Routing Optimization for Energy Efficiency in Software-Defined IoT and RPL Networks

Amir Gasouma (amirkku@gmail.com)
Universiti Teknikal Malaysia - Main Campus: Universiti Teknikal Malaysia Melaka
Kamaludin M. Yusof
Universiti Teknikal Malaysia - Main Campus: Universiti Teknikal Malaysia Melaka
Belgacem Bouallegue
King Khalid University
Abdelmoty M. Ahmed
King Khalid University
Safaa S. Matter
King Khalid University

Research Article

Keywords: Energy Efficiency, Objective Function, IoT, RPL, SDN, Routing Optimization

Posted Date: July 31st, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3125883/v1

Routing Optimization for Energy Efficiency in Software-Defined IoT and RPL Networks

Amir Gasouma[1,2], Kamaludin M. Yusof [1], Belgacem Bouallegue[2], Abdelmoty M. Ahmed[2] , Safaa S. Matter[3]

[1]Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor Bahru, Malaysia.
[2]College of Computer Science, King Khalid University, Abha 61413, Saudi Arabia
[3]Department of Computer Science, Applied College, King Khalid University, Abha, Saudi Arabia

Corresponding author: Amir Alrashid Abdalla Gasouma (amirkku@gmail.com).

Abstract The Internet of Things (IoT) is a rapidly expanding network of interconnected smart devices and an active area of study in the field of information technology. These smart devices may be monitored, managed, and shared information via the internet. Some of them may be tiny gadgets or sensors with restricted power and battery life. These devices short lifespans have consequences for the IoT network due to their high energy consumption. The IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) is defined in RFC6550 by the Internet Engineering Task Force (IETF), which is one of the most important protocols for routing for the IoT and the sole widely used protocol that aids in routing in Low Power and Lossy Networks (LLNs). Adapting objective functions (OFs) for routing and regulating control messages for RPL operations, which improves energy efficiency, is a major need in a method that handles IoT network difficulties and exploits new flexible network designs, such as Software-Defined RPL networks. This paper describes a software defined RPL system optimized for heterogeneous IoT environments. The proposed research made use of TriOF, an acronym for the three objective functions (OF1, OF2, and OF3) used for adaptive OF selection and routing. Use the Killer Whale Optimization (KWO) algorithm to select the best OF based on current network conditions. The energy efficiency of networks and OF selection adaptation both improved. The outcomes of simulations performed in Network Simulator 3 (NS3) are examined. The experiments' outcomes demonstrate the feasibility of our suggestion with reduced control overhead and control messages, as well as 40% and 60% less energy consumption when compared to previous works.

Keywords Energy Efficiency, Objective Function, IoT, RPL, SDN, Routing Optimization.

1 Introduction

IoT stands for "Internet of Things." It refers to the network of physical devices, vehicles, home appliances, and other items that are embedded with electronics, software, sensors, and connectivity, allowing them to connect and exchange data with each other and with the internet [1],[2]. IoT connectivity supports smart cities, industry, e-healthcare, and more [3]. Energy efficiency concerns grow with the Internet of Things [4] IoT creates large-scale heterogeneous environments with varied nodes that have energy, computational, bandwidth, and radio communication limits.

The Software-Defined Networking (SDN) paradigm has emerged as a potential option for the implementation of alternative routing control strategies, thereby by enabling global protocol strategies and network programmability, we can expand the range of Internet of Things applications that can be delivered. This expansion is made possible by the SDN paradigm. [5], [6]. SDN improves IoT energy usage issues. Thus, IoT and SDN solve network energy efficiency [3]. SDN can bring several benefits to IoT networks, some of which are:

a. **Centralized network management:** SDN provides a centralized management approach to networking that can make it easier to manage and control IoT devices. With SDN, all network devices can be managed from a central point, making it easier to provision and configure new devices, troubleshoot problems, and monitor network performance.
b. **Improved scalability:** IoT networks can grow rapidly and unpredictably, which can make it challenging to scale the network infrastructure to accommodate new devices. SDN provides a flexible and scalable architecture that can be easily expanded to accommodate new IoT devices and applications.
c. **Enhanced security:** IoT devices can be vulnerable to security threats, and traditional networking approaches may not provide adequate security protections. SDN can help to improve network security by providing a

centralized control plane that can enforce security policies across the entire network.

d. Efficient network traffic management: IoT networks generate a large amount of data traffic, which can create congestion and impact network performance. SDN can help to optimize network traffic by providing intelligent routing and traffic management capabilities that can prioritize traffic based on application requirements.

e. Better network visibility: IoT devices can be difficult to monitor and manage, particularly in large-scale deployments. SDN provides advanced monitoring and analytics capabilities that can help to improve network visibility, detect anomalies, and identify performance issues before they impact network performance. [7].

SDN controllers contain logically centralized software that controls network behavior by simplifying network design and maintenance. SDN separates the control plane and data plane from traditional networks. Figure 1 shows the Software Defined IoT (SD-IoT) concept. OpenFlow switches forward data in SDN, while controllers keep the network's global perspective [8]. SDN control plane can have one or several controllers [9]. Since IoT networks are huge, the multi-controller SDN approach works [10].

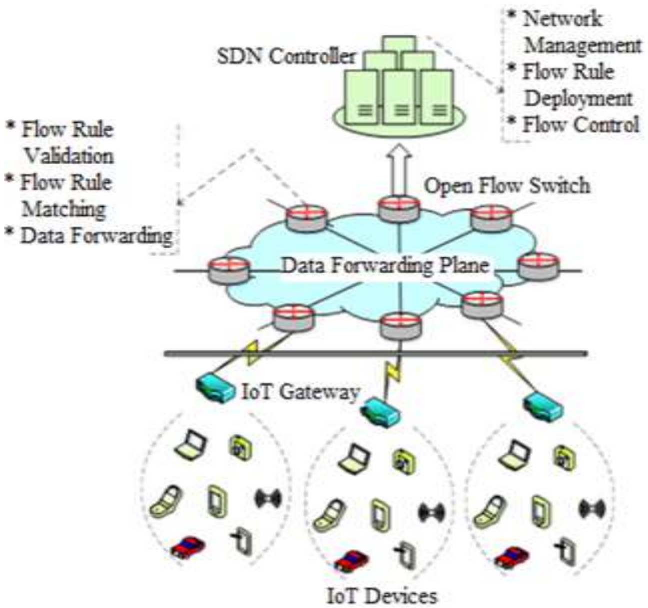

Figure 1 SD-RPL Integrated Network.

In Software Defined RPL (SD-RPL) network, IoT devices generate data, IoT gateways collect it, and SDN switches send it to the SDN controller. Data collection depends on IoT device-gateway routing. For so, The RPL employs an objective-based selection of parents designed for Internet of Things. [11], [12]. By building a Destination-Oriented Directed Acyclic Graph (DODAG), the RPL protocol allows for efficient IoT data routing. Specifically, the DODAG is the directed acyclic graph that has a single point of origin. [13],[14]. Various metrics can be used to characterize this operational function. [15]. Several different metrics are computed so that the performance of the objective function may be analyzed. For performance analysis, several important metrics, such as Expected Transmission Count (ETX), Hop Count (HC), and energy consumption of nodes, have been chosen. The performance analysis of the objective function is generated based on these metrics, and it is dependent upon the results; for example, the optimum OF is chosen with high performance in order to choose the optimum parent. [16].

Statistically, the number of devices connected to IoT networks rapidly increases yearly. IoT networks with many Heterogeneous devices capture information and deliver it to backend servers. IoT devices are resource-constrained, and they lose energy for each process. The critical problems stated for energy consumption in SDN-IoT networks are the main motivations behind this work. RPL handles parent selection by using the objective function for forwarding the captured data to the root, which is always based on the objective function. While the IoT networks are heterogeneous, the properties of each node will not be similar, so the objective functions must be selected based on the current behavior of the IoT devices and the network status.

According to this, the goal of this study is to maximize the energy efficiency of the SD-RPL network. by achieving efficient transmission of data and high packet delivery ratio by defining multiple Objective Functions, selecting optimal OF and routing based on the network status, and considering various routing metrics.

1.1 Major Contributions

To achieve the goals of this work, the following significant contributions are listed.

- An SD-RPL network has been developed and assisted using RPL routing protocol.
- The optimal objective function is adaptive based on the current network status. The Killer Whale Optimization (KWO) algorithm is executed with multiple criteria to select the optimal objective function. Here, we have three categories of objective functions (TriOF) that can be dynamically selected for parent selection as follows,

OF(1): Composed of residual energy, transmitter and receiver energy, and energy consumption for data transmission.

OF(2): Composed of mobility, load, and delivery ratio.

OF(3): Composed of ETX, link stability, and queue factor.

1.2 Paper Organization

The remaining parts of this paper is structured as follows: Section 2 summarizes the research gap by reviewing significant literature works on SD-RPL networks. This section also highlights the research problems formulated in this work. Section 3 discusses the proposed model in great depth. Sections 4 and 5 specify the simulation parameters, comparative analysis, and the comparison of the results in terms of performance metrics. Section 6 summarized the contributions and suggested future research directions.

2 Related Works

This section examines the most significant current works on optimal routing in RPL-IoT networks that have been presented.

RPL-based dynamic data gathering uses learning automata, and it is named LA-RPL [18]. In this work, two objective functions (OFs) were defined, the first was used for graph construction, and the second for learning automata. The first OF is based on the node degree, and the second OF defines the number of data packets. The first OF enabled the maintenance of the topology, and the second was for transmission. Here, both OFs use a single parameter which is insufficient for the heterogeneous IoT environment. A learning automaton for OF (LA-OF) was proposed using the expected transmission count [19]. In this learning automata, both online and offline phases are executed by computing the ETX as the states. According to the actions, rewards, and penalties for them are given. As a result of this, only the packet transmission-based parameter ETX is used. learning automaton was used to learn the network and update the parameters automatically upon ETX. This work mainly uses the Objective function Zero (OF0) which is inefficient to achieve better transmission performance.

The fuzzy logic technique was introduced to define the objective function using context-oriented parameters, i.e., COOF [20]. This OF considers the expected transmission count, queue fluctuation index, and residual energy index. Based on these three constraints, nine rules were established, and using this fuzzy parent was selected. From the fuzzy rules, the quality of the parent was determined as excellent, very good, good, fair, and bad. Then, data transmission was carried out through excellent parent nodes if available. The defined OF is not suitable for all instances, and using fuzzy could be made only when the parameters range between the specified values. If it is new, then the decision will not be efficient. Mobility-aware RPL (MARPL) was presented in an IoT environment to aid with the mobility of nodes and reduce packet loss [21]. MARPL performs three processes mobility detection, parent prediction, and trickle adjustment. The neighbors in RPL were determined based on the received signal strength value. Then the parent was selected based on the rank parameter that was updated concerning mobility. However, mobility is the significant parameter, it is only taken into consideration, even if a node with low mobility will also have lesser energy. In such situations, the preferred parent will fail in transmission.

RPL improvement was achieved for congestion control and energy efficiency [22]. For performing RPL operations, a new metric was formulated as the linear combination of ETX, delay, and node's residual energy. For parent selection, a two-level process in which the first level selects parents based on ETX and the second level uses residual energy as the tiebreaker metric. To avoid congestion, the control mechanism was presented based on time-bound (i.e.), a threshold value was set to control the broadcast. This method first uses a single metric (hop count) for selecting the optimal parent then uses energy as a tiebreaker which results in higher data loss due to the lack of current network status. In the heterogeneous IoT environment, RPL was presented to handle the heterogeneous traffic [23]. For that, queue and workload-based RPL (QWL-RPL) were presented to collect the heterogeneous traffic. The parent selection was performed based on buffer rate and workload which are in a linear combination. The absence of link and network-oriented metrics increases data loss.

Table 1 List of related works.

Techniques Applied	Research Issues
LA-RPL [18] **LA-OF [19]** **COOF [20]** **MARPL [21]** **Two-Level RPL [22]** **QWL-RPL [23]**	• Lack of significant parameters degrades data transmission performance • Using the same OF for all network conditions is not suitable • Mainly increases time and energy consumption

Table 1 summarizes the related works and lists the major research issues. According to the findings, there is still a research gap in achieving energy efficiency.

2.1 Problem Definition

A cross-layer control of data flows (CORAL) was an SDN-inspired RPL routing protocol that relies on the ETX parameter [24]. DODAG Information Objective (DIO) broadcast was handled by regularly doubling the message time. This work fails to achieve better results in parent selection. a single OF (ETX) is considered which is not optimal, because the node could be poor at the energy level and other conditions. A versatile out-of-band (VERO-SD) controls the network topology and selects the shortest path using the Dijkstra algorithm [25]. A threshold-based approach enabled congestion management. In this work, only one controller is used to determine path selection. Because most RPL-IoT devices require path selection, using only one controller takes longer. As a result, transmission delays occur, and packets may be dropped.

As follows, summarize the shortcomings of current methods and how the proposed method will address them:

In CORAL [24]**:** Problem identified is the objective function in RPL is determined only using ETX which is not optimal, because the node could be poor at energy level and other conditions. To address this problem the proposed method defined three different OFs and here each OF comprises of three parameters that mainly focus on energy. The parent selection for forwarding the data is first performed by comparing with the history of the OFs estimated. If not present, then a reinforcement learning algorithm is applied to make decision on it. Since the algorithm learns the environment, it can adapt with the best parent selection for forwarding.

In VERO-SD [25]**:** Problem identified is The controller's routing is handled by the Dijkstra algorithm, which is a blind search algorithm that cannot always find the shortest path. In the case of sensitive data, it cannot be delivered to its destination on time. To address this problem in the proposed work the packet routing is handled by reinforcement learning which can predict the best parent to route based on its learning ability.

3 System Model

We present the overall process of the proposed model in this section. For low-power computation, the suggested approach combines the Internet of Things with a software-defined network. There are several k controllers $[C_1, C_2, .., C_k]$. in this setup, as well as n IoT sensors $N_1, N_2, .., N_n$, m IoT gateways $[G_1, G_2, .., G_m]$, and l Open Flow switches $[S_1, S_2, .., S_l]$. The data is collected by the IoT sensors and sent to the root node. To determine which parent should act as forwarder, this study used RPL, which builds a directed acyclic graph (DAG) using a predefined objective function. In this proposal, the detected data is sent upward, toward the root, or the IoT gateway. Figure 2 illustrates the general structure of the proposed model. After the DODAG was built, the routing was performed by using the objective function.

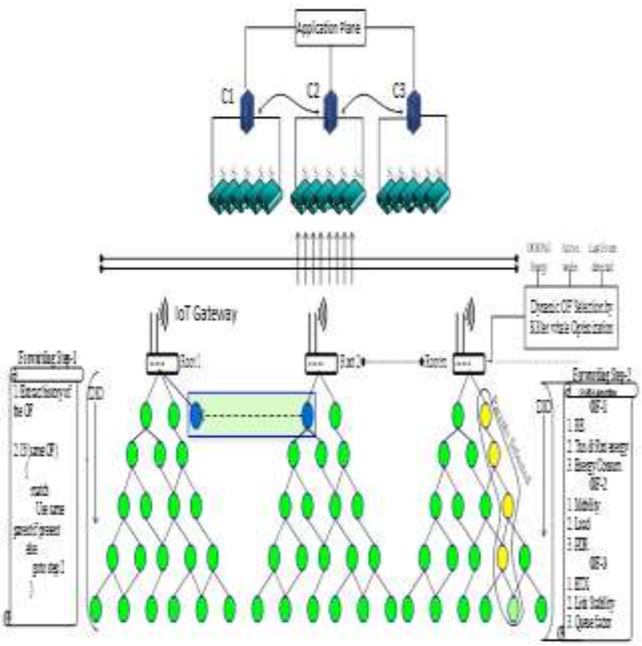

Figure 2 Proposed TriOF Model.

In this work, Tri-OF is introduced (i.e.) three categories of OFs are formulated. At each time, the source node N_{Src} selects an OF based on the current network status. In RPL-based networks, OF plays an important role, and each node needs to compute OF each time. Considering more parameters in a single OF increases computational overhead, and the OF requirements vary over the network states. Thus, an adaptive OF selection procedure is presented with the aid of an optimization algorithm. Data transmission occurs in two stages. The first step is the source node N_{Src} determines the optimal OF for current parent selection. We have proposed KWO, which is the bio-inspired algorithm for optimal OF selection in each route selection. KWO is inspired by the behavior of killer whales [26] which works better than other benchmark optimization algorithms. Mainly, KWO resolves local optimal solution problems by searching population in clusters called Matriline. At first, the solutions (OF_1, OF_2, OF_3) are initialized as population. In each matriline, a leader whale is selected to search for the optimal solution. All other whales presented in that matriline are considered members. The leader whale is responsible for finding prey (optimal OF) direction, and the members need to chase the prey by updating their locations. In this work, each matriline is considered a DODAG, and optimal OF is selected for all DODAGs by searching within the search space.

The proposed KWO algorithm involves the following steps:

1) In the first step, all members in the matriline scan the prey and the leader selects potential prey for chasing. The potentiality of the prey is evaluated in terms of fitness function $(\mathcal{F}n)$. For i^{th} prey (P_i), the Fn_i is computed as follows,

$$Fn_i = \{E(\mathbb{D}), \mathbb{AN}, \mathcal{L}\} \tag{1}$$

Fitness is evaluated as the function of DODAG energy level $(E(\mathbb{D}))$, the number of active nodes $(\mathbb{AN})$, and the last even detection time difference $(\mathcal{L})$. Each OF has a certain effect on the fitness function. If $E(\mathbb{D})$ and $\mathbb{AN}$ are low, then the difference between the latest event detection time and the current time is computed. The $\mathcal{L}$ is also low. Then the DODAG demands energy-efficient OF to minimize energy consumption. In this way, the potential prey is selected by the head.

2) Next, the member whales move the position following the selected potential prey. First, the position of the prey is modeled based on the depth of the prey $(\mathfrak{D}_P)$, depth of the leader $(\mathfrak{D}_L)$ and the angle of the horizontal range (θ). This angle is determined from the following expression,

$$\theta = \sin^{-1}\left(\frac{\mathfrak{D}_P - \mathfrak{D}_L}{R_{P,L}}\right) \tag{2}$$

Here, $R_{P,L}$ denotes the range between the prey and the leader. Once the position of the prey is determined, then all members move toward the prey's position. This movement of members is formulated as follows,

$$\begin{cases} \overrightarrow{v_L} \leftarrow \overrightarrow{v_L} + \vec{U}(0, \sigma_1) \otimes \left(\overrightarrow{\mathrm{P_{best}}} - \overrightarrow{x_L}\right) + \vec{U}(0, \sigma_2) \otimes \left(\overrightarrow{\mathrm{P_g}} - \overrightarrow{x_L}\right) \\ \overrightarrow{x_L} \leftarrow \overrightarrow{x_L} + \overrightarrow{v_L}.t \end{cases}, \tag{3}$$

The movement is modeled from the current position of a killer whale (x_L), previous best position $(\mathrm{P_{best}})$ and velocity (v_L) at a given time t. By following this model, all members move toward the prey.

3) In this step, the best solution search is continued. The matriline creates a search pattern based on previous prey. In this step, the leader computes Fn for searched prey. If this prey has more potential than the previous prey, then the leader changes the entire movement toward new potential prey. Else, the members continue to chase the old prey.

4) If the stopping criteria have been met, then the optimal solution determined in steps 2 & 3 is provided as the best

solution. Else, both steps are executed repeatedly to obtain the optimal solution.

At the end of KWO, the optimal OF that is suitable for the current network state. The three objective functions are formulated as follows,

$$OF_1 = E_{re} - \sum E_{Tx}, E_{rx}, E_c \tag{4}$$

The OF_1 is computed based on residual energy (E_{Re}), transmission energy (E_{Tx}), reception energy (E_{Rx}) and energy consumption (E_C). The transmitter and receiver energy are modeled as follows,

$$E_{Tx} = \mathfrak{B}\left(E_{elec} + \epsilon_{amp} * d^2\right) \tag{5}$$

$$E_{rx} = \mathfrak{B}(E_{da} + E_{elec}) \tag{6}$$

Here, $\mathfrak{B}$ is the number of bits per packet, d is the distance between the source and candidate parent node, E_{da} defines energy consumed for data aggregation and ϵ_{amp} is the multi-path fading signal amplification coefficient. In this work, OF_1 is mainly formulated to assure energy efficiency. If OF_1 is selected as the optimal objective function, then the node which has a higher E_{Re} and lower E_{Tx}, E_{Rx}, E_C is selected as the optimal parent node. Similarly, OF_2 is formulated as follows,

$$OF_2 = \frac{DR}{M+Ld} \tag{7}$$

The second OF is formulated when the DODAG has sufficient energy for further operations. This objective mainly focuses on maximizing the data delivery rate. Thus, a node with a high delivery ratio (DR) and low mobility (M), and load (Ld) is selected as the optimal parent node. The third OF is formulated as follows,

$$OF_3 = \frac{\psi}{ETX+\alpha} \tag{8}$$

This OF is formulated based on link stability (ψ), ETX, and queue factor (α). The N_{Src} first finds the optimal OF for current data transmission.

Pseudocode for KWO-based parent selection

Initialize $\{\mathrm{OF_1}, \mathrm{OF_2}, \mathrm{OF_3}\}$
Initialize Population
Form matriline
For each prey (P_i)
  Compute Fn_i
  Find potential prey // By Leader
  Chase potential prey // By Members
  Determine Prey Position (θ)
  Update the position of Whales x_L
  Move to new prey (P_{new})
  Compute Fn_{new}
  Compare $Fn_i \&\& Fn_{new}$
  If ($Fn_{new} > Fn_i$)
    Set P_{new} →Potential.
  Update position
  Else
    Continue with P_i
  End **If**
  While Stopping Criterion Met
    Return ($\mathrm{OF_{current}}$)
  End **While**
End **For**
If ($\mathrm{OF_{current}} = \mathrm{OF_{Previous}}$)
  Extract history
  Set Previous Parents as Optimal
Else
  Select Optimal Parent as per $\mathrm{OF_1}/\mathrm{OF_2}/\mathrm{OF_3}$
  Transmit data
End **If**
End

In the first step, the optimal OF is selected by N_{Src} is compared with the previous best OF. For that, N_{Src} extracts the history of prior OFs. Then, it compares the current OF with the previous OF to speed up the parent selection process. If the last OF and the current OF are the same, then the N_{Src} considers previous optimal parents as current optimal parents and checks whether the previous parents are available. If the previous parents are available, data transmission is performed through these optimal parents.

In the next step, the following processes are performed,

- If current and previous OFs are the same and parents are unavailable, the source chooses optimum parents for data transfer.
- The source node chooses optimum parents for data transmission if the current and previous OFs are different.

The pseudocode for proposed adaptive OF selection and parent selection explains the step-by-step procedure of proposed RPL operations. In this manner, an optimal route is selected between N_{Src} and corresponding G. After aggregating data from all sensor nodes in the DODAG, G forwards the data through OpenFlow switches deployed in the data plane.

4 Simulation Results and Analysis

In this part of the paper, we will assess the performance of the suggested model by running a number of simulations. Additionally, the performance is evaluated in perspective with previously works.

4.1 Simulation Environment

The performance of the suggested model is assessed by experimental analysis. This is accomplished by creating an Ubuntu-based NS3 model of the prepared network. When it comes to simulating networks and protocols, NS3 can do it all. So, NS3 is commonly used for simulation purposes. All algorithms are written in C++, while Python is used to construct the modules. Table 2 displays the network model simulation parameters. The network is built and simulated using these values as parameters.

Table 2 Simulation Parameters.

Parameter		Value
Simulation Area		1000*1000 m
IoT Nodes		100
Gateways		3
OpenFlow Switches		15
Controllers		3
IoT Node Initial Energy		15 Joules
Size of a Flow Table		1000 KB
Size of the Packet		512 KB
Number of Packets Generated		100
Data Rate		1.1 Mbps
Simulation Time		100 Minutes
Modules		IoT_Module Flow_Monitor_Module WiFi_Module OpenFlow_Module
KWO Configuration	Number of Matriline	10-50
	Initial Population	100
	Number of Leaders	10
	Maximum Iteration	100

The IoT nodes generate TCP and UDP traffic types in the simulation.

4.2 Comparative Evaluation

The simulation outcomes are assessed for further analysis. Important performance metrics such as the number of control message exchange, control message overhead, average energy consumption, packet delivery ratio, and packet loss rate are tracked in this regard. The RPL and SDN aspects of the work are compared to other RPL-based work such as VERO-SD [25] and CORAL [24].

Table 3 Comparison of previous works.

Existing Work	Network Model	OF	Demerits
CORAL [24]	RPL-based SDN-IoT	ETX	• Increases data loss and energy consumption. • Incapable of handling large-scale networks
VERO-SD [25]	RPL-based SDN-IoT	Dijkstra RPL routing	• High E2E delay • Non-optimal routing increases data loss

The suggested model and the previous works are compared in Table 3. According to the findings, the performance of previous works is compromised by several issues. These drawbacks are reflected in decreased efficiency levels.

4.2.1 Comparison of Control Messages

By reducing unnecessary control packet overhead, the suggested method aims to reduce consumption of energy. More control packets are needed for DODAG creation and stabilization when using RPL-based operations. As a result, the proposed study relies heavily on control message analysis. The control message analysis is performed by considering the control message overhead and the total amount of control packets exchanged.

Figure 3 compares the total amount of DODAG control messages sent to the total number of IoT nodes. Many more control messages must be sent as the number of IoT nodes increases. The strength of the broadcast message increases with the size of the network. When compared to the VERO-SD and CORAL methods, which both need 89 control messages for a network size of 100 nodes, our suggested model requires just 61. The threshold value is used to establish the broadcast limit in the VERO-SD method. Unfortunately, the threshold value is too low to accurately characterize the broadcasting limit. At certain periods, CORAL also doubles the broadcast limit. For these reasons, the current state of situations involves the exchange of a significant number of control packets. The suggested approach simultaneously uses adaptive assessment that manages the broadcast.

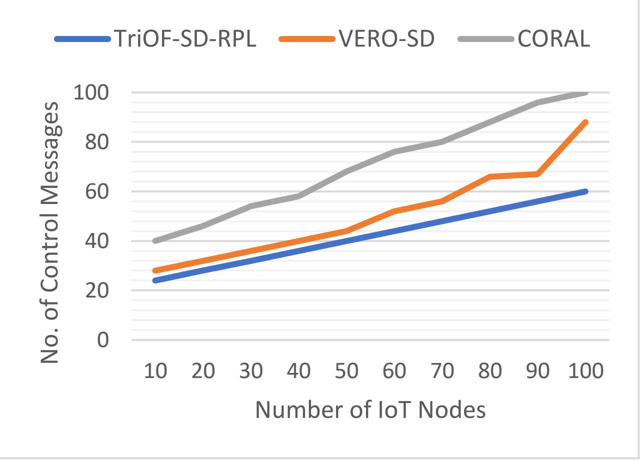

Figure 3 Control Messages.

4.2.2 Control Overhead Evaluation

Figure 4 illustrates a comparison of the simulation time required for sending and receiving control messages. When compared to the total number of packets sent, the control overhead is expressed as a percentage. Control overhead is minimized in the suggested model by a value of 0.52 at n=100. VERO-SD accomplish 0.96, but CORAL reaches overhead as 1, indicating more control packets are sent over the network for the same n value. However, exchanging many control packets is not appropriate for an IoT network because it reduces overall network reliability.

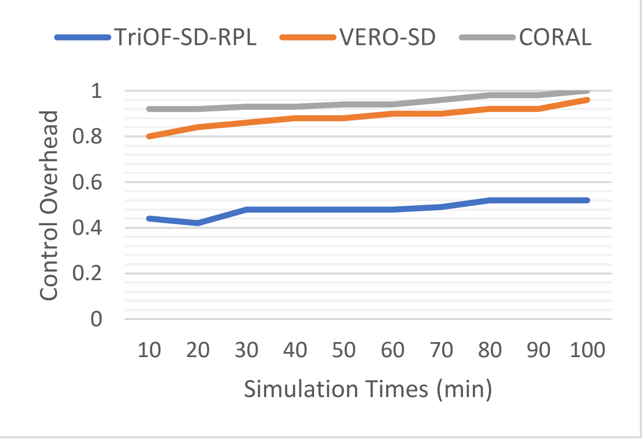

Figure 4 Control Overhead.

However, each IoT node uses more power when transferring more than necessary in the form of control packets.
In the suggested model, the DIO broadcast is used to transmit the control message to a larger number of nodes. The broadcast control in VERO-SD relies on a threshold value, however this value is computed in an inefficient manner. This causes the overhead to remain at 0.8 even when n=10. A similar method, CORAL, doubles the DIO broadcast at regular intervals and sets the duration between each interval, for n=10, the overhead is increased gradually to 0.92. Based on the findings, the suggested model requires the fewest resources and hence uses a small amount of energy.

4.2.3 Energy Consumption Evaluation

The term "energy consumption" refers to the average amount of power used by a certain network. Data transmission, data reception, and environmental sensing all have an impact on overall energy use.
In Figure 5, there is a comparison of energy usage in relation to the number of IoT nodes. Extended simulation times result in higher average network energy usage. Our model's energy usage is 6J after 10 minutes of simulation time and 18J after 100 minutes of simulation time. That amounts to 18J of energy being lost in the network.

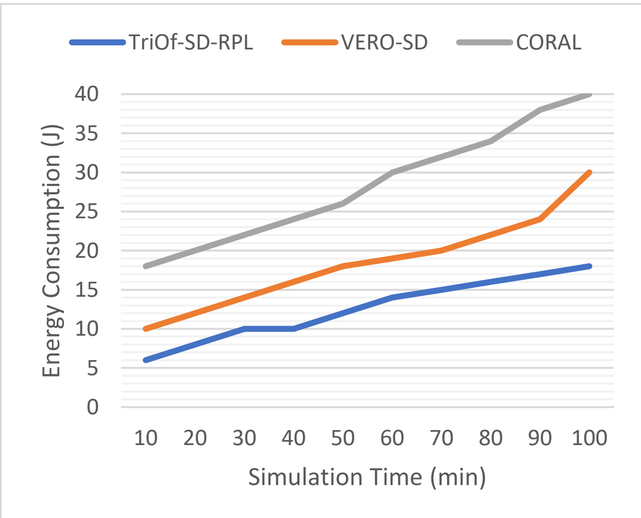

Figure 5 Energy Consumption.

At the same time, VERO-SD and CORAL both waste energy at rates that are about twice as high than the proposed work, at 30J and 40J, respectively. The tracking and transmission of data is the primary source of energy consumption in IoT devices. The creation of a DODAG and the subsequent exchange of control packets in an RPL-based network also wastes energy. Energy loss may be reduced by optimizing the overhead of control packets and the rate with which they are retransmitted. However, due to a significant control packet overhead, the current approaches cannot guarantee minimal energy loss.
By using ETX as the metric, CORAL selects an energy-minimized node as a parent several times (i.e., the energy consumption of that node increases exponentially). Moreover, the limiting function is not optimized while sending control packets. The network energy drain as a result is 40J. However, VERO-SD's use of a single OF and threshold-based broadcast control prevents it from minimizing broadcasts and achieving optimum routing. Since the suggested work reduces control message exchange and determines the best route using an adaptive OF selection technique, it uses less energy than the CORAL and VERO-SD methods.

4.2.4 Data Transmission Evaluation

The effectiveness of data transmission is measured by measuring the delivery ratio and the loss rate of individual data packets. The packet delivery ratio is the proportion of sent packets that were successfully received by the receiver.

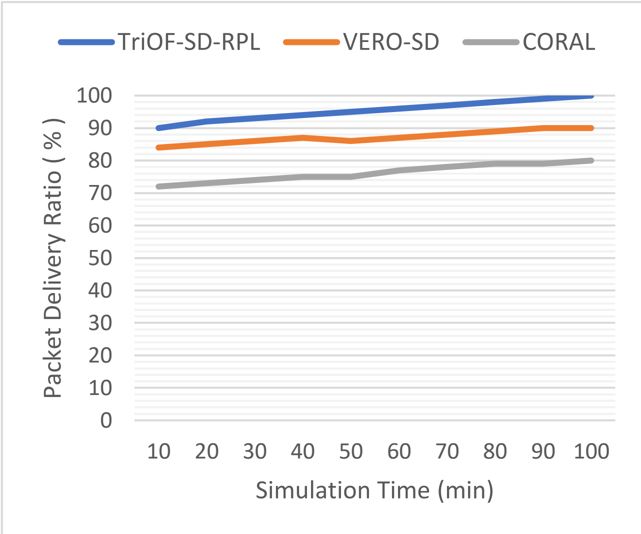

Figure 6 Packet Delivery Ratio.

The packet loss rate is another metric used to assess the reliability of a network. Figure 6 compares the packet delivery ratio while Figure 7 contrasts the packet loss rate.

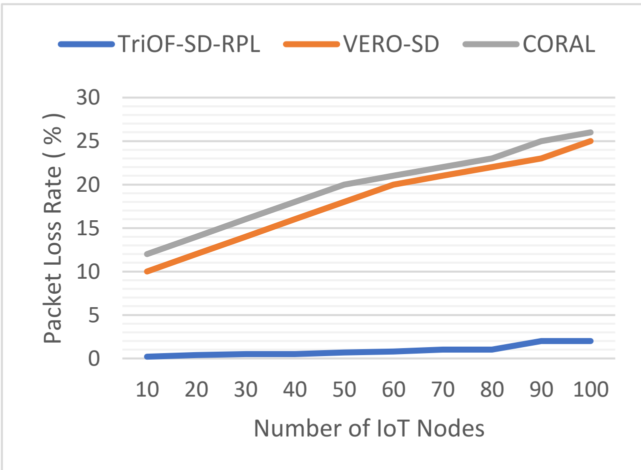

Figure 7 Packet Loss Rate.

Here, the suggested method achieves a PDR of 90%-100%, which means that a respectable percentage of packets arrive at their destination without being corrupted in transit. Additionally, the rate of packet loss is reduced to below 2%. The suggested method improves data transmission by selecting parents with the best OF in consideration of the present network state. The TriOF's data-transmission efficiency is enhanced by each of its parts. Therefore, there is no loss in transmission of any packets at all.

While VERO-SD uses a single OF for routing, the Dijkstra algorithm is used. The ETX metric is used by CORAL to determine the best possible path to take. However, because to a lack of concern for parameters, both approaches suffer from severe packet loss. SDN is also used in both works, although it is poorly managed. So, the data loss in VERO-SD and CORAL approaches is reflected in overloading at the data or control plane.

Table 4 Summary of 100 node Comparison Evaluation

Work	Control Messages	Control Overhead	Energy Consumption
CORAL [24]	101	1	40J
VERO-SD [25]	89	0.96	30J
Proposed	61	0.52	18J

Table 4 shows a summary of the comparative analysis between previous work and the suggested model, both of which have 100 nodes and whose energy consumption is assessed after 100 minutes of simulation.

5 Results & Discussion

The findings demonstrate that the suggested approach performs in all measures of energy efficiency. Each recommended improvement helps in making software-defined IoT networks more efficient with energy The killer whale optimization method, which takes into account energy, active nodes, and the time between the previous event, is used to determine the OF for each region in SD-RPL's parent selection, one of the method's primary features. Once the energy, load, and link metrics have been specified, the selection OF may be integrated into the calculation of the OF parameters.

6 Conclusion

In order to enhance the energy efficiency of IoT networks, a software-defined RPL network model is developed and simulated in this paper. TriOF's formulation makes optimal routing possible by having each OF target to types of network characteristics. The KWO algorithm is used repeatedly to choose the OF in an adaptive fashion in accordance with a number of criteria. The suggested model outperforms previous work at energy efficiency. The NS3 simulator uses parameters like control message count, control overhead, and energy consumption to assess its performance. The suggested work performs better than existing methods in every respect.

The suggested model will be enhanced in the future with a load balancing method for the data and control planes to reduce the amount of retransmission in the IoT sensor plane and hence increase efficiency. Therefore, the preferable future research path for this suggestion is to achieve energy efficiency by providing load balancing in the data and control planes.

Funding This research study is funded by the Deanship of Scientific Research at King Khalid University under research grant number **(R.G.P.2/246/44).**

Data availability All data generated in this study are available in this article and the online supplementary material.

Declarations

Conflict of interest Authors have no conflict of interest.

Ethics Statement No unethical work has been performed in this research work.

References

[1] J. H. Nord, A. Koohang, and J. Paliszkiewicz, "The Internet of Things: Review and theoretical framework," *Expert Systems with Applications,* vol. 133, pp. 97-108, 2019/11/01/ 2019, doi: https://doi.org/10.1016/j.eswa.2019.05.014.

[2] Z. Huang, X. Xu, J. Ni, H. Zhu, and C. Wang, "Multimodal Representation Learning for Recommendation in Internet of Things," *IEEE Internet of Things Journal,* vol. 6, no. 6, pp. 10675-10685, 2019, doi: 10.1109/JIOT.2019.2940709.

[3] P. Asghari, A. M. Rahmani, and H. H. S. Javadi, "Internet of Things applications: A systematic review," *Computer Networks,* vol. 148, pp. 241-261, 2019/01/15/ 2019, doi: https://doi.org/10.1016/j.comnet.2018.12.008.

[4] P. K. Khatua, V. K. Ramachandaramurthy, P. Kasinathan, J. Y. Yong, J. Pasupuleti, and A. Rajagopalan, "Application and assessment of internet of things toward the sustainability of energy systems: Challenges and issues," *Sustainable Cities and Society,* vol. 53, p. 101957, 2020/02/01/ 2020, doi: https://doi.org/10.1016/j.scs.2019.101957.

[5] E. Molina and E. Jacob, "Software-defined networking in cyber-physical systems: A survey," *Computers & Electrical Engineering,* vol. 66, pp. 407-419, 2018/02/01/ 2018, doi: https://doi.org/10.1016/j.compeleceng.2017.05.013.

[6] J. Xie *et al.*, "A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges," *IEEE Communications Surveys & Tutorials,* vol. 21, no. 1, pp. 393-430, 2019, doi: 10.1109/COMST.2018.2866942.

[7] A. H. Mohammed, K. R. M, M. k. Hussein, and I. A. Abdulateef, "A Review Software Defined Networking for Internet of Things," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 26-28 June 2020 2020, pp. 1-8, doi: 10.1109/HORA49412.2020.9152862. [Online]. Available: https://doi.org/10.1109/HORA49412.2020.9152862

[8] P. W. Tsai, C. W. Tsai, C. W. Hsu, and C. S. Yang, "Network Monitoring in Software-Defined Networking: A Review," *IEEE Systems Journal,* vol. 12, no. 4, pp. 3958-3969, 2018, doi: 10.1109/JSYST.2018.2798060.

[9] J. Lu, Z. Zhang, T. Hu, P. Yi, and J. Lan, "A Survey of Controller Placement Problem in Software-Defined Networking," *IEEE Access,* vol. 7, pp. 24290-24307, 2019, doi: 10.1109/ACCESS.2019.2893283.

[10] M. Karakus and A. Durresi, "A survey: Control plane scalability issues and approaches in Software-Defined Networking (SDN)," *Computer Networks,* vol. 112, pp. 279-293, 2017/01/15/ 2017, doi: https://doi.org/10.1016/j.comnet.2016.11.017.

[11] H. Kharrufa, H. A. A. Al-Kashoash, and A. H. Kemp, "RPL-Based Routing Protocols in IoT Applications: A Review," *IEEE Sensors Journal,* vol. 19, no. 15, pp. 5952-5967, 2019, doi: 10.1109/JSEN.2019.2910881.
[12] B. Safaei, A. A. Mohammad Salehi, A. M. Hosseini Monazzah, and A. Ejlali, "Effects of RPL objective functions on the primitive characteristics of mobile and static IoT infrastructures," *Microprocessors and Microsystems,* vol. 69, pp. 79-91, 2019/09/01/ 2019, doi: https://doi.org/10.1016/j.micpro.2019.05.010.
[13] K. S. Bhandari, I. Ra, and G. Cho, "Multi-Topology Based QoS-Differentiation in RPL for Internet of Things Applications," *IEEE Access,* vol. 8, pp. 96686-96705, 2020, doi: 10.1109/ACCESS.2020.2995794.
[14] Z. Shah, A. Levula, K. Khurshid, J. Ahmed, I. Ullah, and S. Singh, "Routing Protocols for Mobile Internet of Things (IoT): A Survey on Challenges and Solutions," *Electronics,* vol. 10, no. 19, p. 2320, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/19/2320.
[15] H. Lamaazi and N. Benamar, "A comprehensive survey on enhancements and limitations of the RPL protocol: A focus on the objective function," *Ad Hoc Networks,* vol. 96, p. 102001, 2020/01/01/ 2020, doi: https://doi.org/10.1016/j.adhoc.2019.102001.
[16] S. S. Solapure and H. H. Kenchannavar, "Design and analysis of RPL objective functions using variant routing metrics for IoT applications," *Wireless Networks,* vol. 26, no. 6, pp. 4637-4656, 2020/08/01 2020, doi: https://doi.org/10.1007/s11276-020-02348-6.
[17] H. Lamaazi and N. Benamar, "OF-EC: A novel energy consumption aware objective function for RPL based on fuzzy logic," *Journal of Network and Computer Applications,* vol. 117, pp. 42-58, 2018/09/01/ 2018, doi: https://doi.org/10.1016/j.jnca.2018.05.015.
[18] M. H. Homaei, E. Salwana, and S. Shamshirband, "An Enhanced Distributed Data Aggregation Method in the Internet of Things," *Sensors*, vol. 19, no. 14, doi: https://doi.org/10.3390/s19143173.
[19] A. Saleem, M. K. Afzal, M. Ateeq, S. W. Kim, and Y. B. Zikria, "Intelligent learning automata-based objective function in RPL for IoT," *Sustainable Cities and Society,* vol. 59, p. 102234, 2020/08/01/ 2020, doi: https://doi.org/10.1016/j.scs.2020.102234.
[20] K. S. Bhandari and G. H. Cho, "A Resource Oriented Route Selection Framework Using Contextual Information Based on Fuzzy Logic," *Electronics*, vol. 8, no. 9, doi: https://doi.org/10.3390/electronics8091023.
[21] J. Kniess and V. Marques, "MARPL: A crosslayer approach for Internet of things based on neighbor variability for mobility support in RPL," *Transactions on Emerging Telecommunications Technologies,* vol. 31, p. e3931, 12/01 2020, doi: https://doi.org/10.1002/ett.3931.
[22] B. Behrouz Vaziri and A. Toroghi Haghighat, "Brad-OF: An Enhanced Energy-Aware Method for Parent Selection and Congestion Avoidance in RPL Protocol," *Wireless Personal Communications,* vol. 114, no. 1, pp. 783-812, 2020/09/01 2020, doi: https://doi.org/10.1007/s11277-020-07393-0.
[23] A. Musaddiq, Y. Zikria, Z. Nain, and S. W. Kim, "Routing protocol for Low-Power and Lossy Networks for heterogeneous traffic network," *EURASIP Journal on Wireless Communications and Networking,* vol. 2020, 01/20 2020, doi: https://doi.org/10.1186/s13638-020-1645-4.
[24] G. Violettas, S. Petridou, and L. Mamatas, "Evolutionary Software Defined Networking-Inspired Routing Control Strategies for the Internet of Things," *IEEE Access,* vol. 7, pp. 132173-132192, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2940465.
[25] T. Theodorou and L. Mamatas, "A Versatile Out-of-Band Software-Defined Networking Solution for the Internet of Things," *IEEE Access,* vol. 8, pp. 103710-103733, 2020, doi: https://doi.org/10.1109/ACCESS.2020.2999087.
[26] T. R. Biyanto *et al.*, "Killer Whale Algorithm: An Algorithm Inspired by the Life of Killer Whale," *Procedia Computer Science,* vol. 124, pp. 151-157, 2017/01/01/ 2017, doi: https://doi.org/10.1016/j.procs.2017.12.141.