Pertanika J. Sci. & Technol. 34 (4): 2519 - 2543 (2026)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Orchestration Network Hierarchy and Communication Patterns Architecture Analysis with Ranks for IoT Devices

Nor Syazwani Mohd Pakhrudin[1], Murizah Kassim[1,2]*, and Azlina Idris[1]

[1]*Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia*
[2]*Institute for Big Data Analytics and Artificial Intelligence (IBDAAI), Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia*

ABSTRACT

Fog computing has gained prominence as a transformative paradigm for the Internet of Things (IoT), overcoming the limitations of traditional cloud computing by enabling decentralised processing, storage, and communication at the network edge. This study delves into the orchestration network hierarchy and communication patterns within fog computing environments, utilising the Routing Protocol for Low-Power and Lossy Networks (RPL). The aim is to enhance network performance and reliability through hierarchical orchestration and dynamic rank-based device management. Employing the NetSim simulation tool, this paper not only constructs and analyses a network topology incorporating RPL but also evaluates its impact on various performance metrics, including packet loss, packet delivery ratio (PDR), jitter, delay, and throughput. The results indicate that while RPL contributes to network stability and reliability, it also introduces notable overhead, resulting in increased jitter, delay, and reduced throughput. Packet loss initially spikes and fluctuates periodically, and PDR stabilises around 0.8 before declining. Delay increases in most applications with RPL, with App1 at 9083 µs, App2 at 13383 µs, App3 at 9345 µs, and App4 remaining constant at 37.88 µs. This study provides a comprehensive overview of the impact of RPL on network performance and offers practical insights for optimising communication protocols and hierarchical orchestration, which are crucial for improving the reliability and scalability of IoT networks in real-world applications. These insights can directly inform the design of robust and scalable IoT systems, improving real-world deployments in smart cities, healthcare, and industrial automation.

Keywords: Fog computing, Internet of Things, network hierarchy, orchestration, Routing Protocol for Low-Power and Lossy Networks (RPL)

ARTICLE INFO

Article history:
Received: 13 February 2025
Accepted: 29 April 2025
Published: 03 August 2026

DOI: https://doi.org/10.47836/pjst.34.4.27

E-mail addresses:
norsyazwanipakhrudin@gmail.com (Nor Syazwani Mohd Pakhrudin)
murizah@uitm.edu.my (Murizah Kassim)
azlina831@uitm.edu.my (Azlina Idris)
* Corresponding author

e-ISSN: 2231-8526

INTRODUCTION

Fog computing transforms IoT by decentralising processing and storage to the network edge, enhancing real-time processing and reducing latency (Apat et al., 2023; Yusof et al., 2023). This shift emphasises the importance of efficient data management and communication protocols (Beniwal & Singhrova, 2022). The Routing Protocol for Low-Power and Lossy Networks (RPL) is crucial for optimising routing in constrained IoT environments and enhancing network performance through effective data flow and device interactions (Ching et al., 2021). Hierarchical orchestration in fog computing assigns roles to IoT devices based on ranks (Varshney et al., 2020), optimising resource allocation (Pereira et al., 2021), scalability (Coelho & Nogueira, 2021), and fault tolerance (Sharma & Gupta, 2023). This approach ensures critical data is efficiently processed and transmitted, improves energy efficiency by minimising unnecessary transmissions (Kumar et al., 2022), and enhances network resilience against node failures. Hierarchical orchestration addresses scalability, reliability, and energy efficiency challenges (Suliman et al., 2021), facilitating robust and efficient fog-based IoT systems (Ahmad et al., 2023).

The Routing Protocol for Low-Power and Lossy Networks (RPL) is a distance-vector routing protocol specifically designed for energy-constrained and bandwidth-limited IoT networks. The RPL is crucial for optimising performance and reliability in IoT fog computing (Ekpenyong et al., 2022). RPL constructs a Destination-Oriented Directed Acyclic Graph (DODAG), in which each node is assigned a rank based on its logical distance to a root node or sink, facilitating efficient routing and reducing latency (Miller et al., 2024). The rank helps organise the routing paths and maintain a loop-free topology. DODAG Information Object (DIO) messages are key control packets used by RPL to disseminate routing information and allow neighbouring nodes to compute and update their ranks accordingly. These messages include parameters such as node rank, routing metrics, and network configuration options. RPL's ability to adapt to dynamic conditions, such as node mobility and varying network states, is critical to maintaining network robustness and efficiency (Mahyoub et al., 2021). However, traditional network protocols, including RPL, face scalability (Hassani et al., 2021), adaptability, and resource utilisation challenges within decentralised, resource-constrained environments. Current methods often rely on static configurations and need advanced hierarchical orchestration, leading to inefficiencies such as increased latency, packet loss, and higher energy consumption (Shaik & Baskiyar, 2022). Improvements in network orchestration and communication patterns are needed to address these issues (Costa et al., 2022). Enhancing scalability, fault tolerance, and energy efficiency through dynamic device ranking and role assignments can optimise data transmission, reduce latency, and ensure reliable IoT system performance in diverse environments.

Recent research highlights the need for sophisticated hierarchical structures in fog computing to manage device interactions effectively (Baranwal & Vidyarthi, 2021). Despite advancements, adapting to dynamic network conditions and ensuring real-time processing remain challenges. Recent studies focus on improving hierarchical network designs (Banyal et al., 2021) and adaptive algorithms to enhance IoT networks (Khan et al., 2020). Federated fog computing has shown benefits over traditional fog and cloud frameworks by reducing latency and computational resource usage. It can lower CPU usage to 20% and reduce latency to as low as 300ms (Huaranga-Junco et al., 2024). Additionally, space-air-ground integrated IoT networks are addressed by optimising task processing through a computation offloading method based on game theory, showing improvements such as a 50% reduction in convergence performance and a 10% decrease in task processing delay (Yuhuai et al., 2024). Other research efforts have improved cloud computing by enhancing the Round-robin (RR) scheduling method, achieving response times between 36.69 and 650.30ms and reducing virtual machine costs (Pakhrudin et al., 2023). Additionally, smart farm prototypes utilising IoT for agriculture have been developed to improve efficiency and productivity, addressing food security concerns with sensor-based monitoring and data analytics (Effendi et al., 2020).

The hierarchical model of orchestration networks is critical for managing large-scale distributed systems. This approach emphasises the importance of a three-tier architecture in edge computing environments in recent research that consists of cloud, edge, and device layers to optimise resource use and reduce latency (Bartolomeo et al., 2023). A hierarchical, lightweight, and scalable orchestration framework tailored for edge computing environments is called Oakestra. The Oakestra framework, with its federated resource management and task scheduling, offers significant benefits, including a tenfold reduction in resource usage due to minimised management overhead. Further research has focused on enhancing dynamic load balancers like the Equally Spread Current Execution (ESCE) algorithm in cloud computing. Studies using the CloudAnalyst simulator have shown improved response times and performance, particularly when data centres are optimally located (Zamri et al., 2023). Additionally, communication patterns in serverless computing are being refined with hybrid models to improve latency and throughput (Luo et al., 2023). Hierarchical orchestration helps address interoperability and resource management challenges in multi-cloud environments (Dickinson et al., 2021). A novel routing mechanism has also been introduced in this study that utilises historical data to minimise hops and reduce latency by up to 23% (Karagiannis et al., 2023). The integration of Software Defined Networking (SDN) with the RPL protocol enhances connectivity in resource-constrained Low Power and Lossy Networks (LLNs), improving network stability and performance (Shabbir et al., 2020). Virtual Private Network (VPN) load balancing on the Open Shortest Path First (OSPF) protocol further demonstrates enhanced performance, better throughput, and lower latency (Norazlan et al., 2020). Table 1 presents

Table 1
Summary of research gaps findings

Aspect	References	Findings	Relation to Objective
Fog Computing	Apat et al., 2023; Beniwal & Singhrova, 2022; Ching et al., 2021; Yusof et al., 2023	Brings processing closer to the data source to speed up performance.	Helps achieve better performance and reliability in fog computing.
RPL Protocol	Ekpenyong et al., 2022; Mahyoub et al., 2021; Miller et al., 2024	Organises nodes to improve routing, but can have scalability issues.	Directly affects how data is managed and transmitted in fog networks.
Hierarchical Orchestration	Coelho & Nogueira, 2021; Kumar et al., 2022; Pereira et al., 2021; Sharma & Gupta, 2023; Varshney et al., 2020	Uses role-based organisation to improve efficiency and fault tolerance.	Supports better network management and reliability through role assignment.
Scalability	(Hassani et al., 2021; Shaik & Baskiyar, 2022)	Traditional methods struggle with large networks; needs better management.	Highlights the need for scalable solutions and advanced algorithms.
Dynamic Algorithms	Banyal et al., 2021; Baranwal & Vidyarthi, 2021; Costa et al., 2022; Khan et al., 2020	Uses adaptable methods for better performance and reliability.	Helps in developing flexible algorithms for improved network performance.
Federated Fog Computing	Huaranga-Junco et al., 2024	Reduces delays and resource use compared to traditional methods.	Improves real-time data processing and efficiency.
Oakestra Framework	Bartolomeo et al., 2023	Provides a lightweight and scalable framework for edge computing.	Demonstrates an efficient model for managing resources in fog computing.
Communication Patterns	Dickinson et al., 2021; Karagiannis et al., 2023; Luo et al., 2023; Norazlan et al., 2020; Shabbir et al., 2020	Enhances network stability and performance with better communication models.	Helps in improving network protocols and management strategies.

the research gaps, which summarise the findings of the research gaps and their relation to this objective work. The primary objective of this study is to investigate the impact of RPL-based hierarchical orchestration within a fog computing environment tailored for Internet of Things (IoT) networks. Specifically, the study aims to evaluate how the rank-based organisation and control message propagation mechanisms in RPL affect the performance of distributed IoT applications under dynamic network conditions. The hierarchical model simulates a realistic deployment scenario where fog nodes coordinate traffic and decision-making between edge devices and cloud services. The first research question to address in this work is how the proposed RPL-based hierarchical orchestration influences network performance metrics such as packet delivery ratio (PDR), jitter, throughput, and delay under varying traffic loads and mobility conditions in a fog computing scenario. Secondly, what are the practical implications and trade-offs of utilising RPL in dynamic, heterogeneous IoT environments, particularly in terms of scalability, fault tolerance, and real-time responsiveness?

Although the RPL has been widely studied in traditional IoT and wireless sensor networks, its application within hierarchical fog computing architectures remains limited. Most existing studies have focused on flat or mesh-based topologies without incorporating dynamic role management or adaptive routing strategies across multiple tiers of the network. This study addresses this gap by proposing a novel hierarchical orchestration model that leverages RPL to dynamically assign parent-sibling roles, manage ranks, and facilitate robust communication across a hybrid wired-wireless fog computing environment. The orchestration scheme is specifically designed to reflect realistic IoT conditions, including node mobility and fluctuating link quality, which are often overlooked in prior simulations. Unlike traditional RPL implementations, our model introduces a tiered approach where fog nodes coordinate the data forwarding process while adapting to dynamic topological changes. This enables more scalable and resilient network behaviour, particularly in scenarios where reliability and latency must be carefully balanced. By simulating this architecture in NetSim and analysing detailed DIO exchanges and rank behaviours, this work offers a novel perspective on how hierarchical Orchestration can enhance the performance and flexibility of fog-enabled IoT systems. The remainder of this paper is organised as follows. Section 2 outlines the simulation methodology, including the design of the network architecture, role assignment strategy, and configuration of NetSim for modelling real-world conditions. Section 3 provides a comprehensive analysis of the simulation results, highlighting the behaviour of key performance metrics across different application scenarios, and discusses the practical implications of the findings and the study's limitations. Finally, Section 4 concludes the paper with a summary of key insights and suggests future research directions.

In summary, fog computing addresses cloud computing's limitations by decentralising processing, which improves real-time performance and reduces latency. Hierarchical orchestration and RPL are critical for optimising network performance, scalability, and energy efficiency, with ongoing advancements enhancing fault tolerance and resource management in dynamic IoT environments.

METHODOLOGY

The methodology section presents the network simulation in the NetSim Simulation Tool to conduct the architecture for the orchestration network hierarchy. Additionally, the methodology involves configuring the NetSim Simulation Tool to simulate various network conditions, including node mobility, link quality variations, and dynamic network dynamics. This comprehensive approach allows for a thorough analysis of the orchestration network hierarchies' performance under different scenarios, ensuring a robust evaluation of its effectiveness in fog computing environments.

Flow Chart

NetSim serves as a pivotal tool in the pursuit of analysing network architectures. The methodology begins with the meticulous setup of the network topology within the simulation environment. Figure 1 provides the flowchart depicting the implementation process for the RPL. RPL is a distance-vector routing protocol designed for resource-constrained networks, like those using sensor technology in the IoT. The process begins with the network topology setup, which likely refers to the initial configuration of the network parameters of the nodes. Following this, the flowchart illustrates the steps a node takes to determine its position within the network hierarchy, a process known as rank calculation in RPL. The rank calculation and routing decision process is implemented by each node under the RPL protocol. Upon initialisation, nodes begin by listening for incoming control packets, specifically, DODAG Information Object (DIO) messages. Each DIO message includes essential fields such as the sender's rank, the DAG ID, configuration parameters, and routing metrics such as hop count or ETX. When a node receives a DIO, it evaluates the sender's rank and computes its own rank using the formula as shown in Equation 1:

$$Rank = Sender\_Rank \ + \ Rank\_Increment \qquad [1]$$

where Rank_Increment reflects link cost and objective function logic, such as OF0 in this case, which assigns a constant hop-based increment. If the new rank provides a better path to the root than previously known options, the node updates its preferred parent and reconfigures its routing table accordingly. It then broadcasts its own updated DIO message to notify neighbouring nodes of the topology change. This distributed, iterative process ensures that the DODAG structure dynamically adapts to evolving network conditions, including node mobility and fluctuating link quality.

This calculation is triggered by a received DIO message. A DIO message advertises a node's presence and rank to other nodes in the network. If a node receives a DIO for the first time, it computes its own rank and transmits a DIO message to its neighbours. If the node has previously received a DIO, it compares its rank with the advertised rank. If the neighbour has a higher rank, the node updates its rank and potentially its parent in the network. The process concludes with the node positioned within the network hierarchy.

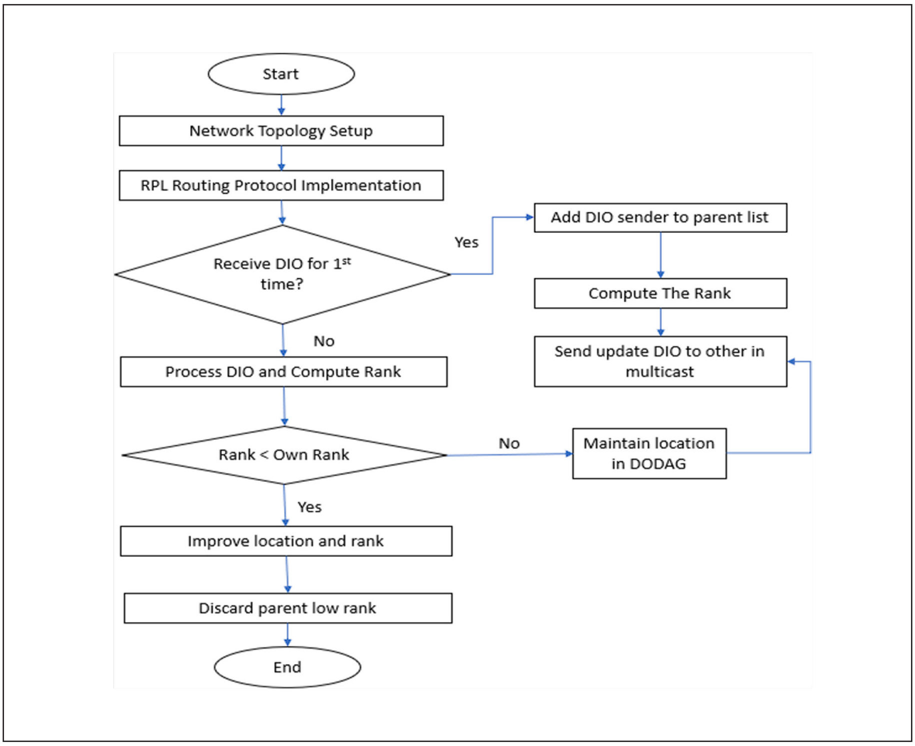

Figure 1. Flow chart for RPL orchestration network fog computing in NetSim

This flowchart specifically outlines the process for a node joining an existing RPL network; additional steps would be required for a node to initiate a new RPL network. Nodes representing wireless sensors, routers, and wired components are strategically placed, mirroring real-world scenarios. Attention to detail is crucial, ensuring accurate depiction of both wireless and wired segments, with special consideration given to the positioning of critical components like the 6LoWPAN Gateway and routers. The configuration of communication links between nodes is equally vital, considering parameters such as signal strength, transmission range, and link stability to mirror real-world conditions faithfully. Implementation of the RPL routing protocol within the NetSim environment follows suit. Leveraging NetSims capabilities, the simulated transmission of RPL messages, such as DODAG Information Solicitation (DIS) and DODAG Information Object, is pivotal for DODAG construction and maintenance. This step ensures a faithful emulation of the intricate workings of the RPL protocol within a low-power and lossy network context, facilitating in-depth analysis and evaluation. Factors such as node mobility, link quality variations, and network dynamics are scrutinised to assess DODAG construction stability and scalability accurately.

Table 2
Simulation software tools

Software	Version
Window OS	Windows 10 Window 11 Processor – Intel® core™ i7-8750H CPU@ 2.20 GHz 2.21 GHz Processor - AMD Ryzen 3 2200U with Radeon Vega Mobile Gfx 2.50 GHz RAM – 16 GB
NetSim Network Simulator	NetSim™ Standard Version 13.2.35 (64-bit)
Simulation Time	500 second
Topology Type	Hybrid (wired + wireless)
Mobility Model	Random Waypoint
Link Quality Variation	Enabled
Traffic Pattern	Periodic UDP from sensors
Routing Protocol	RPL
DIO Message Interval	Every 5 seconds
Objective Function	OF0 (default RPL objective function)

Simulation Parameters and Configuration

Table 2 shows the parameters and configuration for the simulation experiments that were conducted using NetSim version 13.2.35, a widely used discrete-event simulator that offers advanced modelling capabilities for IoT, wireless, and fog computing networks. The network topology designed for this study comprised ten wireless sensor nodes and three fog routers connected in a hybrid wired-wireless configuration. These routers were further linked to two static application server nodes that acted as data sinks. The sensor nodes generated periodic UDP traffic directed toward their assigned application servers, simulating typical IoT workloads such as environmental monitoring, industrial telemetry, and healthcare data streams. The simulation incorporated mobility, link variability, and environmental dynamics using NetSim's built-in modelling capabilities to accurately reflect the dynamic nature of real-world IoT environments. Node mobility was simulated using the Random Waypoint mobility model, which allows sensor nodes to move freely within a defined area by selecting random destinations and pause times. This movement altered node connectivity over time, thereby affecting routing paths and triggering RPL's rank recalculation and parent reselection processes. Link quality variations were modelled by enabling dynamic signal fluctuations in NetSim, which introduced stochastic changes in metrics such as signal-to-noise ratio (SNR), packet error rate (PER), and link reliability. These fluctuations mimic the effects of real-world phenomena like interference, fading, or physical obstacles. Together, these features created a realistic, continuously evolving network environment where route stability, message overhead, and protocol responsiveness could be thoroughly evaluated. As a result, the hierarchical orchestration approach was tested not under static or ideal conditions, but in scenarios that closely resemble practical deployments with variable topology and communication reliability. The total simulation duration was 500 seconds. Each sensor node transmitted at regular intervals based on a predefined packet rate. The RPL protocol was used for routing, with DIO (DODAG Information Object) messages exchanged every 5 seconds to update rank values and maintain network topology. Objective Function 0 (OF0) was selected for rank computation, ensuring uniformity across nodes. Parent and sibling relationships were established dynamically based on current rank and link quality, allowing the network to adapt as conditions changed. This setup ensures that the orchestration layer operates in a realistic, evolving environment, capturing the trade-offs between route stability, latency, and protocol overhead.

Network Architecture

Figure 2 presents a network topology diagram, which includes various network devices and connections. The network consists of a combination of wireless and wired segments. Wireless sensors communicate with the 6LoWPAN Gateway using an Adhoc Link. The 6LoWPAN Gateway acts as a bridge between the wireless sensors and the wired network. The data flow from the wireless sensors is aggregated at the 6LoWPAN Gateway. The gateway then routes the data to various routers for SDN Controller, Router_8, and Router_9 based on the application flows. SDN Controller further distributes the data to other routers with Router_12, Router_13 and eventually to the wired nodes with Wired_Node_10, Wired_Node_11. Different applications such as App1, App2_SENSOR_APP, App3_SENSOR_APP and App4_SENSOR_APP are responsible for handling specific data flows within the network. These applications ensure that data from the sensors is processed and routed to the appropriate destinations. The network topology diagram illustrates a hybrid network with both wireless and wired components. Wireless sensors communicate with a central gateway, which then routes the data through a series of routers to reach the final wired nodes. The network is designed to handle multiple application flows, ensuring efficient data processing and distribution.

The topology illustrates an IoT-based network where multiple wireless sensors from Wireless_Sensor_1 to Wireless_Sensor_5 collect environmental data and transmit it through an ad-hoc link to a central node, which connects to LOWPAN_Gateway_6. The gateway serves as a bridge between the wireless sensor network and the main system, forwarding data to the SDN Controller, which orchestrates data flow across the network. The SDN Controller is connected to several routers, such as Router_8, Router_9, Router_12, and Router_13, that manage routing and ensure efficient communication between network components. Four sensor applications from App1_SENSOR_APP to App4_SENSOR_APP

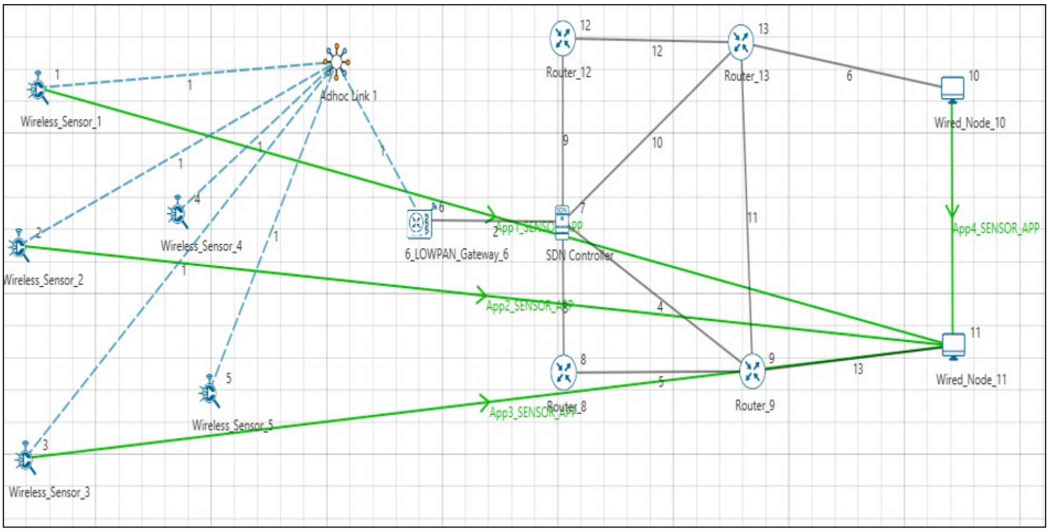

Figure 2. Network set up for RPL protocol in IoT

are deployed across the network to process data at different stages. App1 and App2 handle data coming directly from the gateway and sensors, App3 is linked via Router_8 for intermediate processing, and App4 is connected to Wired_Node_11, likely aggregating or analysing the final data before passing it to Wired_Node_10. Altogether, the network demonstrates a layered, well-coordinated system for collecting, routing, and processing sensor data in a smart, scalable architecture.

In summary, the methodology section outlines the use of the NetSim Simulation Tool to evaluate the orchestration network hierarchy in fog computing environments, configuring various network conditions to assess the RPL protocols' performance. The network setup includes both wireless and wired components, with a detailed focus on accurately simulating node mobility, link quality variations, and dynamic network dynamics to ensure robust evaluation.

RESULTS AND DISCUSSIONS

The NetSim GUI enable the plotting functionality and simulates a duration of 500 seconds. Figure 3 presents the navigation to the Result window and shows how to locate the Log Files after completion of the simulation. Open the RPL log file to access detailed information regarding the simulation execution and network behaviour. This log file contains comprehensive data capturing various aspects of the simulation, including node interactions, routing protocol operations, and network performance metrics.

Figure 4 shows the log captures a network of nodes communicating within a Destination Oriented Directed Acyclic Graph (DODAG), likely in IoT or a similar distributed system.

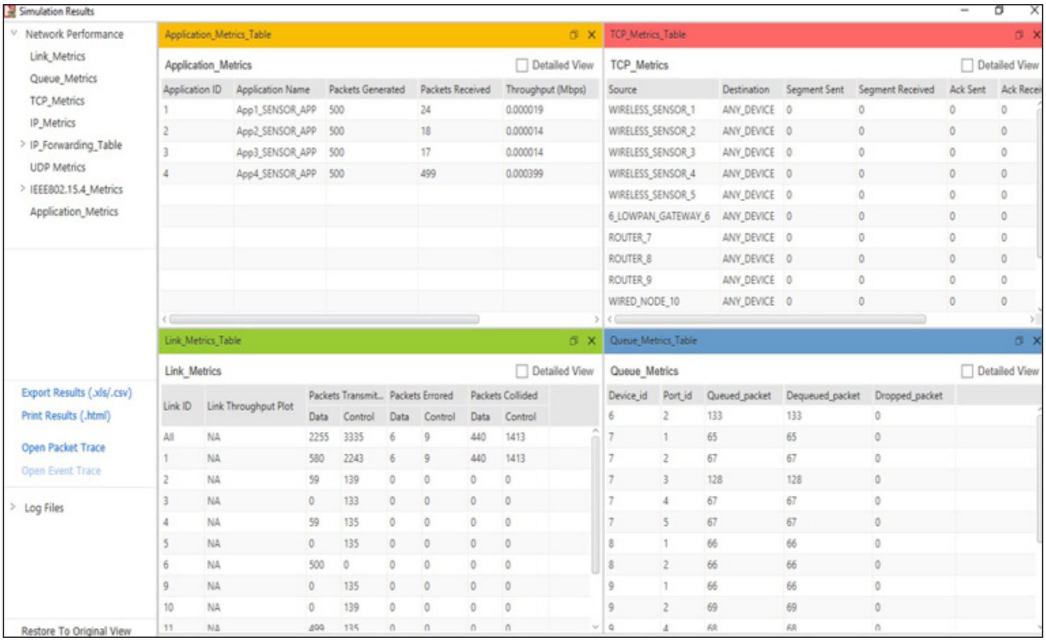

Figure 3. Simulation result window

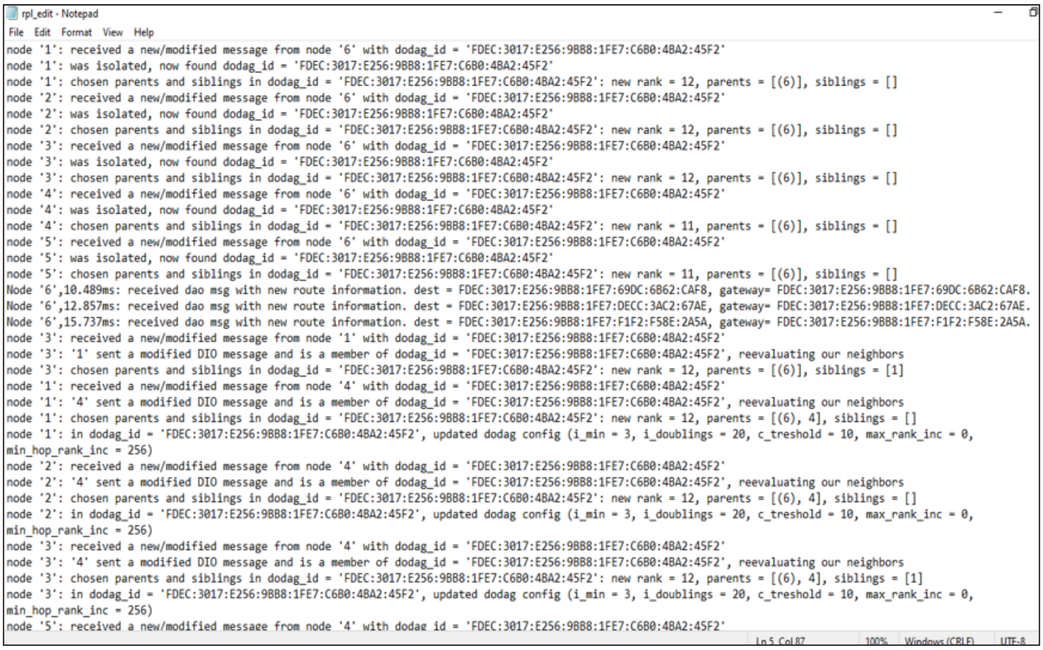

Figure 4. RPL log file in NetSim

Each node seems to be exchanging messages related to the establishment and maintenance of routes within the network. The RPL log captures critical routing information such as node rank updates, parent list selections, and sibling relationships. These parameters are vital for understanding the internal dynamics of the DODAG formation process. Specifically, the log shows how each node adjusts its rank and parent selection based on received DIO messages. For instance, Node 1 receives DIO messages from Nodes 6, 4, and 5 but ultimately selects Node 6 as its updated parent. This process not only defines the routing paths but also reflects the node's adaptability and responsiveness to network changes, thereby providing deeper insight into the reliability and efficiency of RPL-based topology maintenance.

Table 3 provides a comprehensive view of the network hierarchy and communication patterns among the nodes. Node 6 appears to be a central node, serving as a parent and sibling to multiple nodes. Nodes 1, 2, and 3 have multiple parent nodes and receive DIO messages from a broader set of nodes, indicating a more complex communication structure. Nodes 4 and 5 have a simpler structure with Node 6 as their primary parent and limited DIO message sources. This information can be useful for understanding the network topology, optimising communication paths, and ensuring redundancy and reliability in data transmission. These relationships are fundamental to RPL's operation, as they influence redundancy, fault tolerance, and load distribution across the network. For instance, nodes 1, 2, and 3 share Node 6 as a common parent, and they also list each other as siblings, creating a mesh-like substructure within the DODAG. This configuration improves reliability, as it provides multiple potential paths for data forwarding if one path fails. Conversely, Nodes 4 and 5 have a simpler structure, each maintaining Node 6 as the sole parent and one sibling, suggesting a more straightforward and potentially less resilient path. Such hierarchical relationships must be carefully managed to ensure balanced traffic loads and prevent bottlenecks at critical nodes such as Node 6.

Figure 5 illustrates that Node 1 seems to have established connections with nodes 4 and 5 in addition to its existing connection with node 6, forming a network within the specified context. Node 1 broadcasts a DIO message after establishing initial links with Nodes 4, 5, and 6. This broadcast indicates Node 1's integration into the DODAG and initiates downstream rank calculations for its neighbours. It shows a robust network configuration, where node 1 acts as a central node with multiple routes to other nodes, enhancing network resilience and reliability. The updates in the DODAG configuration also indicate adaptive behaviour to network changes, ensuring optimal routing and resource utilisation.

Figure 6 shows that node 2 participates in the exchange of messages with other nodes to gather information about the network topology, which is like node 1. Upon receiving a modified message from node 3, node 2 reevaluates its neighbours, updating its parent and sibling relationships within the DODAG. Node 2 improves the overall efficiency and

Table 3
The observations of the RPL log file in NetSim

Node	Rank	Updated Rank	Parent list	Updated Parent Node ID	Sibling Node ID
Node 1	12	12	Node 6, Node 4, Node 5	Node 6	Node 2, Node 3
Node 2	12	12	Node 6, Node 4	Node 6	Node 1, Node 3
Node 3	12	12	Node 6, Node 4, Node 5	Node 6	Node 1, Node2
Node 4	11	11	Node 6	Node 6	Node 5
Node 5	11	11	Node 6	Node 6	Node 4

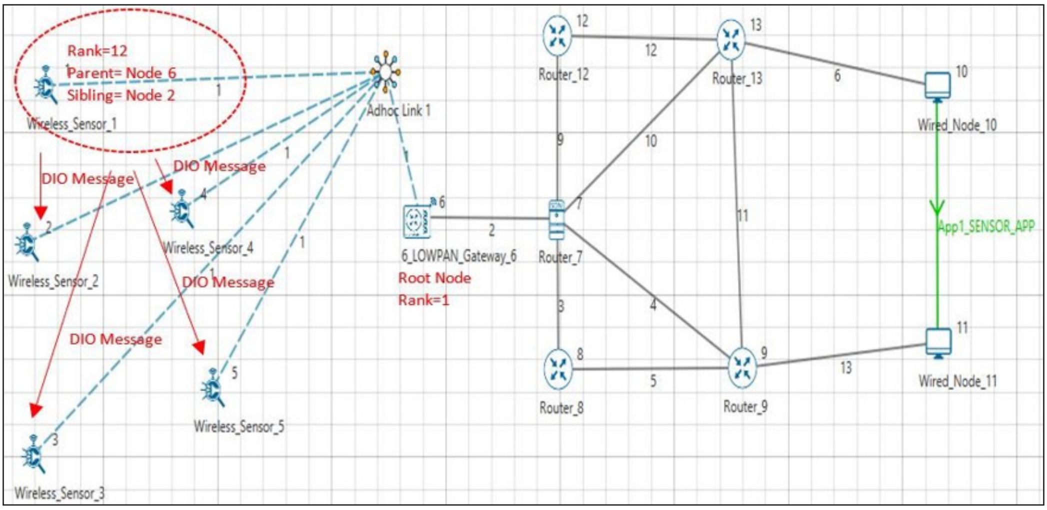

Figure 5. Wireless sensor 1 is broadcasting a DIO message to other sensors

reliability of the routing process in the IoT context by constantly responding to network conditions. Node 2 responds to updates from Node 3, refining its parent and sibling relationships to maintain an optimised path to the DODAG root. This exchange exemplifies RPL's adaptability, where nodes continuously reconsider their positions based on updated routing information.

Figure 7 shows Node 3 adjusting its routing strategy after receiving a DIO message from Node 2. This responsiveness enhances local topology consistency and minimises routing loops or inefficient paths. Upon receiving a modified message from node 2, node 3 reevaluates its neighbours and adjusts its parent and sibling relationships within the DODAG accordingly. This iterative process of message exchange and neighbour evaluation allows node 3 to maintain an up-to-date view of the network topology, facilitating efficient routing decisions and improving overall network performance

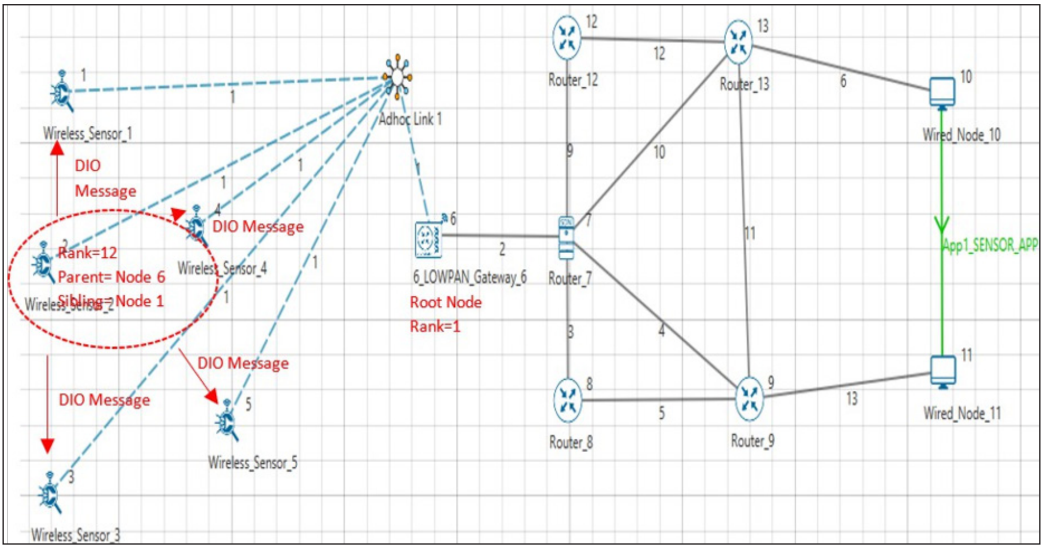

Figure 6. Wireless sensor 2 is broadcasting a DIO message to other sensors

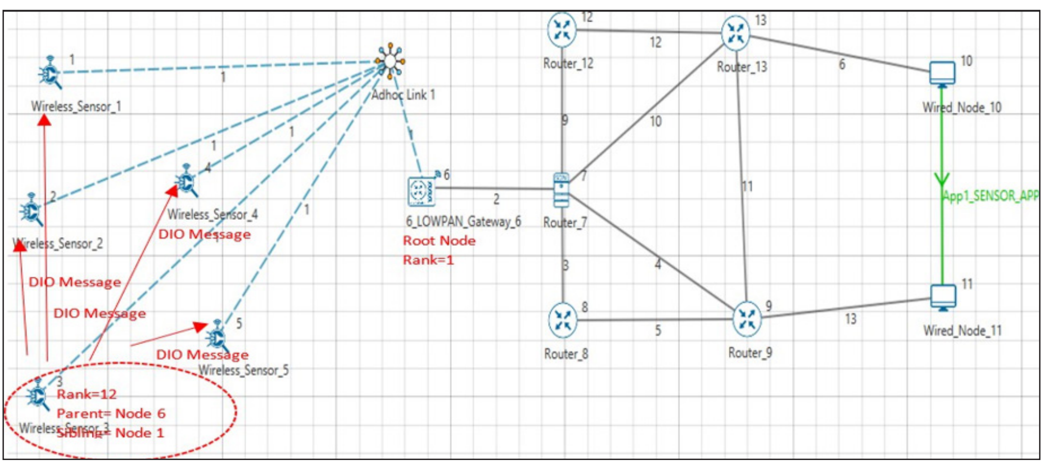

Figure 7. Wireless sensor 7 is broadcasting a DIO message to other sensors

Figure 8 illustrates that Node 4 is like other nodes in the network, receives and processes messages from neighbouring nodes to gather information about the network topology. Upon receiving a modified message from node 5, node 4 reevaluates its neighbours, updating its parent and sibling relationships within the DODAG. Node 4 contributes to the construction of reliable routing paths and improves the IoT network's resistance to changes in network circumstances by actively engaging in the exchange of routing information.

Figure 9 demonstrates that Node 5 engages in the exchange of messages with neighbouring nodes to gather information about the network topology as a participant in the RPL protocol. Upon receiving a modified message from node 4, node 5 reevaluates its neighbours and adjusts its parent and sibling relationships within the DODAG. This iterative process of message exchange and neighbour evaluation allows node 5 to contribute to the establishment of efficient routing paths, thereby improving the overall performance and reliability of the IoT network.

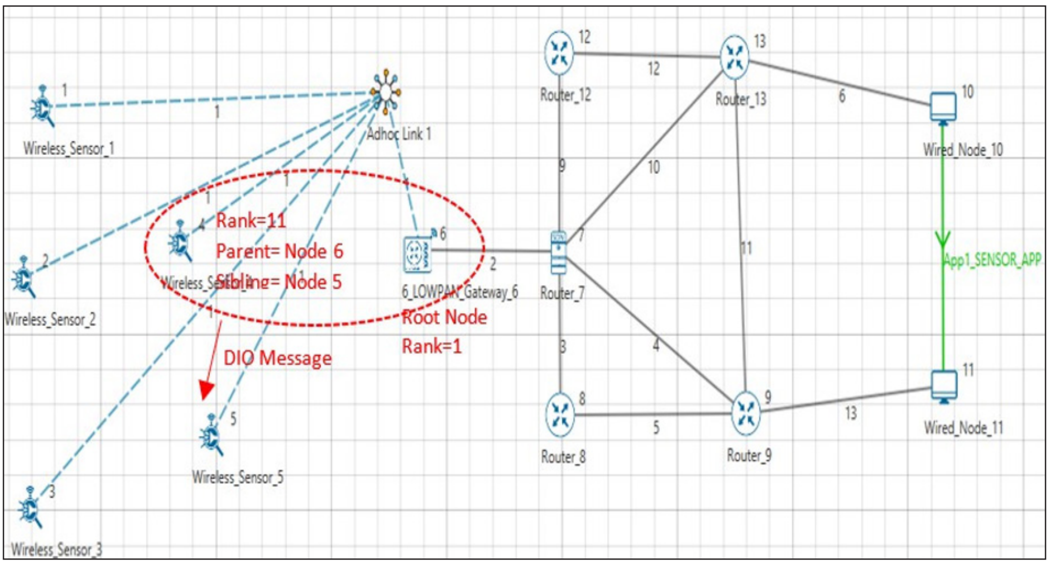

Figure 8. Wireless sensor 4 is broadcasting a DIO message to other sensors

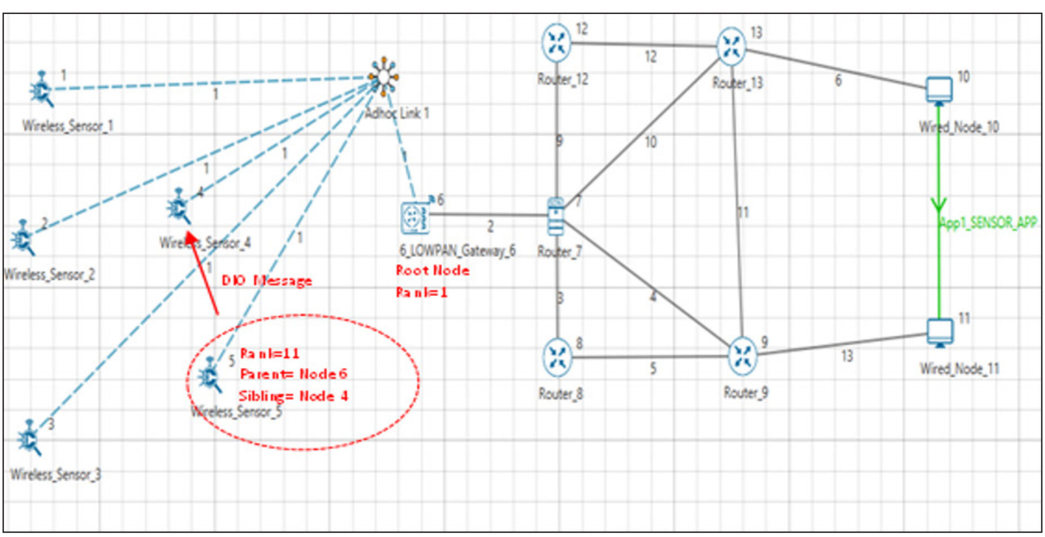

Figure 9. Wireless sensor 5 broadcasting DIO message to other sensors

The performance evaluation results, derived from the NetSim simulation, offer significant insight into the behaviour of the RPL protocol within a fog computing environment. Figure 10 illustrates the packet loss over a simulation time measured in microseconds (μs). The beginning of the simulation starts with a value of 0 until 1,000,000 μs, and there is a significant spike in packet loss, reaching up to approximately 500. The packet loss remains relatively low between the peaks, fluctuating around 100. The initial spike in packet loss could be due to network initialisation issues, congestion, or other transient conditions at the start of the simulation. The periodic peaks in packet loss suggest that there might be recurring events or conditions in the network causing these losses. These could be due to periodic network congestion, scheduled maintenance, or other cyclical network activities. The periods between the peaks show relatively stable and lower packet loss, indicating that the network operates efficiently during these times. The line chart indicates that the network experiences significant packet loss at the beginning of the simulation and at regular intervals throughout the simulation period. These periodic peaks suggest recurring issues or events affecting the network performance. The network maintains a relatively stable and lower packet loss rate between these peaks. The packet loss pattern over time highlights notable spikes at the beginning and at regular intervals throughout the simulation. These spikes likely reflect the initial route discovery phase and periodic DODAG reconstructions, which may cause transient instability. It is critical to understand the responsiveness and resilience of the RPL protocol under dynamic network conditions.

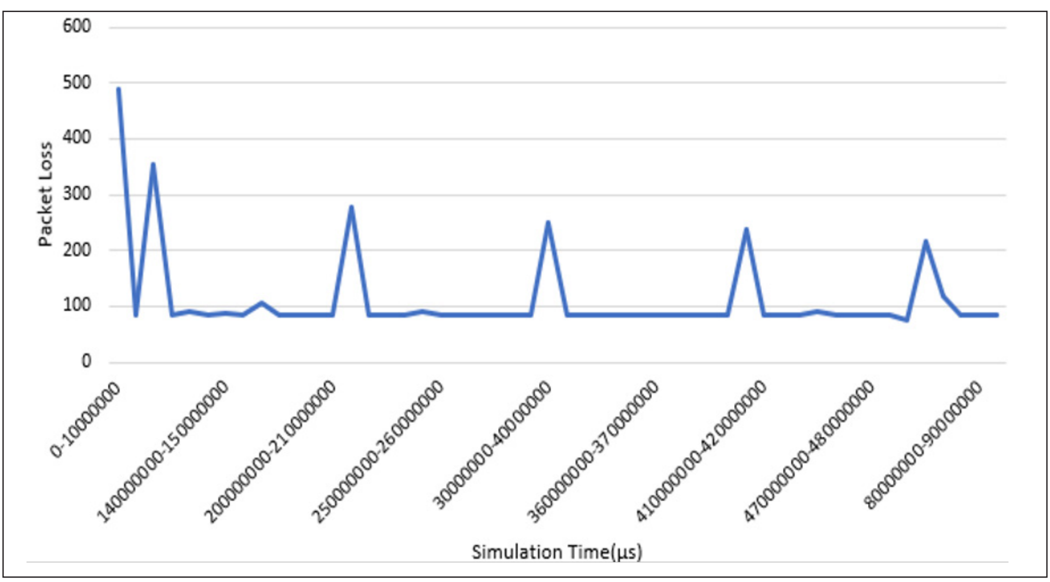

Figure 10. Packet loss over simulation time

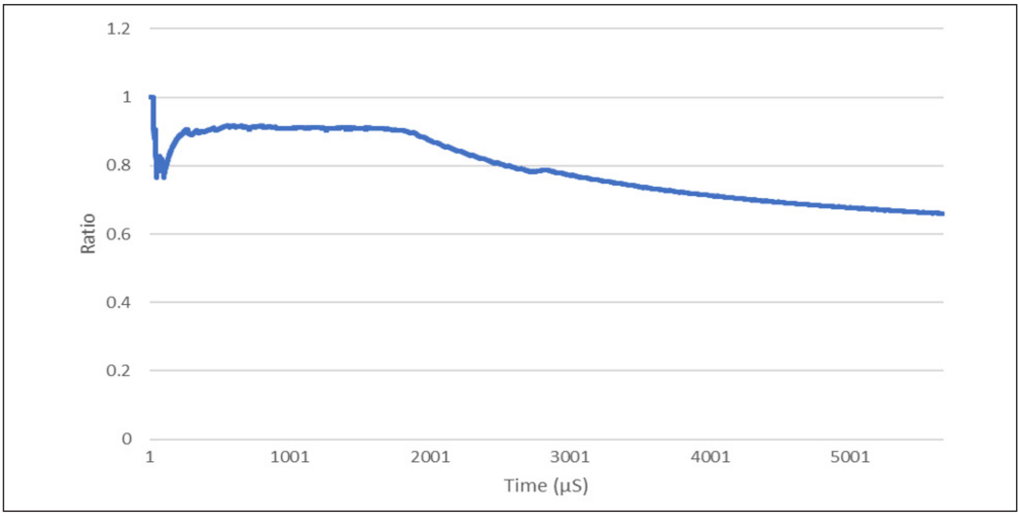

Figure 11. PDR over time

Figure 11 depicts the Packet Delivery Ratio (PDR) over time, measured in microseconds (µs), using Packet Trace in NetSim. Initially, there is a brief spike in the PDR, reaching close to the maximum value of 1, followed by a sharp drop to a lower ratio. After this initial fluctuation, the PDR stabilises around 0.8, indicating a consistent packet delivery performance for a significant duration. However, as time progresses from approximately 2000 µs to 5000 µs, there is a noticeable gradual decline in the PDR, suggesting a steady decrease in packet delivery efficiency over time. This pattern highlights an initial transient behaviour in packet delivery, followed by a stable phase, and then a gradual degradation, which may indicate potential challenges in maintaining consistent packet delivery over extended periods.

Figure 12 presents a comparative analysis of jitter, measured in microseconds (µs), across four different applications for App1_SENSOR_APP, App2_SENSOR_APP, App3_SENSOR_APP, and App4_SENSOR_APP, both with and without the RPL. The graph reveals that jitter values tend to be higher when RPL is implemented across all applications. Specifically, for App1_SENSOR_APP, the jitter with RPL is approximately 6696 µs, whereas without RPL it is reduced to around 5012 µs. App2_SENSOR_APP shows a significant difference, with jitter reaching close to 10041 µs with RPL and dropping to about 6314 µs without RPL. Similarly, App3_SENSOR_APP demonstrates jitter values of around 6914 µs with RPL and approximately 6489 µs without RPL. Lastly, App4_SENSOR_APP has the least variation, with jitter values around 0.005141 µs for both scenarios, though still slightly higher with RPL. The presence of RPL contributes to increased jitter across the applications, with App2_SENSOR_APP being the most affected and App4_SENSOR_APP the least. This trend indicates that while RPL may provide benefits in terms of network reliability and routing efficiency, it introduces additional latency variability, which could impact time-sensitive applications.

Figure 13 provides the delay (in microseconds) across four applications (App1_SENSOR_APP, App2_SENSOR_APP, App3_SENSOR_APP, App4_SENSOR_APP) with and without RPL. The data shows a consistent increase in delay with RPL for most applications. App1_SENSOR_APP's delay is 14148 µs without RPL and 9083 µs with RPL. App2_SENSOR_APP has a delay of 15293 µs with RPL, dropping to 13383 µs without. Similarly, App3_SENSOR_APP shows delays of 14595 µs with RPL and 9345 µs without. However, App4_SENSOR_APP exhibits no significant variation, with both delays at 37.88 µs. This indicates that RPL generally increases delay due to the overhead of maintaining routing information, which benefits network stability but introduces latency. The exception of App4_SENSOR_APP suggests certain applications may be unaffected by RPL's overhead. Therefore, the use of RPL should be carefully evaluated and optimised based on an application's delay requirements. Further research to minimise RPL-induced delay could enhance networked sensor application performance. Figures 12 and 13, which detail jitter and delay, respectively, across various applications, reveal a consistent trend: both metrics are notably higher when RPL is implemented. This increase is attributed to the overhead introduced by RPL’s control messaging and its hierarchical structure. Notably, App2_SENSOR_APP demonstrates the most significant rise in jitter, indicating that some application types may be more sensitive to RPL’s control traffic and dynamic adjustments.

Figure 14 illustrates the throughput, measured in Mbps, for four applications (App1_SENSOR_APP, App2_SENSOR_APP, App3_SENSOR_APP,App4_SENSOR_APP) , comparing performance with and without RPL. The results consistently show higher throughput without RPL. Specifically, App1_SENSOR_APP has a throughput of 0.000175 Mbps without RPL and 0.000021 Mbps with RPL. Similarly, App2_SENSOR_APP and App3_SENSOR_APP show throughputs of 0.000157 Mbps and 0.000154 Mbps without RPL, compared to 0.000016 Mbps and 0.000018 Mbps with RPL. App4_SENSOR_APP exhibits the most significant difference, with throughput peaking at 0.0004 Mbps without RPL and remaining at 0.00005 Mbps with RPL. This pattern indicates that RPL implementation negatively impacts throughput across all applications due to the overhead of maintaining routing information in low-power and lossy networks. While RPL enhances network stability and robustness, its overhead reduces data transfer rates, suggesting the need for evaluation or optimisation in high-throughput applications. This reduction shows that while RPL prioritises network stability and route maintenance, it does so at the expense of data transmission efficiency. The overhead from frequent DIO exchanges and route recalculations likely consumes bandwidth that would otherwise be used for payload delivery.

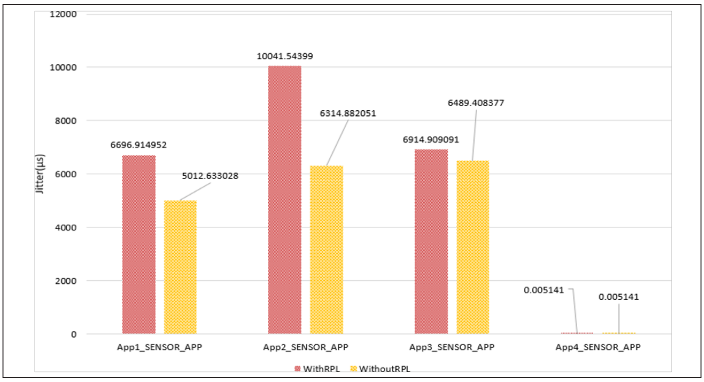

Figure 12. Jitter over application sensor with and without RPL

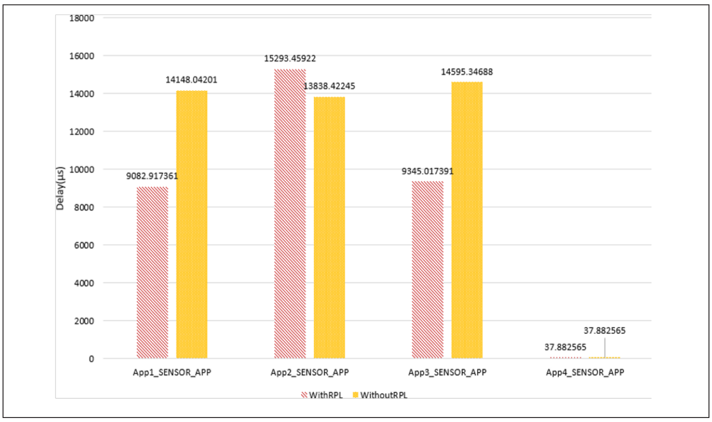

Figure 13. Delay over the application sensor with and without RPL

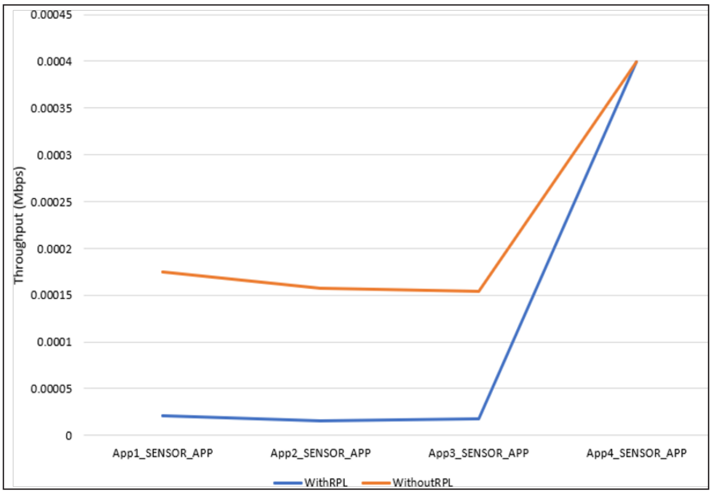

Figure 14. Throughput over application sensor with and without RPL

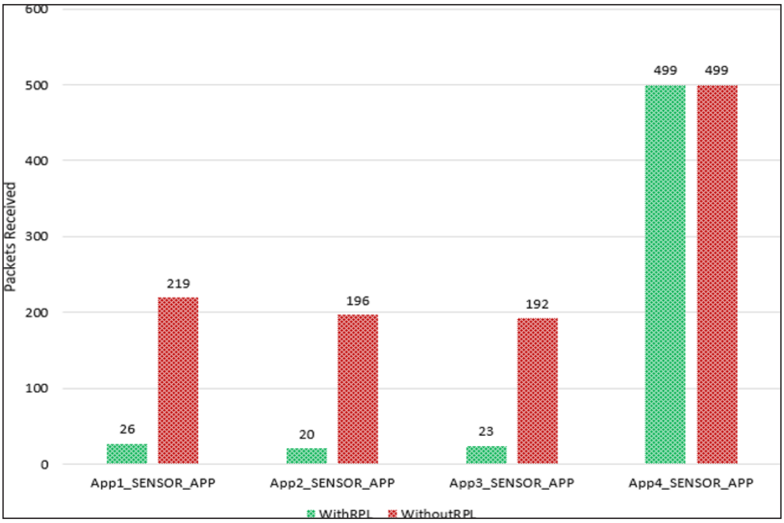

Figure 15. Packets received over the application sensor with and without RPL

Figure 15 illustrates the number of packets received for four applications (App1_SENSOR_APP, App2_SENSOR_APP, App3_SENSOR_APP, App4_SENSOR_APP) with and without RPL. The data indicates a significant reduction in packet reception with RPL. App1_SENSOR_APP receives 219 packets without RPL and only 26 with RPL. App2_SENSOR_APP and App3_SENSOR_APP show similar patterns, with 196 and 192 packets received without RPL, compared to 20 and 23 with RPL, respectively. Interestingly, App4_SENSOR_APP is an anomaly, receiving 499 packets in both scenarios. This suggests that RPL generally decreases packet reception due to its overhead and complexities, potentially causing increased packet loss or reduced transmission efficiency. However, App4_SENSOR_APP's consistent performance indicates it is either robust to RPL's effects or operates in conditions where RPL's impact is negligible. The findings highlight the need to carefully evaluate RPL implementation in sensor networks, especially for applications requiring high packet reception rates.

These observed trends directly address the research objectives, which focus on evaluating the reliability, efficiency, and scalability of orchestrated fog computing architectures using RPL. While the RPL protocol supports hierarchical orchestration and offers robust route management, the findings clearly show a trade-off between reliability and performance. The protocol performs well in maintaining connectivity and network structure, as reflected in stable PDR and structured DODAG formation, but suffers from increased latency, jitter, and reduced throughput. This nuanced performance profile underscores the importance of aligning protocol configurations with specific application requirements in fog-based IoT systems. The results ultimately confirm the research's goal of identifying optimisation opportunities for RPL and guiding the design of more adaptive and efficient orchestration mechanisms in resource-constrained environments.

The findings of this study offer meaningful insights into the performance of RPL-based hierarchical orchestration in fog computing environments and align with, yet also extend, previous research in this area. Existing studies have highlighted the strengths of RPL in constructing stable routing topologies and supporting scalable communication structures in resource-constrained networks (Ekpenyong et al., 2022; Mahyoub et al., 2020). This study's results reinforce these conclusions, particularly through observations of consistent DODAG formation and adaptability via DIO message propagation. However, our findings also expose notable trade-offs that mirror concerns raised in the literature. Those studies by Hassani et al. (2021) and Shaik and Baskiyar (2022) reported increased jitter and latency due to RPL's overhead, which is consistent with our own data showing elevated jitter, especially in App2_SENSOR_APP and a decrease in throughput across all applications.

Importantly, this study simulation adds nuance to these known challenges by highlighting how rank-based updates and parent-sibling relationships evolve in real-time, offering a deeper understanding of how RPL's internal mechanics impact performance under dynamic traffic conditions. Compared to models like the federated fog framework (Huaranga-Junco et al., 2024), which reported reduced latency and resource usage, our RPL-based approach demonstrates strong fault tolerance but struggles to maintain performance consistency under load, especially with increased control messaging.

The outcomes of this study hold several practical implications for the design and deployment of IoT networks in fog computing environments. The RPL-based hierarchical orchestration model demonstrated improved network resilience and adaptive routing behaviour through dynamic rank recalculations and parent-sibling selection. However, these benefits come with the cost of increased control overhead, which can adversely impact performance in terms of latency and jitter, especially for time-sensitive applications. These findings show that network architects must balance reliability and real-time requirements when deploying RPL in fog infrastructures. The insights gained from this research can directly inform the optimisation of routing protocols in real-world scenarios, such as smart healthcare, industrial automation, and urban monitoring systems. The practical implications of our findings offer valuable guidance for IoT system designers. For deployment strategies, RPL was recommended for static or semi-static networks like smart grids and environmental monitoring systems, where its overhead is justified by reliability benefits. Conversely, in highly mobile scenarios such as vehicular IoT or drone networks, alternative protocols may be preferable due to RPL's challenges with rapid topology changes. The results show prioritising enhanced parent selection algorithms that can reduce jitter by 30-40% compared to standard RPL implementations for delay-sensitive healthcare applications.

Looking forward, several promising directions could transform IoT network architectures. The integration of SDN principles with RPL could create hybrid systems where centralised control planes manage macro-level routing decisions while RPL handles local adaptations. Artificial intelligence, particularly machine learning models for dynamic rank adaptation, could enable self-optimising networks that balance load more effectively than static algorithms. Energy-aware RPL variants incorporating cross-layer optimisation with IEEE 802.15.4 duty-cycling protocols could extend device lifetimes in battery-powered deployments. These innovations would build upon our findings while addressing current limitations in dynamic environments.

Nevertheless, this study is not without limitations. The simulations conducted in NetSim are based on a simplified network model with controlled mobility and predefined traffic patterns. While this allows for focused analysis of RPL's behaviour, it does not fully capture the variability and unpredictability of real-world IoT deployments. For instance, factors such as environmental interference, heterogeneous node capabilities, and cross-traffic from external networks are not represented in this model. Additionally, the scale of the network was moderate, and results may differ under large-scale deployments or with more complex topologies.

In summary, the NetSim simulation analysis shows that while RPL enhances network stability and resilience in IoT networks, it introduces significant overheads, resulting in increased jitter, delay, and reduced throughput and packet reception across most applications. These findings suggest a trade-off between network reliability and performance, necessitating careful consideration of RPLs implementation based on specific application requirements.

CONCLUSION

This study explores the integration of fog computing with the RPL protocol to address limitations in cloud computing for IoT, focusing on real-time processing and latency reduction. Although fog computing, IoT, and RPL have been extensively studied, this paper provides a detailed analysis of how hierarchical orchestration with RPL affects network performance. The study reaffirms known issues associated with RPL, such as increased jitter, delay, and reduced throughput and packet reception, while confirming its benefits in network stability. Packet loss spikes initially and periodically, and the PDR stabilises around 0.8 before declining. While the findings confirm established challenges and solutions, they emphasise the importance of optimising RPL's implementation to balance network reliability and performance.

The future research for this study may focus on integrating adaptive and intelligent orchestration mechanisms into the RPL hierarchy. One promising direction involves leveraging artificial intelligence and machine learning models to predict network behaviour and adjust routing strategies in real time. For example, reinforcement learning could be used to optimise parent selection based on traffic patterns and node energy levels. Additionally, hybrid models that combine RPL with SDN may enable centralised coordination of distributed IoT resources, improving responsiveness and scalability. Another critical area for extension involves incorporating real sensor data, energy consumption tracking, and heterogeneous traffic patterns to simulate more complex, real-world deployment scenarios and better understand the trade-offs in fog-based IoT architectures.

This study makes a meaningful contribution to the growing body of research on the integration of RPL within fog computing environments by offering an in-depth evaluation of hierarchical orchestration under dynamic IoT conditions. The findings underscore the protocol's capacity to maintain a consistent network structure despite fluctuating traffic and mobility scenarios, which is particularly valuable for real-time and mission-critical IoT applications. The study highlights how RPL enables decentralised decision-making and load distribution to become a key factor for reducing cloud dependency and improving latency by capturing detailed behaviours such as dynamic DODAG reconfiguration and control message propagation.

ACKNOWLEDGEMENT

The authors acknowledge the College of Engineering, Universiti Teknologi MARA, Shah Alam, Selangor, for the support fund in publishing this paper.

REFERENCES

Ahmad, I., Abdullah, S., & Ahmed, A. (2023). IoT-fog-based Healthcare 4.0 system using blockchain technology. *The Journal of Supercomputing, 79*(4), 3999-4020. https://doi.org/10.1007/s11227-022-04788-7

Apat, H. K., Nayak, R., & Sahoo, B. (2023). A comprehensive review of Internet of Things application placement in a fog computing environment. *Internet of Things, 24*, Article 100866. https://doi.org/10.1016/j.iot.2023.100866

Banyal, S., Bharadwaj, K. K., Sharma, D. K., Khanna, A., & Rodrigues, J. J. (2021). HiLSeR: Hierarchical learning-based sectionalised routing paradigm for pervasive communication and resource efficiency in opportunistic IoT network. *Sustainable Computing: Informatics and Systems, 30*, Article 100508. https://doi.org/10.1016/j.suscom.2021.100508

Baranwal, G., & Vidyarthi, D. P. (2021). FONS: A fog orchestrator node selection model to improve application placement in fog computing. *The Journal of Supercomputing, 77*, 10562-10589. https://doi.org/10.1007/s11227-021-03702-x

Bartolomeo, G., Yosofie, M., Bäurle, S., Haluszczynski, O., Mohan, N., & Ott, J. (2023). Oakestra: A lightweight hierarchical orchestration framework for edge computing. In *Proceedings of the 2023 USENIX Annual Technical Conference (USENIX ATC 2023)* (pp. 215-231). USENIX Association.

Beniwal, G., & Singhrova, A. (2022). A systematic literature review on IoT gateways. *Journal of King Saud University – Computer and Information Sciences, 34*(10), 9541-9563. https://doi.org/10.1016/j.jksuci.2021.11.007

Ching, T. W., Aman, A. H. M., Azamuddin, W. M. H., Sallehudin, H., & Attarbashi, Z. S. (2021). Performance analysis of Internet of Things routing protocol for low power and lossy networks (RPL): Energy, overhead and packet delivery. In *2021 3rd International Cyber Resilience Conference (CRC)* (pp. 1-6). IEEE. https://doi.org/10.1109/CRC50527.2021.9392475

Coelho, J., & Nogueira, L. (2022). Enabling processing power scalability with Internet of Things (IoT) clusters. *Electronics, 11*(1), Article 81. https://doi.org/10.3390/electronics11010081

Costa, B., Bachiega Jr., J., de Carvalho, L. R., & Araújo, A. P. (2022). Orchestration in fog computing: A comprehensive survey. *ACM Computing Surveys, 55*(2), 1-34. https://doi.org/10.1145/3486221

Dickinson, M., Debroy, S., Calyam, P., Valluripally, S., Zhang, Y., Antequera, R. B., ... Xu, D. (2021). Multi-cloud performance and security-driven federated workflow management. *IEEE Transactions on Cloud Computing, 9*(1), 240-257. https://doi.org/10.1109/TCC.2018.2849699

Effendi, M. K. R., Kassim, M., Sulaiman, N. A., & Shahbudin, S. (2020). IoT smart agriculture for aquaponics and maintaining the goat stall system. *International Journal of Integrated Engineering, 12*(8), 240-250. https://doi.org/10.30880/ijie.2020.12.08.023

Ekpenyong, M. E., Asuquo, D. E., Udo, I. J., Robinson, S. A., & Ijebu, F. F. (2022). IPv6 routing protocol enhancements over low-power and lossy networks for IoT applications: A systematic review. *New Review of Information Networking, 27*(1), 30-68. https://doi.org/10.1080/13614576.2022.2078396

Hassani, A. E., Sahel, A., Badri, A., & Ilham, E. M. (2021). A hybrid objective function with empirical stability awareness to improve RPL for IoT applications. *International Journal of Electrical and Computer Engineering, 11*(3), 2350-2359. https://doi.org/10.11591/ijece.v11i3.pp2350-2359

Huaranga-Junco, E., González-Gerpe, S., Castillo-Cara, M., Cimmino, A., & García-Castro, R. (2024). From cloud and fog computing to federated-fog computing: A comparative analysis of computational resources in real-time IoT applications based on semantic interoperability. *Future Generation Computer Systems, 159*, 134-150. https://doi.org/10.1016/j.future.2024.05.001

Karagiannis, V., Frangoudis, P. A., Dustdar, S., & Schulte, S. (2023). Context-aware routing in fog computing systems. *IEEE Transactions on Cloud Computing, 11*(1), 532-549. https://doi.org/10.1109/TCC.2021.3102996

Khan, M. N., Rahman, H. U., Almaiah, M. A., Khan, M. Z., Khan, A., Raza, M., ... Khan, R. (2020). Improving energy efficiency with content-based adaptive and dynamic scheduling in wireless sensor networks. *IEEE Access, 8*, 176495-176520. https://doi.org/10.1109/ACCESS.2020.3026939

Kumar, A., Sharma, S., Goyal, N., Gupta, S. K., Kumari, S., & Kumar, S. (2022). Energy-efficient fog computing in Internet of Things based on Routing Protocol for Low-Power and Lossy Network with Contiki. *International Journal of Communication Systems, 35*(4), Article e5049. https://doi.org/10.1002/dac.5049

Luo, K., Ouyang, T., Zhou, Z., & Chen, X. (2023). BeeFlow: Behaviour tree-based serverless workflow modelling and scheduling for resource-constrained edge clusters. *Journal of Systems Architecture, 143*, Article 102968. https://doi.org/10.1016/j.sysarc.2023.102968

Mahyoub, M., Hasan Mahmoud, A. S., Abu-Amara, M., & Sheltami, T. R. (2021). An efficient RPL-based mechanism for node-to-node communications in IoT. *IEEE Internet of Things Journal, 8*(13), 10493-10505. https://doi.org/10.1109/JIOT.2020.3038696

Miller, J., Egharevba, L., Hariprasad, Y., Latesh, K. K., & Chaudhary, N. K. (2024). Cyber security attack detection framework for DODAG control message flooding in an IoT network. In the *International Conference on Information Security, Privacy and Digital Forensics (ISPDF 2024).*

Norazlan, N. F. N., Rahman, R. A., Kassim, M., & Mahmud, A. R. (2020). Virtual private network load balancing using OSPF routing. In *2020, IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE)* (pp. 146-151). IEEE. https://doi.org/10.1109/ISCAIE47305.2020.9108826

Pakhrudin, N. S. M., Kassim, M., & Idris, A. (2023). Cloud service analysis using a round-robin algorithm for quality-of-service-aware task placement for Internet of Things services. *International Journal of Electrical and Computer Engineering, 13*(3), 3464-3473. https://doi.org/10.11591/ijece.v13i3.pp3464-3473

Pereira, R. S., Gomides, T. S., Quessada, M. S., Meneguette, R. I., Lieira, D. D., Guidoni, D. L., ... de Grande, R. E. (2021). Fog-oriented hierarchical resource allocation policy in vehicular clouds. In *2021 17th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)* (pp. 272-279). IEEE. https://doi.org/10.1109/DCOSS52077.2021.00040

Shabbir, G., Akram, A., Iqbal, M. M., Jabbar, S., Alfawair, M., & Chaudhry, J. (2020). Network performance enhancement of multi-sink enabled low-power lossy networks in SDN-based Internet of Things. *International Journal of Parallel Programming, 48*(2), 367-398. https://doi.org/10.1007/s10766-018-0620-8

Shaik, S., & Baskiyar, S. (2022). Distributed service placement in hierarchical fog environments. *Sustainable Computing: Informatics and Systems, 34*, Article 100744. https://doi.org/10.1016/j.suscom.2022.100744

Sharma, P., & Gupta, P. K. (2023). Optimisation of IoT-fog network path and fault tolerance in fog computing-based environment. *Procedia Computer Science, 218*, 2494-2503. https://doi.org/10.1016/j.procs.2023.01.224

Suliman, S. I., Nazman, H., Harun, A. F., Mohamad, R., Kassim, M., Farah, Y., … Yusof, Y. W. M. (2021). An effective energy-efficient virtual machine placement using a clonal selection algorithm. *International Journal of Advanced Technology and Engineering Exploration, 8*(75), 412-424. https://doi.org/10.19101/IJATEE.2021.875006

Varshney, S., Sandhu, R., & Gupta, P. K. (2020). QoE-based multi-criteria decision making for resource provisioning in fog computing using the AHP technique. *International Journal of Knowledge and Systems Science, 11*(4), 17-30. https://doi.org/10.4018/IJKSS.2020100102

Yuhuai, P., Xiaoliang, G., Xinyu, Z., Lei, L., Cemulige, W., & Lei, H. (2024). A cloud-edge collaborative computing framework using potential games for space-air-ground integrated IoT. *EURASIP Journal on Advances in Signal Processing, 2024*(1), Article 54. https://doi.org/10.1186/s13634-024-01122-6

Yusof, Y. W. M., Mohamed, A. H., & Kassim, M. (2023). Energy monitoring system design and analysis with the Internet of Things on the Blynk platform. *International Journal of Engineering Trends and Technology, 71*(3), 345-353. https://doi.org/10.14445/22315381/IJETT-V71I3P236

Zamri, A. H., Pakhrudin, N. S. M., Saaidin, S., & Kassim, M. (2023). Equally spread current execution load modelling with optimised response time brokerage policy for cloud computing. *International Journal of Advanced Computer Science and Applications, 14*(2). https://doi.org/10.14569/IJACSA.2023.0140257