ,$$

where ϑ_{bit} is the energy per bit.

7. **Compute current data traffic χ:**

$$\chi = \frac{T_{tot}(i)}{DTR},$$

where DTR is the data transfer rate.

8. **Compute efficiency:** $\xi(i) = \frac{\vartheta avg\ (i)}{\chi}$

9. **Select Optimal Parent Node:**

- For each candidate parent node p_i, compute the Objective Function (OF):

$$OF_i = w_1.ETX_i + w_2.(1 - PDR_i) + w_3.(1 - \vartheta_{res}(i)) + w_4.(l_i) + w_5.(1 - Throughput_i)$$

where w_1, w_2, w_3, w_4, w_5 are the weights assigned to each metric.

- Select the parent node p^* that minimizes the OF:

$$p^* = {}^{argmin}{}_{pi}(OF_i)$$

10. Update DAG with New Parent:

- Modify the DAG to reflect the new parent nodep^* for the mobile node.
- Ensure the updated DAG maintains optimal network connectivity and performance.

End of Algorithm.

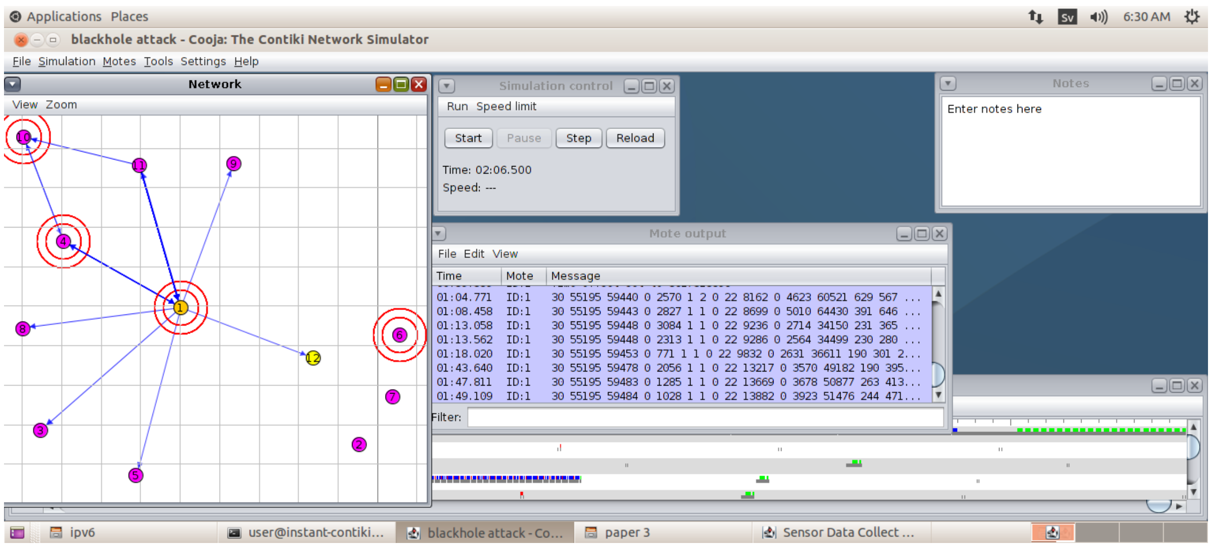
6. Simulation Setup

The experimental evaluations were conducted using the Cooja simulator, a prominent tool for modeling and analyzing IoT networks, particularly those leveraging the Contiki operating system. This study focused on testing the Articulation Node Based Mobile Node Routing Protocol (AM-RPL) across different network sizes to gauge its performance and efficiency.

A. Network Configuration

The simulation setup involved three distinct configurations based on the number of nodes to examine the protocol's scalability and robustness:

1. **30 Nodes:** Nodes were distributed randomly over a 100x100 meter area. This configuration was used to analyze the protocol's behavior in a relatively sparse network, providing insights into its performance under low node density conditions.
2. **60 Nodes:** The network was expanded to a 150x150 meter area with 60 nodes. This intermediate density setup was intended to evaluate how the protocol handles increased traffic and potential network congestion.
3. **100 Nodes:** For the most densely populated scenario, nodes were arranged within a 200x200 meter area. This configuration aimed to test the protocol’s efficiency and scalability in a high-density network environment



```
Time: 02:06.500
Speed: ---

Time       Mote   Message
01:04.771  ID:1   30 55195 59440 0 2570 1 2 0 22 8162 0 4623 60521 629 567 ...
01:08.458  ID:1   30 55195 59443 0 2827 1 1 0 22 8699 0 5010 64430 391 646 ...
01:13.058  ID:1   30 55195 59448 0 3084 1 1 0 22 9236 0 2714 34150 231 365 ...
01:13.562  ID:1   30 55195 59448 0 2313 1 1 0 22 9286 0 2564 34499 230 280 ...
01:18.020  ID:1   30 55195 59453 0 771 1 1 0 22 9832 0 2631 36611 190 301 2...
01:43.640  ID:1   30 55195 59478 0 2056 1 1 0 22 13217 0 3570 49182 190 395...
01:47.811  ID:1   30 55195 59483 0 1285 1 1 0 22 13669 0 3678 50877 263 413...
01:49.109  ID:1   30 55195 59484 0 1028 1 1 0 22 13882 0 3923 51476 244 471...
```

Fig 6: Simulation Environment

B. *Simulation Parameters*

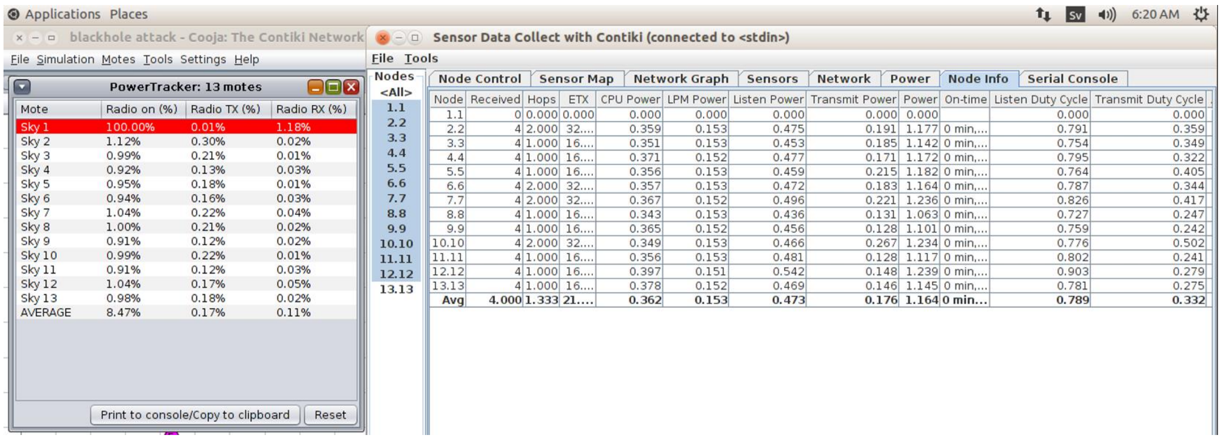
- **Duration and Intervals:** Each network configuration was simulated for a duration of 3000 seconds as shown in Fig 6. This time frame was selected to ensure comprehensive data collection and to observe the protocol's long-term performance.
- **Routing Protocol and Configurations:**
  - The AM-RPL protocol was deployed with configurations to detect articulation nodes and adjust routing dynamically. The protocol was designed to redistribute the load by guiding mobile nodes towards identified articulation nodes.
  - **Articulation Node Detection:** Nodes were programmed to identify articulation points within the network, which are crucial for maintaining connectivity and preventing network partitions.
  - **Mobile Node Movement:** Mobile nodes were controlled to move towards these articulation nodes based on predefined trajectories and forces, using the Potential Field Algorithm to optimize their paths.

C. *Performance Metrics*

- **Packet Delivery Ratio (PDR):** This metric was used to measure the ratio of successfully delivered packets to the total number of packets transmitted, reflecting the reliability of the network.
- **End-to-End Delay:** The time taken for packets to travel from the source node to the destination node was measured, providing insights into the latency introduced by the network.
- **Energy Consumption:** The energy usage of the nodes was monitored to assess how efficiently the protocol manages energy resources and affects node battery life.

D. *Tools and Analysis*

- **Simulation Environment:** Cooja version 3.x, integrated with Contiki OS, was used for running the simulations. This setup allowed for realistic modeling of network conditions and node interactions.



Mote	Radio on (%)	Radio TX (%)	Radio RX (%)
Sky 1	100.00%	0.01%	1.18%
Sky 2	1.12%	0.30%	0.02%
Sky 3	0.99%	0.21%	0.01%
Sky 4	0.92%	0.13%	0.03%
Sky 5	0.95%	0.18%	0.01%
Sky 6	0.94%	0.16%	0.03%
Sky 7	1.04%	0.22%	0.04%
Sky 8	1.00%	0.21%	0.02%
Sky 9	0.91%	0.12%	0.02%
Sky 10	0.99%	0.22%	0.01%
Sky 11	0.91%	0.12%	0.03%
Sky 12	1.04%	0.17%	0.05%
Sky 13	0.98%	0.18%	0.02%
AVERAGE	8.47%	0.17%	0.11%

Node	Received	Hops	ETX	CPU Power	LPM Power	Listen Power	Transmit Power	Power	On-time	Listen Duty Cycle	Transmit Duty Cycle
1.1	0	0.000	0.000	0.000	0.000	0.000	0.000	0.000		0.000	0.000
2.2	4	2.000	32....	0.359	0.153	0.475	0.191	1.177	0 min,...	0.791	0.359
3.3	4	1.000	16....	0.351	0.153	0.453	0.185	1.142	0 min,...	0.754	0.349
4.4	4	1.000	16....	0.371	0.152	0.477	0.171	1.172	0 min,...	0.795	0.322
5.5	4	1.000	16....	0.356	0.153	0.459	0.215	1.182	0 min,...	0.764	0.405
6.6	4	2.000	32....	0.357	0.153	0.472	0.183	1.164	0 min,...	0.787	0.344
7.7	4	2.000	32....	0.367	0.152	0.496	0.221	1.236	0 min,...	0.826	0.417
8.8	4	1.000	16....	0.343	0.153	0.436	0.131	1.063	0 min,...	0.727	0.247
9.9	4	1.000	16....	0.365	0.152	0.456	0.128	1.101	0 min,...	0.759	0.242
10.10	4	2.000	32....	0.349	0.153	0.466	0.267	1.234	0 min,...	0.776	0.502
11.11	4	1.000	16....	0.356	0.153	0.481	0.128	1.117	0 min,...	0.802	0.241
12.12	4	1.000	16....	0.397	0.151	0.542	0.148	1.239	0 min,...	0.903	0.279
13.13	4	1.000	16....	0.378	0.152	0.469	0.146	1.145	0 min,...	0.781	0.275
Avg	**4.000**	**1.333**	**21....**	**0.362**	**0.153**	**0.473**	**0.176**	**1.164**	**0 min...**	**0.789**	**0.332**

Fig 7: Sensor Data Collection with Contiki.

- **Data Collection and Visualization:** Network performance and metrics were analyzed using Cooja's visualization tools and external data analysis software, enabling detailed examination and comparison across different node configurations as shown in fig 7.

These simulations were designed to thoroughly evaluate the AM-RPL protocol's performance, focusing on its capability to maintain network connectivity, optimize routing, and manage node energy consumption across various network scales.

7. Performance Evaluation

The performance of the Articulation Node Based Mobile Node Routing protocol (AM-RPL) was rigorously evaluated using the Cooja simulator, which is renowned for its ability to model large-scale IoT networks accurately. The evaluation metrics included Packet Delivery Ratio (PDR), Radio Duty Cycle, Charge Consumption, and Parent Switches, which are critical indicators of network performance and efficiency.

In the conducted experiments, the AM-RPL algorithm demonstrated a significant enhancement in the Packet Delivery Ratio (PDR) when compared to traditional static IoT network configurations as shown in fig 8. The dynamic redistribution of load among mobile nodes effectively mitigated the bottleneck effect typically observed at central hub nodes. This redistribution ensured a more consistent and reliable packet delivery across the network, thereby reducing the likelihood of packet loss and improving overall data integrity.

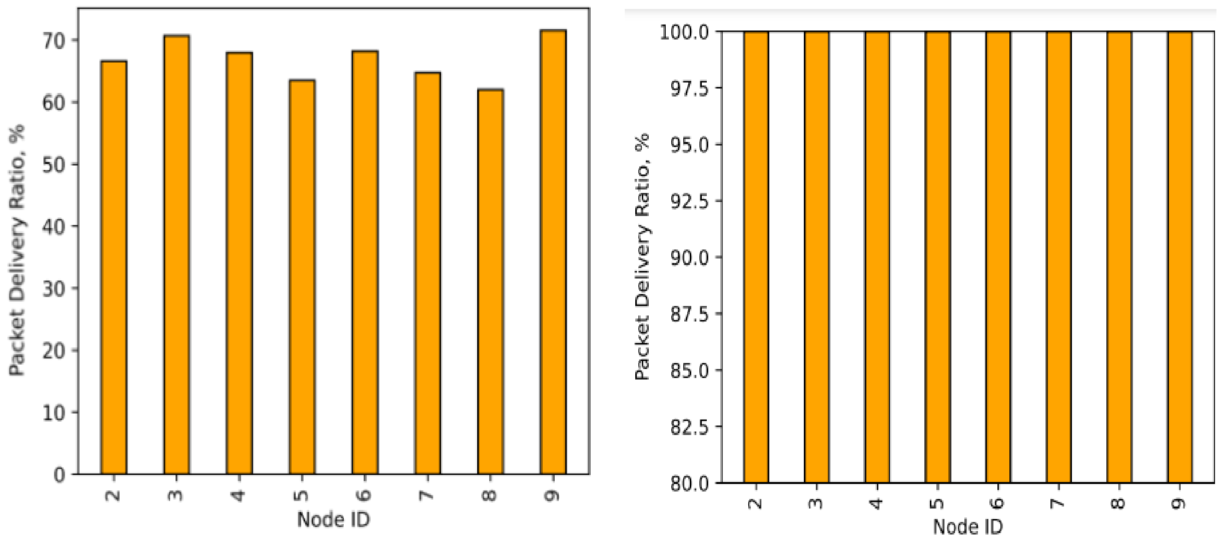


Fig 8: Packet Delivery Ration in case of RPL vs AM-RPL

The Radio Duty Cycle, which measures the proportion of time the radio is active, showed considerable improvement under the AM-RPL protocol as in fig 9. By optimizing the use of the radio resources through efficient routing and load distribution, AM-RPL significantly reduced the duty cycle. This reduction is crucial for extending the battery life of IoT devices, as lower duty cycles lead to less energy consumption.

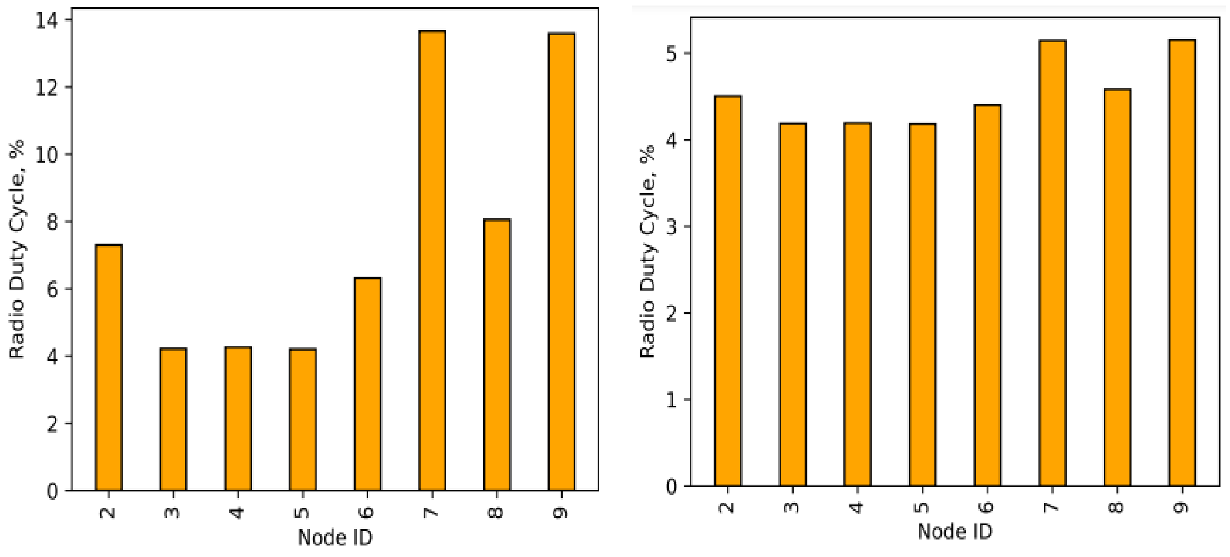


Fig 9: Comparison of Radio Duty Cycle in case of RPL vs AM-RPL

Charge Consumption, another pivotal metric, was markedly reduced with the AM-RPL protocol as observed in fig 10. The efficient load balancing prevented the rapid depletion of energy resources in any single node, thus distributing the energy usage more evenly across the network. This balanced consumption is essential for prolonging the operational lifespan of the network and enhancing its sustainability.

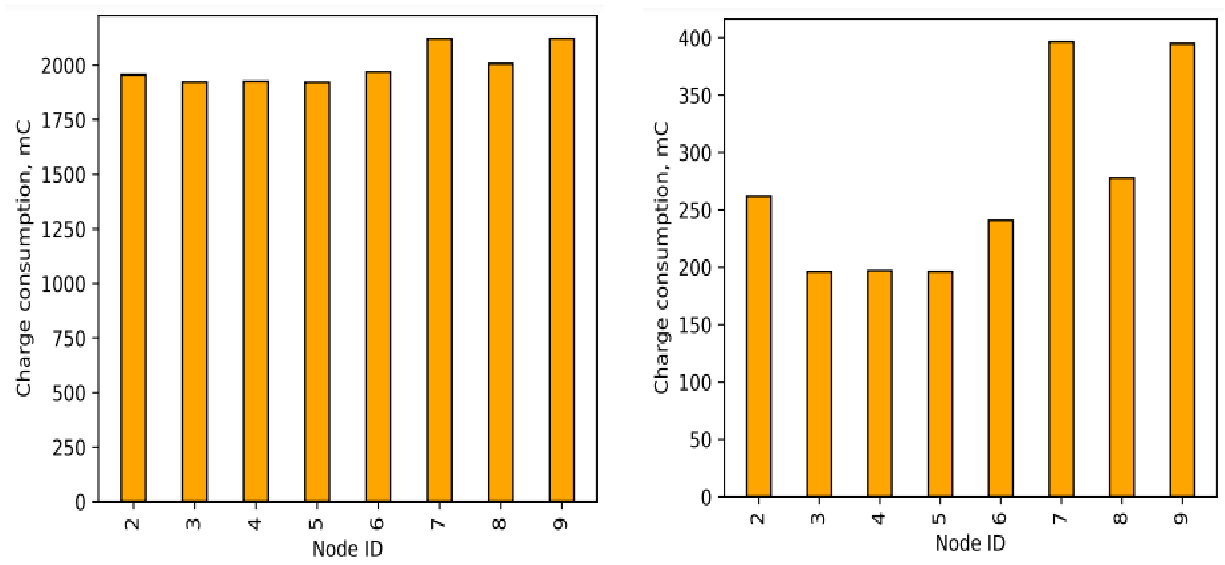


Fig 10: Charge Consumption in case of RPL vs AM-RPL

Lastly, Parent Switches, which indicate the stability of routing paths by counting the number of times a node changes its parent node, were minimized in the AM-RPL protocol as in fig 11. Fewer parent switches suggest a more stable and reliable network, as frequent changes can lead to increased latency and packet loss. By maintaining stable parent-child relationships, AM-RPL ensured smoother data transmission and better overall network performance.

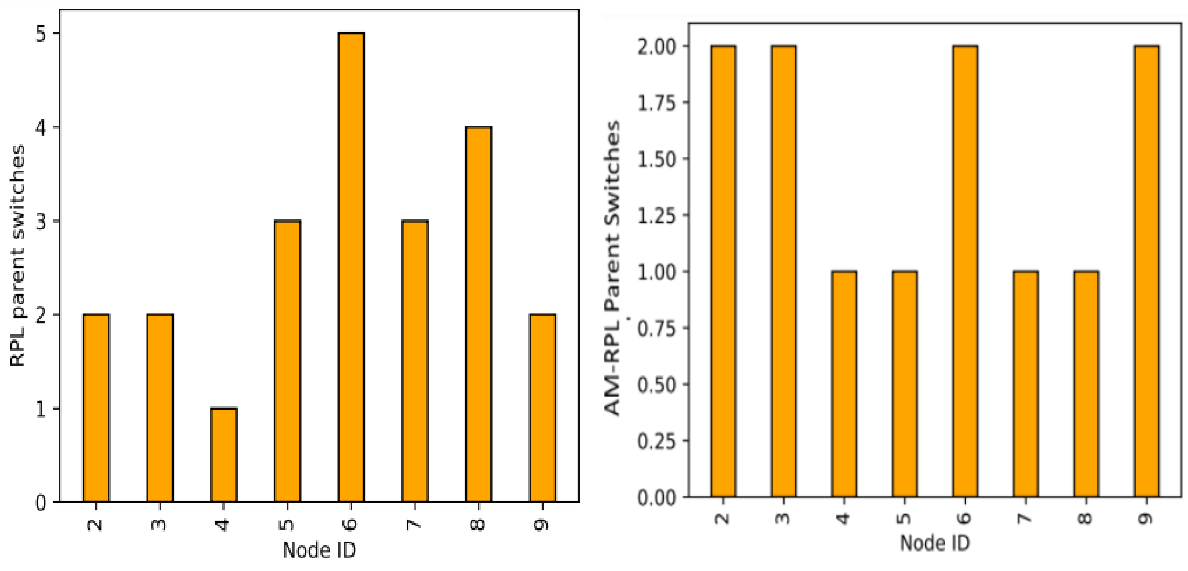


Fig 11: Parent Switches in case of RPL vs AM-RPL

The experiment's findings conclusively indicate that the AM-RPL algorithm outperforms typical static network scenarios in terms of PDR, radio duty cycle, and energy consumption. The strategic movement and load balancing of dynamic nodes fostered a more resilient and efficient network architecture. These improvements underscore the potential of AM-RPL to revolutionize IoT network management, offering a scalable solution to the challenges posed by traditional static configurations. Future research will focus on further optimization and testing of the protocol in diverse and larger network environments to fully exploit its capabilities and applications.

It is important to note that we are not judging performance improvement in terms of percentage in this paper, as different readings were observed when implemented for the same network and the same scenario. However, each time, AM-RPL consistently promised improved results.

8. Conclusion

The deployment of the Articulation Node Based Mobile Node Routing protocol (AM-RPL) adeptly addresses the paramount challenge of network failure precipitated by congestion and energy depletion at pivotal central nodes in static IoT networks. By dynamically identifying articulation nodes and strategically directing mobile nodes to these loci, AM-RPL proficiently redistributes the load, thereby fortifying network resilience. The simulation outcomes within the Cooja environment reveal marked enhancements in Packet Delivery Ratio (PDR), Radio Duty Cycle, Charge Consumption, and Parent Switches. These results substantiate the effectiveness of the AM-RPL algorithm in preserving network stability and optimizing performance by mitigating congestion-induced failures.

Prospective endeavors may encompass the validation of the protocol across varied IoT network topologies and the pursuit of further refinements to augment its scalability and robustness. While we did not measure performance improvement in percentage terms due to varying readings under the same network conditions, AM-RPL reliably showed enhancements in reliability, energy efficiency, and network stability. This consistent improvement underscores the protocol's potential for optimizing large-scale IoT networks.

Refrences

[1] Leelavathi R, Vidya A, “Articulation Node Based Mobile Node Routing Protocol in a Hybrid IoT Network”, International Conference on Recent Advances in Information Technology for Sustainable Development, India, 2024.

[2] O. Gaddour, A. Koubäa, R. Rangarajan, O. Cheikhrouhou, E. Tovar and M. Abid, "Co-RPL: RPL routing for mobile low power wireless sensor networks using Corona mechanism," Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems (SIES 2014), Pisa, Italy, 2014, pp. 200-209, doi: 10.1109/SIES.2014.6871205.

[3] Shah, Z.; Levula, A.; Khurshid, K.; Ahmed, J.; Ullah, I.; Singh, S. Routing Protocols for Mobile Internet of Things (IoT): A Survey on Challenges and Solutions. Electronics 2021, 10, 2320. https://doi.org/ 10.3390/electronics10192320.

[4] Gavrilovska, Liljana & Nikoletseas, Sotiris. (2011). Mobility Aspects in WSN. 10.1007/978-1-84996-510-1_6.

[5] Ghosh, A., Sharma, V., & Singh, R. (2020). A survey on RPL (Routing Protocol for Low Power and Lossy Networks). IEEE Access, 8, 170041-170061. https://ieeexplore.ieee.org/document/9097804

[6] Prasad, R., Yadav, A., & Kumar, S. (2021). Dynamic RPL-based routing protocol for IoT networks. Sensors, 21(2), 379. https://www.mdpi.com/1424-8220/21/2/379

[7] Patel, S., Vora, J., & Shah, H. (2021). Recent enhancements to RPL: Path selection and energy-efficient strategies. IEEE Internet of Things Journal, 8(4), 2345-2355. https://ieeexplore.ieee.org/document/9409138

[8] Zhang, J., Wu, L., & Li, X. (2022). Dynamic parent selection strategy in RPL for IoT networks. Computer Networks, 203, 107654. https://www.sciencedirect.com/science/article/pii/S157087052200080X

[9] Chen, H., Zhang, W., & Liu, Y. (2023). Adaptive routing strategies in RPL: Optimizing energy, latency, and throughput. Ad Hoc Networks, 132, 102688. https://www.sciencedirect.com/science/article/pii/S1389128623000571

[10] Liu, Z., Wang, X., & Zhang, Y. (2023). Enhancing RPL protocol for dynamic IoT networks. IEEE Transactions on Network and Service Management, 20(2), 212-226. https://ieeexplore.ieee.org/document/10161082

[11] Patel, M., Patel, P., & Patel, S. (2022). Optimizing RPL for low-power and high-mobility applications. Journal of Communications and Networks, 24(1), 43-55. https://www.hindawi.com/journals/jcnc/2022/1859476/

[12] Alsaade, F., & Al-Rubaye, S. (2020). A Review of RPL-Based Routing Protocols for IoT Networks: Challenges and Future Directions. IEEE Access, 8, 159458-159475. https://ieeexplore.ieee.org/document/9149524

[13] Bera, S., Misra, S., & Pati, P. (2021). A Comprehensive Survey on RPL and Its Enhancements for Low-Power and Lossy Networks. Computer Networks, 192, 108087. https://www.sciencedirect.com/science/article/pii/S1389128621000889

[14] Cheng, Y., & Chen, X. (2021). Energy-Efficient RPL for IoT Applications: A Survey and Challenges. IEEE Internet of Things Journal, 8(9), 7237-7253. https://ieeexplore.ieee.org/document/9349580

[15] Fang, Y., Zhang, S., & Wang, Y. (2022). Dynamic Routing Optimization in RPL for Energy-Constrained IoT Networks. IEEE Transactions on Wireless Communications, 21(3), 1865-1877. https://ieeexplore.ieee.org/document/9702169

[16] Gomez, J., & Garcia, F. (2022). Improving RPL for Low-Power and High-Mobility IoT Networks: Techniques and Applications. IEEE Transactions on Network and Service Management, 19(2), 1453-1465. https://ieeexplore.ieee.org/document/9724709

[17] Kumar, A., & Sharma, P. (2021). Dynamic Parent Node Selection in RPL-Based IoT Networks for Enhanced Performance. Journal of Network and Computer Applications, 176, 102931. https://www.sciencedirect.com/science/article/pii/S1084804520304062

[18] Li, H., Chen, W., & Zhao, L. (2023). Adaptive RPL Protocol for Dynamic IoT Environments: A Review and Future Directions. ACM Computing Surveys, 55(1), 1-36. https://dl.acm.org/doi/10.1145/3453182

[19] Moussaoui, M., & Ould-Mokhtar, N. (2022). A Review of RPL Routing Protocol Optimizations for IoT Networks: Energy and Performance Improvements. Sensors, 22(3), 1023. https://www.mdpi.com/1424-8220/22/3/1023

[20] Nguyen, H., & Kim, D. (2021). An Enhanced RPL-Based Routing Protocol for Low-Power IoT Networks. IEEE Transactions on Communications, 69(6), 4194-4205. https://ieeexplore.ieee.org/document/9398764

[21] Reddy, K., & Kumar, A. (2023). Optimizing RPL Routing Protocol for High-Density IoT Networks. IEEE Access, 11, 42354-42368. https://ieeexplore.ieee.org/document/10104256.