Securing RPL Networks with Enhanced Routing Efficiency with Congestion Prediction and Load Balancing Strategy

Saumya Raj[1*], Rajesh R[2]
Research Scholar, Department of Computer Science, Bharathiar University, Coimbatore, India[1]
Associate Professor, CHRIST (Deemed to be university) Bangalore, Bharathiar University, Coimbatore, India[2]

***Abstract*—Low power and Lossy Networks (LLNs) are essential components of the Internet of Things (IoT) environment. In LNNs, the Routing Protocol for LLN (RPL)-based Internet Protocol Version 6 (IPv6) routing protocol is regarded as a standardized solution. However, the existing models did not account for the issues with congestion and security when modeling the RPL. Thus, to resolve these issues, this paper proposes a novel Exponential Poisson Distribution–Fuzzy (EPD-Fuzzy) model and Kullback Leibler Divergence-based Tunicate Swarm Algorithm (KLD-TSA) for developing a reliable RPL model. The hash codes are first generated for the registered nodes at the network end in order to achieve security; the hash codes are subsequently compared via requests with the immediate nodes. Each node sends a request to its neighbors using the hash value; if the hash value matches, a path is formed. The parent nodes are then chosen and ranked using the Pearson Correlation Coefficient-Spotted Hyena Optimization Algorithm (PCC-SHOA) technique to minimize latency. To avoid congestion, the EPD-Fuzzy is employed to predict congestion; then, a genitor node is introduced in the congested scenarios. The big data and videos are split, compressed, and sent via multiple paths to reduce the losses in the RPL. Moreover, to avoid network traffic, a novel KLD-TSA load balancing is introduced at the user end. The experiential outcomes exhibited the proposed technique's effectiveness regarding Packet delivery ratio (PDR).**

Keywords*—Low power and Lossy Network (LLN); Routing Protocol for LLN (RPL); load balancing; Internet of Things (IoT); Internet Protocol Version 6 (IPv6)

I. INTRODUCTION

A platform for extending the communication paradigm to novel along with varied levels is provided by the IoT for researchers. In the IoT, computing, as well as sensor devices, are related to the internet that provides services anywhere and anytime [1]. The devices in the IoT are connected over the internet via a gateway node. In various applications like smart homes, smart farming, smart healthcare, et cetera, the IoT is wielded [2]. A network layer in the IoT architecture that utilizes diverse standards and protocols is required by the devices in those applications. Such standards and protocols are Wireless Personal Area Network (WPAN), Internet Protocol Version 4 (IPv4), IPv6 over Low Power WPAN (6LoWPAN), IPv6 and Transmission Control Protocol (TCP) [3]. However, the devices utilized in IoT are deployed as LLNs. The LLN, which features restrictions on processing speed, power, and storage capacity make up an interconnected network of resource-constrained IoT devices [4].

Owing to the quality of radios and the minuscule size of LLN, the wireless links in LLN are lossy when analogized with other wireless networks; also, poor routing is provided by the weak routing protocols in LLN owing to the limitations like higher energy consumption as well as higher data loss within the network [5]. Thus, choosing the best routing protocol, which considers lower transmission range, lower power, along lower hardware capabilities, is significant in LLN [6]. Considering these issues, the IPv6 Routing Protocol for LLNs was standardized as a consequence of the working group efforts of the Internet Engineering Task Force (IETF), which identified RPL as the leading option to handle the routing requirements of a variety of LLN-centric applications [7].

A multi-hop routing tree rooted at a single LLN Border Router (LBR), also known as the sink node or gateway node, is constructed by the RPL, a distance-vector routing protocol, by creating Destination-Oriented Directed Acyclic Graphs (DODAGs) between nodes [8]. A sorted pair of nodes is chosen in the RPL to serve as a data packet source along with a target. Data packets are transmitted via intermediate nodes from one to another [9]; lastly, the data is passed to the internet via LBR. Although the RPL has the possible to enhance and prosper, it has limitations like load imbalance and disregard for stability [10]. Moreover, the network traffic is mounted by the load imbalance on the user side while accessing the data from the internet. Thus, to resolve these problems, a novel KL-TSA model is proposed for load balancing. Also, to enhance the RPL, a modified PCC-SHOA model is proposed for parent selection with an EPD-Fuzzy congestion prediction.

A. Problem Statement

Despite developing multiple measures for efficiently transferring data in the RPL, several problems are still unnoticed and need to be resolved. Some of such problems are,

- In prevailing works, energy efficiency is mostly concentrated on the RPL network and is not concentrated on node security and congestion.

*Corresponding Author.

- As reliability is affected by rate-limiting, optimum solutions are required by load balancing as well as congestion control.
- For example, collecting a substantial quantity of data leads to mounted traffic congestion in the network. The network's unpredictable and unreliable performance is yielded by the network traffic.

By considering these problems, the proposed technique aims to develop a reliable load-balancing model at the user end and develop a reliable congestion mitigation technique with optimal parent nodes at the network end. The major contributions of the work are as follows:

- The study introduces a genitor node-centric method with the PCC-SHOA-based parent node selection, offering a novel strategy for congestion control in the RPL.
- To identify parent node congestion during data transfer, a novel EPD-Fuzzy is used.
- Kullback-Leibler Divergence-Time Series Analysis (KLD-TSA) is a novel load-balancing model that is proposed to alleviate user congestion and enhance network performance.
- Effective network load balancing at the user results in a 4675 ms latency for 250 requests from users, demonstrating effective data handling and quick access time.

B. Motivation and Benefits of the Proposed Approach

The motivation of the proposed approach comes from the issues that currently exist in the RPL protocol: Energy efficiency, load balancing, congestion control, and reliability. Although various improvements have been made, most of the current solutions consider only one-by-one problems and ignore some very critical factors that bring performance degradation, latency, and instability, mainly in IoT environments with heavy traffic. This work gives a holistic solution to dynamic user-centric load balancing through the introduction of a KLD-TSA model, a method of selecting parent nodes using PCC-SHOA for the optimization of traffic distribution, and an EPD-Fuzzy congestion prediction mechanism in an effort to reduce energy waste and enhance stability. It reduces latency and improves access times, hence making the network more reliable with better energy efficiency to meet a more scalable, flexible, and sustainable IoT network that will help different types of applications, including mission-critical services like healthcare and smart city infrastructure.

The paper's formation is systematized as: Section II implies the recent related works of RPL for IoT. Section III states the proposed approaches. Section IV elaborates on the experimental outcomes. Section V ends the paper conclusion and a better suggestion for future enhancement.

II. RELATED WORKS

This section examines current research on load balancing, congestion, and the RPL network routing mechanism.

Safara *et al*., (2020) [11] established a priority-centric energy-efficient routing (PriNergy) technique for IoT systems. The RPL model developed its own routing protocol, which determined routing method through contents with an emphasis on energy consumption. The results showed that the PriNergy mechanism decreased the overhead on the use of energy. However, the energy consumption increased when the speed of nodes increased, this influenced the PriNergy model's performance.

Conti *et al.,* (2020) [12] presented a strong multicast communication protocol for LLNs. A lower-overhead cluster-centric multicast routing mechanism was wielded on the RPL protocol's top by the presented technique. The implementation outcomes proved the protocol's efficacy over conventional protocols regarding Packet Delivery Ratio (PDR) to 25%. But the model's overall energy consumption was more than the prevailing techniques.

Mutalemwa & Shin, (2020) [13] employed secure routing protocols for safeguarding source nodes in wireless networks with multiple hops of communication. Two phantom-centric source location privacy routing protocols were developed by the presented technique. The outcomes exhibited that the protocols had better performance features with controlled energy consumption as well as PDR. However, the model's complexity reduced data transmission reliability.

Hassan *et al.,* (2020) [14] introduced a Control layer-centered trust mechanism for supporting secure routing in RPL-grounded IoT applications. The technique was named CTrust-RPL, which assessed the nodes' trust grounded on the forwarding behaviors. The presented model's outcomes proved the superiority of the model with 35% more energy efficiency. Yet, CTrust-RPL could be confronted with energy preservation, scalability, along decentralization issues.

Preeth *et al.,* (2020) [15] deployed a proficient parent selection approach in the RPL by utilizing Ant Colony Optimization (ACO) along with coverage-centric dynamic trickle systems. For parent selection, an energy-efficient RPL protocol with ACO-grounded multi-factor optimization was generated by the study. The outcomes exposed that the E-RPL had 90% of PDR over 30 node topologies. Although it was a better model, the E-RPL could not achieve better routing overhead when the DODAG was increased.

Seyfollahi & Ghaffari, (2020) [16] explored a Lightweight Load balancing and Route Minimizing solution for RPL (L^2RMR). The L^2RMR scheme encompassed an Objective Function (OF) together with a routing metric grounded on the path route minimization. The outcomes exhibited that the developed model could enrich the energy consumption, End-to-End delay, along average Packet Loss Ratio (PLR). However, the L^2RMRscheme could not perform reliably during high traffic betwixt the nodes.

Manikannan & Nagarajan, (2020) [17] propounded a framework for the RPL/6LoWPAN-centric IoT network with the firefly approach. An RPL-based firefly optimization algorithm was developed to establish a stable and dependable protocol mobility management framework. The experiment proved that the mPRL-firefly optimizer enhanced the PDR by

an average of 2.31% more than the other prevailing algorithms. Nevertheless, the average power consumption in the developed system increased when contrasted with the conventional RPL model.

Chiti *et al.,* (2021) [18] implemented a green routing protocol with power transfer for IoT. An OF for RPL grounded on a composite metric, which considered the parent node's remaining power together with the child node, could handover to the parent node as per the Wireless Power Transfer (WPT) concept. The performance evaluation exhibited remarkable energy saving, which prolonged the network lifetime. Yet, for a long-range, the model could not perform routing efficiently.

Bidai, (2022) [19] enriched the RPL for supporting video traffic for Internet of Multimedia Things (IoMT) applications. A multi-Path version of RPL (MP-RPL), which leveraged the multi-parent feature provided by RPL for constructing various end-to-end paths of diverse qualities regarding radio link quality, was wielded by the enhanced model. The simulations exhibited that feasible and acceptable Quality of Service (QoS) was provided by the presented model when contrasted with the conventional single-path RPL. Since the conventional RPL performs better, the model is limited to the average end-to-end delay.

Karami and Derakhshanfard, (2020) [20] illustrated a Routing Protocol grounded on Remaining Time to encounter nodes with Destination nodes (RPRTD) utilizing an Artificial Neural Network (ANN). The routing was carried out by identifying the contact node with more effective conditions. The results showed that the RPRTD model efficiently and with higher accuracy anticipated the time needed for interacting nodes with the destination node while requiring less storage. However, with the ANN model, the RPRTD framework took more time for training in the LLNs.

Royaee *et al.,* (2021) [21] demonstrated a context-aware system for RPL load balancing of LLNs in the IoT. Therefore, load balancing and Automata-ant colony-centric Multiple Recursive RPL (AMRRPL) were developed to prevent congestion. The Cooja simulator experiments showed that the AMRRPL algorithm significantly improved with increased PDR and network lifetime. Nevertheless, the node ranking took more time to converge, which could cause a delay.

Sahraoui & Henni, (2021) [22] developed a Secure and Adaptive Multi-Path RPL (SAMP-RPL) for enriched security along with reliability in the heterogeneous IoT. For IPv6 RPL, the SAMP-RPL relied on three variants of adaptive together with safe multipath routing. The outcomes of the Cooja simulator exhibited the SAMP-RPL model's efficacy for enhanced dependability and security of communication at lower costs. The simulation on the Cooja platform triggered the inaccuracy issues.

Yassien *et al.,* (2021) [23] developed the RPL and Load Balancing Time-Based (LBTB) model to optimize the load balancing procedure with the capability for attaining superior network reliability as well as service time. The LBTB was employed with the modification of the trickle Timer algorithm. The outcomes displayed that higher performance was achieved regarding time-saving and power-saving. However, the imbalance among the nodes caused a congestion problem.

Musaddiq *et al.,* (2020) [24] employed an RPL for the heterogeneous traffic network. Here, various RPLs under heterogeneous traffic were evaluated; also, a protocol named Queue and Work-Load-based RPL (QWL-RPL) was introduced. The outcomes displayed that QWL-RPL could enhance the heterogeneous traffic network's performance concerning the amount of overhead, jitter, along average delay. However, for scheduling, the control messages as well as service discovery had issues associated with overhead and convergence time.

Hadaya & Alabady, (2021) [25] designed an enhanced RPL protocol for the IoT environment. An enhancement in the RPL OF is suggested by the presented work that considered 3 metrics, namely Expected Transmission count (ETX), residual energy, and load. The outcomes exposed that the RPL protocol was enhanced by the model concerning total power consumption, PLR, along PDR. However, the nodes' data security was not efficiently maintained since it was learned with only a limited number of Cognitive Packet Network's features.

A. Research Gap

Literature research gaps in the improvement of RPL protocols for IoT networks are based on some key challenges. While energy efficiency, scalability, and security have improved, there are related trade-offs that need to be addressed. On the other hand, energy efficiency improvements mostly come at a cost in terms of performance under different conditions, including high traffic or node mobility. On their part, scalability issues are manifested with increased network size, contributing to increased routing overhead. The reliability is poor, and security enhances the complexity under high traffic for the architecture. Current context-aware and adaptive protocols are overload with long convergence times, having computational overhead. Multimedia support suffers due to end-to-end delays. Decentralized approaches that extend the lifetime of the network remain underdeveloped. There is also a need for better support of multimedia services and QoS to reduce latency, particularly in real-time IoT applications. Other challenges include decentralize and efficiently manage over long distances. A critical research gap is thus the development of scalable, reliable, energy-efficient RPL-based solutions that ensure efficient handling of high traffic, enhanced security, support for real-time multimedia, and uniform performance across different IoT environments.

III. PROPOSED ROUTING APPROACHES IN THE RPL

The RPL concept was promoted by the rapid development of the IoT. However, in prevailing models, secure packet delivery and traffic congestion control were hardly performed; also, much importance was not given to the user-side traffic. Thus, to overcome these issues, a novel EPD-Fuzzy-centric congestion control in the RPL is proposed with the KL-TSA load balancing technique. Fig. 1 depicts the architecture of the proposed RPL.

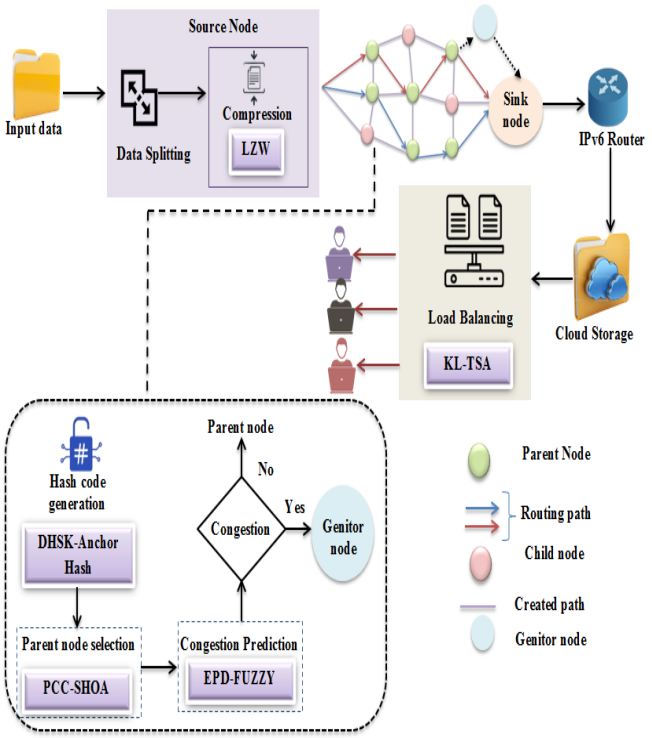


Fig. 1. Framework of the proposed RPL.

A. Node Registration

The proposed model begins with the registration of nodes participating in the network. All the nodes are registered in the network with their ID(ID), IP(IP), and MAC(M) Address. The registered details are mathematically represented in Eq. (1),

$$reg \leftarrow \langle ID, IP, M \rangle \tag{1}$$

Here, regspecifies node registration.

B. Hash Code Generation

During the registration, the hash code is generated for nodes utilizing the hybrid Diffie Hellman Secret Key-based Anchor Hashing (DHSK-Anchor Hash). In the DHSK-Anchor Hash, the keys are generated with the Diffie Hellman Algorithm (DHA), and then the generated keys are hashed with the Anchor hash. The DHSK-Anchor Hash procedure is explicated further.

1) Key generation: During the node registration, the keys are generated for every single node utilizing the DHA.

- Public and private key generation

The sender and the receiver side agree on a prime $(e)$and generator(r) in DHA. After that, the private keys$k$and $z$are chosen by the sender and the receiver side. With these values, the public keys generated at both sides are expressed in Eq. (2) and Eq. (3),

$$p = r^k \bmod e \tag{2}$$

$$a = y^z \bmod e \tag{3}$$

Where, p, aportray the public key generated at the sender side and receiver side, correspondingly. Afterward, $p, a$are shared betwixt the sender and receiver.

- Shared secret key calculation

After the public key is exchanged, the symmetric secret key(s) is generated at both sides, which is expressed in Eq. (4),

$$s = a^k \bmod e = p^z \bmod e \tag{4}$$

Here, $a^k \bmod e$ is assessed at the sender's side, $p^z \bmod e$and is assessed at the receiver's side.

2) Anchor hash: After the keys are generated, the hash value is computed utilizing the anchor hashing technique which $ID_{sender}, ID_{receiver}, p, aands$ are considered the key values(ϖ). Hence, the anchor hashing is given as follows,

a) Anchor representation: In the anchor hashing, a set of integer arrays(ϑ) is utilized for representing the anchors, which is mathematically represented as in Eq. (5),

$$\vartheta = [0,1,\ldots..,d] \tag{5}$$

where the size of the array is portrayed as$d$. After that, a bucket(B) of size $d-1$ is selected from the integer setϑ. Here, the bucket encloses$ID_{sender}, ID_{receiver}, p, a$and s. The current working buckets ϑ is symbolized as W, where $W \subseteq \vartheta$. Therefore, the bucket within the integer set is signified as$\vartheta[B]$, which is expressed as in Eq. (6),

$$\vartheta[B] = \begin{cases} 0 if B \in W \\ |W_B| if B \in \aleph \end{cases} \tag{6}$$

Where, $\aleph$represents the stack of the removed bucket and $W_B$indicates the size of the working set.

b) Hashing: In anchor hashing, a hashing function $H$is wielded to map the key values of the buckets, which is mathematically denoted in Eq. (7),

$$u_B(\varpi) \equiv \nabla(B, \varpi) \bmod \vartheta[B] \tag{7}$$

Here, $u_B(\varpi)$denotes the hashed output. During the path creation, each node sends a request to neighboring nodes with the $u_B(\varpi)$. A path will be created between such nodes if the neighboring nodes give the same hash value.

C. Optimal Parent Node Selection Using PCC-SHOA

During the path creation, the parent nodes are selected through which the data packets are forwarded. Here, utilizing the PCC-SHOA, the parent nodes get selected. In the conventional SHOA, position updation has more variation betwixt the prey and the hyena. Therefore, the Pearson Correlation Coefficient (PCC) technique is included in the SHOA model. Spotted hyena optimizer (SHO) is a recently created popular metaheuristic algorithm that draws its main inspiration from social ties between hyenas. The females in the family of spotted hyenas are the dominant ones. The spotted hyenas follow their prey using their inherent senses of sight, hearing, and scent. Spotted hyenas make a sound to interact with one another while searching for a new food source. They rely on a pack of about 100 hyenas who are their

closest companions for hunting. Thus, the working steps of PCC-SHOA are given further.

1) Initialization: The PCC-SHOA's input parameters are initialized in which the Spotted Hyena (SH) population(H) is the node involved in the path creation that can be mathematically formulated as in Eq. (8),

$$H = \{h_1, h_2, \ldots, h_l\} or h_x, x = 1,2, \ldots, l \quad (8)$$

Where, $h_l$specifies the position ofl^{th}SH, $l$signifies the population size. Also, the SH has four behaviors, namely Encircling, hunting, attacking, and searching for prey.

2) PCC-based encircling prey: In the PCC-SHOA algorithm, the best SH has obtained whose position is near the prey. The ability to locate their prey and encircle them is possessed by spotted hyenas. Since the search space is unknown in advance, the best contender at this time is assumed to be the spotted hyena that is closest to the target or prey. Once the optimal search solution has been determined, the locations of the other search agents are updated. By calculating the fitness function, the best position is attained. In the proposed model, the lower Residual energy, Transmission count, Distance, and bandwidth are considered as the fitness function. Afterward, during encircling, the distance between h_x and the prey position $(\alpha_{pos})$is calculated utilizing the PCC as in Eq. (9) and Eq. (10),

$$\lambda_{dist} = \left| \vec{C}. \frac{\Sigma(\vec{\alpha}^I_{pos} - \alpha'_{pos})(\vec{h}^I_x - h'_x)}{\sqrt{\Sigma(\vec{\alpha}^I_{pos} - \alpha'_{pos})^2 \Sigma(\vec{h}^I_x - h'_x)^2}} \right| \quad (9)$$

$$\vec{h}^{I+1}_x = \vec{\alpha}^I_{pos} - \vec{Q}.\lambda_{dist} \quad (10)$$

where, λ_{dist}specifies the distance betwixt SH and the prey, $\vec{C}, \vec{Q}$symbolizes the vector coefficients,$\vec{h}, h'$indicates the current and the mean position of SH,h_x^{I+1}implies the position of SH $x$in the iteration $I+1$, and iteration is signified as $I.\alpha, \alpha'$represent the current and the mean position of prey. The $\vec{C}, \vec{Q}$values are mathematically expressed as Eq. (11) and Eq. (12),

$$\vec{C} = 2 * \vec{R}_1 \quad (11)$$

$$\vec{Q} = 2 * \vec{\omega}.\vec{R}_2 - \vec{\omega} \quad (12)$$

Here, $\vec{R}_1, \vec{R}_2$symbolizes the random vectors and $\vec{\omega}$ portrays the reduction vector, which is computed as in Eq. (13),

$$\vec{\omega} = 5 - \left(I \times \frac{5}{I_{max}}()\right) \quad (13)$$

where the maximum iteration is notated as I_{max}. To ensure that exploration and exploitation are properly balanced, $\vec{\omega}$ falls linearly from 5 to 0 for the maximum iterations. With an increase in the number of iterations (MaxIteration), this method allows for further development. By modifying the values of $\vec{C}$and $\vec{Q}$, spotted hyenas can update their position in relation to the location of their prey.

3) Hunting prey: Spotted hyenas can detect prey, hunt in packs, and depend on a network of reliable companions. Assume that the prey is known to the best search agents, whichever is optimal, in order to define spotted hyena behaviour mathematically. Other search agents should update their location in accordance with the best solution and move in the direction of the best search agent. Here, the mathematical model is constructed by considering the best SH that knows the optimal position, whereas the other SHs update their corresponding position towards the best positions. This mathematical model is specified in Eq. (14),

$$\vec{\lambda}_{dist} = \left| \vec{C}.\vec{h}^*_x - \vec{h}_x \right| \quad (14)$$

$$\vec{h}_x = \vec{h}^*_x - \vec{Q}.\vec{\lambda}_{dist} \quad (15)$$

Where, $\vec{h}^*_x$specifies the first best spotted SH position, $\vec{h}_x$denotes the other SH positions near $\vec{h}^*_x$ which is defined in Eq. (15). Therefore, the cluster$(\vec{\Re})$with the number of the optimal solution is represented in Eq. (16),

$$\vec{\Re} = \vec{h}_x + \vec{h}_{x+1} + \ldots\ldots + \vec{h}_{x+\ell} \quad (16)$$

Here, ℓ indicates the number of SH in the best position and is defined in Eq. (17),

$$\ell = count_{nos}(\vec{h}^*_x, \vec{h}^*_{x+1}, \vec{h}^*_{x+2}, \ldots, (\vec{h}^*_x + \vec{M})) \quad (17)$$

where nos indicates the number of solutions and counts all candidate solutions after addition with $\vec{M}$, which are significantly close to the best optimal solution in the search space and $\vec{M}$ is a random vector with a value of [0.5, 1].

4) Attacking: For performing the attacking behavior, $\vec{\omega}$ is reduced. Moreover, the variation in the $\vec{Q}$is reduced to change the value of$\vec{\omega}$. The SH attacks the prey when $|Q| < 1$and the prey attacking is equated as in Eq. (18),

$$\vec{h}^{I+1}_x = \frac{\vec{\Re}}{\ell} \quad (18)$$

Update$\vec{h}^{I+1}_x$ if the fitness of the current position $\left(\vec{h}^{I+1}_x\right)$ is better than the previous position, and by continuously updating $\vec{h}^{I+1}_x$, the optimal solution (parent nodes) is attained.

5) Prey search: The SHs search for their prey in the cluster vector$(\vec{\Re})$. Moreover, the SHs diverge from each other to attack and search the prey. The prey search is grounded on the changes in the $\vec{Q}$, which is utilized to randomly search the prey. If $(|Q| > 1)$, the SHs leave the prey and move to the next prey or else perform the attack on the selected prey. By this mechanism, global searches can be attained. Hence, the final parent nodes selected $\vec{h}^{I+1}_x$ $or n_\delta$ are signified as in Eq. (19),

$$P = \{n_1, n_2, \ldots, n_q\} or n_\delta \quad (19)$$

Where, P illustrates the parent node set and$n_q$represents the q^{th}selected parent node. The pseudocode of PCC-SHOA is given in Algorithm 1.

Algorithm 1: Pseudocode of PCC-SHOA

Input: Nodes$\{h_1, h_2, \ldots, h_l\} or h_x$,

Output: Selected parent node

Begin

Initialize *SH population,l,* $\vec{R}_1, \vec{R}_2$, *and maximum iteration* I_{max}

Set $\boldsymbol{I = 1}$

While $(\boldsymbol{I \leq I_{max}}())$***do***

Calculate *fitness*

Determine $\boldsymbol{\lambda_{dist}}$*using PCC*

Define $\vec{\Re} = \vec{h}_x + \vec{h}_{x+1} + \ldots\ldots + \vec{h}_{x+\ell}$

If *reducing factor* $(\vec{Q} > 1)_{\{}$

Update *position using* $\vec{h}_x^{I+1} = \vec{\alpha}_{pos}^{I} - \vec{Q}.\lambda_{dist}$

} ***Else*** *{*

Update *position using* $\vec{h}_x^{I+1} = \frac{\vec{\Re}}{\ell}$

}

End If

If the *fitness of* $\vec{h}_x^{I+1}$ *greater than* $\vec{h}_x^{I}$ ***Then***

Update $\vec{h}_x^{I+1}$

Else

$$I = I + 1$$

End If

End While

Return *optimal value*

End

After that, the rank is assigned to each selected parent node as per the fitness values of the parent nodes.

D. Data Splitting and Compression

After all the nodes are connected, the source node senses the data to be transferred to the destination node. If the sensed data size is huge, the files are split into small files, then compressed and sent to the destination node via multiple paths. This process is done to reduce the data loss in the LLN. The split parts are compressed with the Lempel–Ziv–Welch (LZW) lossless compression, then the data is transferred via nodes. The big file (D) is split into a small file as in Eq. (20),

$$D = \{v_1, v_2, \ldots, v_k\} or v_o \tag{20}$$

Hence, the k^{th}small file is illustrated asv_k. By utilizing the LZW algorithm, this small file $v_o$is compressed. The file$v_o$is compressed utilizing a table-centric lookup model in the LZW algorithm by performing encoding of the information in the file. The table formed is named dictionary or code table. The number of entries commonly accepted in the table is 4096; also, a single byte from the input file v_o is coded with the codes 0-255. While encoding is initiated, only the first 256 entries are present in the dictionary. The compression is attained by utilizing the 256 codes through 4095 entries for representing the sequence of bytes.

During compression, LZW identifies repeated sequences in v_o, then added to the dictionary. Suppose the string in the file $v_o$is represented as $babc$, which is compressed with LZW is given as in Eq. (21),

$$ababbabc = 12452 \tag{21}$$

This compressed value is sensed in the source node and transmitted to the server. Fig. 2 elucidates the flow diagram of the proposed system,

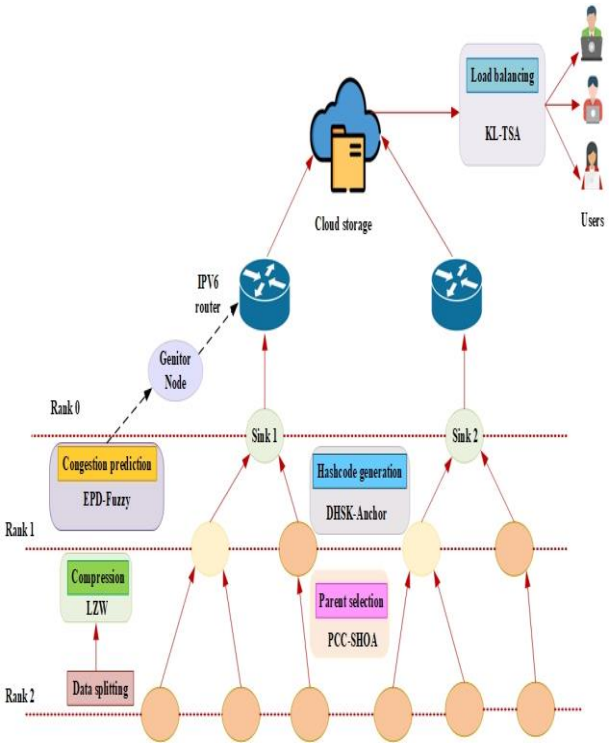


Fig. 2. Flow diagram of the proposed RPL.

E. Congestion Prediction Using Novel EPD-Fuzzy Model

During the data transfer, a novel EPD-Fuzzy detects the congestion among the parent nodes. Fuzzification, rule evaluation, and defuzzification are the three processes performed by the fuzzy algorithm. However, the fuzzy inference process has a lower level of rule generation processing than the prevailing Fuzzy algorithm. Thus, to resolve this issue, the Exponential Poisson Distribution technique is included in the prevailing Fuzzy algorithm. Hence, the congestion prediction with the EPD-Fuzzy is stated as follows,

1) Fuzzification: Primarily, the incoming number of packets (g), the number of outgoing packets (t) and the hop count (w)data $\{n_\delta\}$are given as the crisp set to the fuzzy control system, which gets mapped by a membership function for generating fuzzy sets. Hence, the membership function is represented in Eq. (22), Eq. (23) and Eq. (24),

$$m(g) = \frac{exp(-\varsigma \times \tau)(\varsigma \times \tau)^{g}}{g!} \quad (22)$$

$$m(w) = \frac{exp(-\varsigma \times \tau)(\varsigma \times \tau)^{w}}{w!} \quad (23)$$

$$m(t) = \frac{exp(-\varsigma \times \tau)(\varsigma \times \tau)^{t}}{t!} \quad (24)$$

Where, $m(\ )$signifies the EPD membership function andς, τelucidates the center and width of the fuzzy set.

2) Rule generation: After the membership function is defined, the $m(n_{\delta})$ is correlated to generating the fuzzy rules as in Eq. (25), (26) and (27),

$$\rho(g) = \begin{cases} exp\left(\frac{\psi[m(g)-m(g')]}{m(g^*)-m(g^*)}\right) if m(g) = \{g \in (g^*, \infty) \\ 1 else \end{cases} \quad (25)$$

$$\rho(w) = \begin{cases} exp\left(\frac{\psi[m(w)-m(w')]}{m(w')-m(w^*)}\right) if m(w) = \{w \in (w^*, \infty) \\ 1 else \end{cases} \quad (26)$$

$$\rho(t) = \begin{cases} exp\left(\frac{\psi[m(t)-m(t')]}{m(t')-m(t^*)}\right) if m(t) = \{t \in (t^*, \infty) \\ 1 else \end{cases} \quad (27)$$

where $\rho(\ )$implies the fuzzy rules generated in the inference, $*,'$ are the membership function's lower bound and upper bound. After that, the fuzzy rules are aggregated by utilizing IF-THEN statements. The aggregation method is given by max, which is also namedORoperator, which is expressed as in Eq. (28),

$$\Delta = max\big(\rho(g), \rho(w), \rho(t)\big) \quad (28)$$

Where, Δspecifies the aggregated outputs with the result of the implication technique.

3) Defuzzification: A process that converts fuzzy values to crisp values is named defuzzification. Hence, by computing the centroid technique, the crisp value is attained. Here, the center of the area of the fuzzy set is attained, which determines the crisp output (congestion rate)f.

F. Genitor Node

Here, a novel genitor node is included, which acts as the parent node for sending data if the $(f > Th)$is predicted by EPD-Fuzzy; where, Thindicates the threshold value. The sensed data is securely transferred to the cloud server via these processes.

G. KLD-TSA-based Novel Load Balancing

Conversely, users who want to access data from the cloud server give requests to access the data. But, multiple requests at the same time mount the network traffic. To avoid such congestion in the network, Load balancing is performed in the proposed model. Here, for load balancing, KLD-TSA is wielded. In the prevailing Tunicate Swarm Algorithm (TSA), the conflicts among the search agents are more, which affects the algorithm's performance. Hence, to avoid conflicts, Kullback Leibler Divergence (KLD) is introduced in the prevailing TSA. Tunicate is capable of locating food sources in the ocean. On the other hand, the food source in the specified search space is unknown. To locate the optimal food supply, tunicates use two different behaviors. Swarm intelligence and jet propulsion are these tendencies. Thus, the proposed load balancing is given further.

The users who request to access the resources from the server are considered as the initial population of tunicates and the position of the tunicate population is expressed as in Eq. (29),

$$J = \{j_1, j_2, \ldots, j_{\wp}\} or j_y \quad (29)$$

where the tunicate population is denoted as$J$, j_{ρ}represents the position of the tunicate$\wp$, and$\wp$indicates the population size. To attain the optimal solution, the tunicates perform jet-propulsion and swarm behavior. The mathematical model of jet propulsion satisfies three behaviors: Prevent conflicts, move toward the best search agent, and remain close to the best tunicate. Utilizing the fitness value, the best search agent is computed. Here, fitness is considered as less response and waiting time.

a) Prevent conflicts among agents: In the proposed KLD-TSA, the initialization of the new position of the search agent$(\vec{N})$ to avoid inter-agent conflict is given by utilizing the KLD in Eq. (30).

$$\vec{N} = \sum_{y=1}^{\wp} \Omega(\vec{V}_y)\, ln \frac{\Omega(\vec{V}_y)}{\Theta(\vec{S})} \quad (30)$$

Where,$\vec{V}_y$is the gravity force of tunicate y, and$\vec{S}$are the social forces betwixt tunicates. The gravity force is expressed in Eq. (31),

$$\vec{V} = r_2 + r_3 - \vec{G} \quad (31)$$

$$\vec{G} = 2.r_1 \quad (32)$$

Here,$r_1, r_2$and $r_3$epitomize the random values that lie in the range of 0 to 1. The water flow advection in the deep sea is symbolized by $\vec{G}$ and is defined in Eq. (32). $\vec{S}$ stands for the social dynamics among search agents. The vector $\vec{S}$ is computed as in Eq. (33),

$$\vec{S} = \left[A1min_{max_{min}}\right] \quad (33)$$

Here, the initial and subordinate speeds of social interaction are represented by A_{min} and A_{max}.

b) Move towards the best neighbor: After avoiding the conflicts betwixt the agents, the search agents move toward the direction of the best agent as in Eq. (34),

$$\vec{T} = \overrightarrow{\vec{F}_l - L(\vec{j}_y^{It})} \quad (34)$$

where$\vec{T}$indicates the distance between the tunicates and the food, $\vec{F}_l$symbolizes the food location, $L$is a random value between [0, 1], and $\vec{P}_i$signifies the tunicate positions.

c) Keeping close to the best agent: The search agent is able to stay in the direction of the optimal search agent (food source). Now, the tunicate move towards the prey is computed as in Eq. (35),

$$\vec{J}_y^{It} = \begin{cases} \left(\left|\vec{F}_l\right|^2 + |(N)(T)\sin\theta|^2\right)^{1/2} for L \geq 0.5 \\ \left(\left|\vec{F}_l\right|^2 - |(N)(T)\sin\theta|^2\right)^{1/2} for L < 0.5 \end{cases} \quad (35)$$

Where, θ signifies the angle between N and T. The updated position of tunicates in relation to the location of food sources is represented by $\vec{J}_y^{It}$.

d) Position update: The tunicate's swarm behavior is updated by updating the position of all search agents concerning the first two best search agents is revealed as follows in Eq. (36),

$$\left|\vec{J}_y^{It+1}\right| = \left(\frac{\left|\vec{J}_y^{It}\right|^2 + \left|\vec{J}_y^{It+1}\right|^2}{4+r^2}\right)^{1/2} \quad (36)$$

Where, $\left|\vec{J}_y^{It+1}\right|$ is the magnitude of the updated position of the tunicates. After that, the fitness $\left|\vec{J}_y^{It+1}\right|$ is evaluated. If the fitness $\left|\vec{J}_y^{It+1}\right|$ is greater than the $\left|\vec{J}_y^{It}\right|$, then the position is updated. Therefore, the optimal solution (i.e., optimal user) is obtained by updating the position. Thus, by selecting the optimal user, traffic is avoided. Hence, the network load is balanced. The pseudocode of the proposed KLD-TSA is given in Algorithm 2.

Algorithm 2: Pseudocode of KLD-TSA

Input: Users

Output: optimal user

Begin

Initialize *tunicate population, parameters* $\vec{N}, \vec{V}, \vec{S}$, *and maximum iterations* It_{max}

Calculate *fitness*

Set *Iteration* $It = 1$

While $(I_t \leq \omega)$***do***

Update *New position using KLD*

Move *toward the best search agent*

If $(L \geq 0.5)${

Update *tunicate position using* $\left(\left|\vec{F}_l\right|^2 + |(N)(T)\sin\theta|^2\right)^{1/2}$

} Else If $(L < 0.5)${

Update *tunicate position using* $\left(\left|\vec{F}_l\right|^2 - |(N)(T)\sin\theta|^2\right)^{1/2}$

}

End If

Update *the position of all tunicate*

End while

Set $I_t = I_t + 1$

Return $\left|\vec{J}_y^{It+1}\right|$

End

IV. RESULTS

Here, the proposed RPL methodologies' performance is experimentally evaluated with conventional techniques to demonstrate the reliability of the proposed protocol model. The performances are experimentally verified on the working platform of JAVA and the cloud sim simulation tool.

A. Performance Analysis

Here, the proposed protocol's performance is assessed in three phases, namely hash code generation, parent node selection, and load balancing. Here, regarding hash code generation time, the performance of the proposed hash code generation model DHSK-Anchor hash is comparatively analyzed with Anchor hash, SWIFFT, SHA512, and MD5 techniques.

The time taken to generate the hash value is named hash code generation time. The hash code generation time attained by the proposed protocol is 2163ms, which is 3051ms, 5178ms, and 5972ms lower than the prevailing SWIFFT, SHA-512, and MD5 techniques. This shows that the proposed RPL outperforms the conventional models. Fig. 3 elucidates the pictorial representation of hash code generation time.

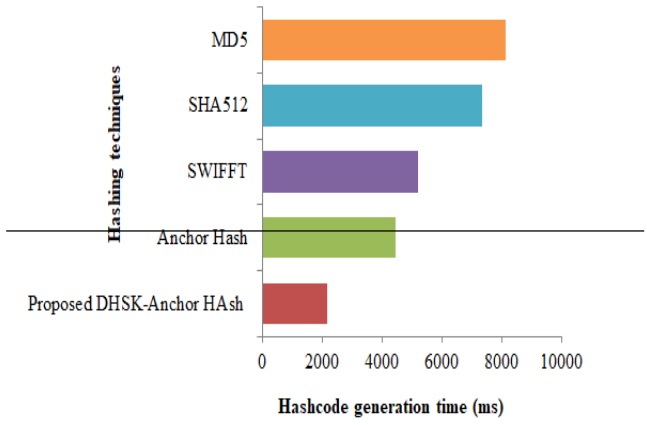


Fig. 3. Time analysis for hash code generation.

Fig. 4 depicts the graphical analysis of iteration vs. fitness for the proposed and existing algorithms.

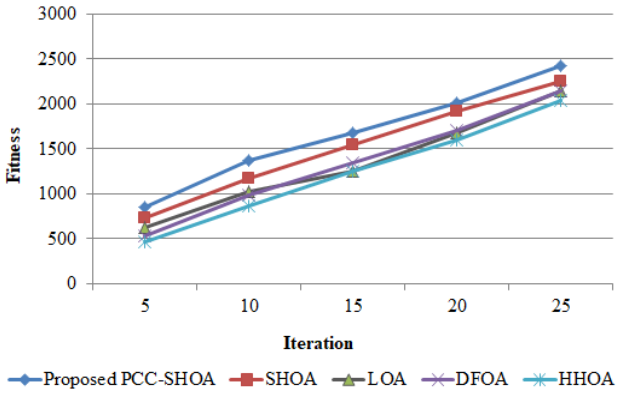


Fig. 4. Fitness vs. iteration analysis.

The Fitness value evaluation of the proposed PCC-SHOA approach and the conventional SHOA, LOA, DFOA, and HHOA selection algorithms is elucidated in Fig. 4. Here, at the 25th iteration, the proposed PCC-SHOA obtained an optimal parent node whose fitness is 2423, which is higher when contrasted with the fitness value achieved by the prevailing SHOA (2257), LOA (2148), DFOA (2144), and HHOA (2046). This proves that the proposed PCC-SHOA converged much faster than the prevailing techniques. Fig. 5 depicts the graphical analysis of throughput.

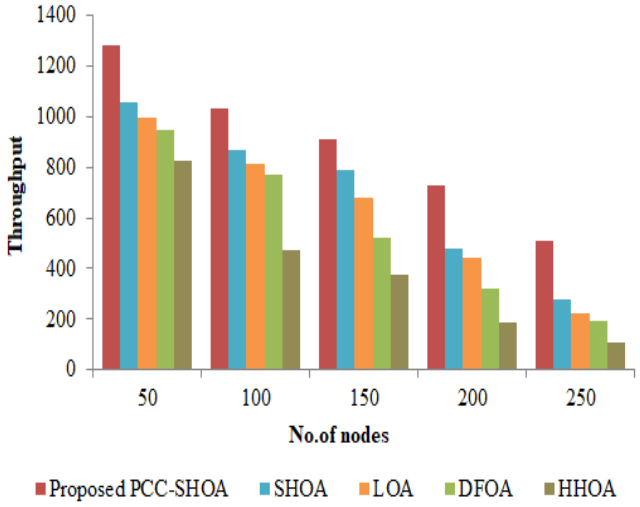
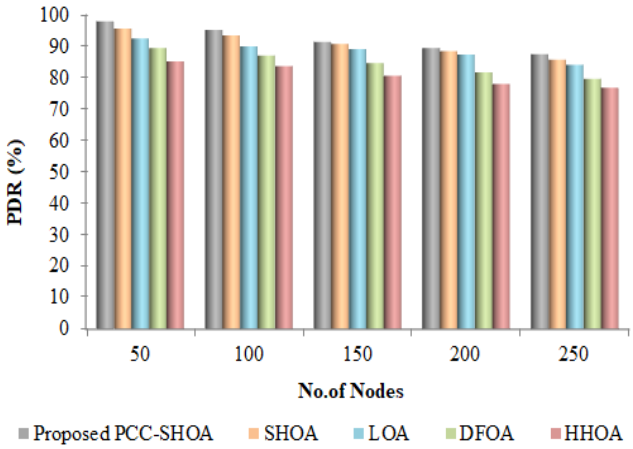
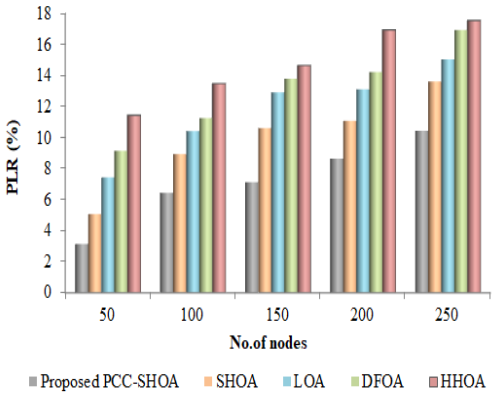


Fig. 5. Throughput analysis.

To determine how efficiently the algorithms achieved better data transmission with the selected parent nodes, the throughput is evaluated. Fig. 5 demonstrates that the throughput is analyzed for 50 to 100 nodes. Here, the proposed algorithm achieved the highest throughput of 1281 for 50 nodes, which is higher than the prevailing approaches that attained 1054 for SHOA, 993 for LOA, and 828 HHOA approaches. This concludes that with the proposed PCC-SHOA, the parent with lower Residual energy, Transmission count, Distance, and bandwidth is selected, which could enhance the proposed RPL. Fig. 6 (a) and (b) illustrate the analysis of PDR and PLR.



(a)



(b)

Fig. 6. (a) PDR and (b) PLR analysis.

The PDR and PLR are the metrics evaluated to determine the rate of packets delivered successfully and the rate of packets dropped during the data transmission. Fig. 6(a) displays that the rate of packets successfully delivered by the proposed PCC-SHOA for 100 nodes is 1.78%, 9.37%, and 13.70% higher than the prevailing SHOA, DFOA, and HHOA approaches. Then, from Fig. 6(b), it is revealed that the PLR of the proposed algorithm is 3.15%, 6.45%, 7.12%, and 10.45% for 50,100, 150, and 250 nodes, which are lower than the existing algorithms. This proves that with the use of PCC-SHOA-centric parent selection, more data is efficiently transferred with less loss, which is owing to the splitting and compression of large files.

Here, the latency, waiting time, and TAT performance of the proposed KLD-TSA approach are analyzed in comparison with the prevailing TSA, Cockroach Swarm Optimization Algorithm (CSOA), LOA, and DFOA approaches. Fig. 7 represents the latency attained by the proposed and the prevailing models.

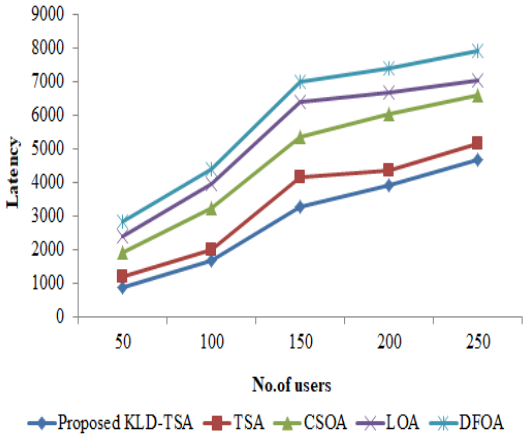


Fig. 7. Latency of the proposed framework.

Here, latency is the delay that occurs between when the user requests access and the response. The Fig. 7 displays that the Proposed KLD-TSA approach has lower latency than all other algorithms followed by TSA, CSOA, et cetera. However, the latency attained by the proposed KLD-TSA is 3289ms for 150 users, whereas 4176ms, 5347ms, and 6981ms

latency were attained by the existing TSA, CSOA, and DFOA schemes. Thus, the overall time efficiency of the proposed model is proved by this analysis.

B. Discussion

The performance of the proposed PCC-SHOA parent node selection algorithm and KLD-TSA load balancing algorithm is compared to that of current techniques in the discussion section.

1) Performance analysis of parent node selection: In this phase, the proposed PCC-SHOA algorithm's performance is comparatively analyzed with the prevailing SHOA, Lion Optimization Algorithm (LOA), Dragon Fly Optimization Algorithm (DFOA), and Harris Hawks Optimization Algorithm (HHOA) regarding the parent selection time, fitness value, throughput, response time, Turn-Around Time (TAT), PDR, and PLR. The time taken by various algorithms to select the parent node is illustrated in Table I.

TABLE I. TIME TAKEN TO SELECT PARENT NODES

Algorithms	Parent node selection time (ms)
Proposed PCC-SHOA	6034
SHOA	6513
LOA	8274
DFOA	9627
HHOA	10344

Among the prevailing algorithms, SHOA takes less time to choose the optimal parent node, which is 6513ms. Yet, with the implementation of the PCC technique in the SHOA, 479 ms lesser time is taken for choosing the optimal parent node, which displays the time effectiveness of the proposed PCC-SHOA approach.

The response and the turnaround time of the proposed and existing approaches for 50 to 250 nodes are illustrated in Table II.

TABLE II. RESPONSE TIME AND TAT

Metrics	Algorithms	Number of nodes				
		50	100	150	200	250
Response time (ms)	Proposed PCC-SHOA	3781	4796	5447	6145	6794
	SHOA	5142	6834	7402	9247	10375
	LOA	6753	7664	8314	9924	10852
	DFOA	7348	8016	9307	10267	11576
	HHOA	8457	9374	10493	11752	12055
TAT (ms)	Proposed PCC-SHOA	5423	6942	7581	9427	10524
	SHOA	7156	8123	9076	10072	11543
	LOA	8056	9365	10786	11898	12630
	DFOA	9546	10498	11966	12756	13277
	HHOA	10374	11863	13757	14624	15371

The time taken to send the data to the immediate node is named the response time, whereas the TAT is the time taken by the RPL to transmit data to the server. Here, the response time and TAT increase with the number of nodes. Here, for 250 nodes, the response time of the proposed PCC-SHOA is 6794ms, which is lower than the prevailing algorithms. Also, the best TAT is achieved by the proposed algorithm, which is 5423ms for 50 nodes.

2) Performance analysis of load balancing: The waiting and turnaround time determined for 250 users with the proposed KLD-TSA in comparison with the prevailing methodologies is illustrated in Table III.

TABLE III. WAITING TIME AND TAT OUTCOMES OF THE KLD-TSA APPROACH

Metrics	Algorithms	Number of users				
		50	100	150	200	250
waiting time (ms)	Proposed KLD-TSA	2781	4653	6447	7256	7649
	TSA	4133	6922	7464	8046	10953
	CSOA	5613	7914	8706	10264	11952
	LOA	7394	8672	9767	10527	11543
	DFOA	8857	9325	10335	11442	12594
TAT (ms)	Proposed PCC-SHOA	4423	6602	7921	9597	10554
	TSA	6969	8513	9276	10172	11643
	CSOA	7658	9265	10662	11658	12560
	LOA	9661	10598	11454	12656	13761
	DFOA	10456	11935	13746	14404	15879

The time taken by the users to access data after requesting is called waiting time, whereas TAT is the overall time taken to get data concerning the number of users. Here, the proposed model's waiting time for 50 users is 2781ms, which is lower than the prevailing CSOA (4133ms), LOA (7394ms), and DFOA (8857ms) approach. Moreover, the proposed model's TAT for 50 users is the least (4423ms). This exhibits that with the KLD-TSA data balancing, the data can be accessed from the server in the least time.

C. Comparative Analysis with the Related Works

Here, the PDR for 50 to 100 nodes is analyzed for the proposed routing protocol and the prevailing works of (Conti et al., 2020) [12], (Preeth et al., 2020) [15], and (Hadaya & Alabady, 2021) [25]. Table IV illustrates the comparative analysis of PDR with the proposed and existing algorithms.

TABLE IV. COMPARATIVE ANALYSIS WITH THE RELATED RESEARCH

Metric	Algorithms	Number of nodes		
		50	55	60
PDR (%)	Proposed EPD-Fuzzy	97.96	96.42	95.37
	(Hadaya & Alabady, 2021)	97.145	95.185	94.575
	(Preeth et al., 2020)	84.96	83.24	82.59
	(Conti et al., 2020)	82	81.16	79.60

The PDR metric is evaluated for determining the ratio of packets delivered successfully to the cloud server. The PDR is evaluated for 50, 55, and 60 nodes in Table IV, which displays that PDR is inversely proportional to the number of nodes that participated in the network. Here, the PDR is assessed for 55 nodes. With 55 nodes participating in the network, the PDR attained by the proposed routing protocol is 96.42%, which is 1.29%, 15.83%, and 18.80% higher than the prevailing works of [25], [12] and [15]. This proves that with the approaches introduced in the proposed routing protocol, more data packets are transmitted.

V. Conclusion

This paper proposes a genitor node-centric congestion control in the RPL with the PCC-SHOA-based parent node selection. The KLD-TSA-centric load-balancing model is proposed to avoid congestion among users. The proposed technique's experiments are performed on the Cloudsim simulator; also, the performance was assessed. The performance evaluation showed that the path betwixt the nodes is created in less time since the hash codes are generated in less time. Moreover, the optimal parent node is selected with the fitness of 2423 in lesser time; also, with the selected parent node, the PDR of the proposed model gets enhanced by 95.23% more than the prevailing algorithms. At the user end, the network load is balanced with a latency of 4675ms for 250 user requests. After that, the proposed RPL model's overall efficiency is proved by attaining higher PDR than the conventional systems. These outcomes proved that the proposed protocol was superior to other routing protocols. Several data are still lost even after utilizing the LZW compression in the proposed model. The research indicates that to further minimize data loss, future work may incorporate sophisticated deep-learning models for congestion prediction and make use of modified compression techniques.

Declarations

Conflict of interest: The authors declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Consent of publication: Not applicable.

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

Competing interests: The authors declare that they have no competing interests.

Funding: This work has no funding resource.

Author's contributions: All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Saumya Raj, Dr. Rajesh R. The first draft of the manuscript was written by Saumya Raj and all authors commented on previous versions of the manuscript.

All authors read and approved the final manuscript.

Acknowledgment

We thank the anonymous referees for their useful suggestions.

References

[1] S. Sankar, S. Ramasubbareddy, A. K. Luhach, A. Nayyar, and B. Qureshi, "CT-RPL: Cluster tree based routing protocol to maximize the lifetime of internet of things," Sensors, vol. 20, no. 20, p. 5858, 2020.

[2] S. Sennan, R. Somula, A. K. Luhach, G. G. Deverajan, W. Alnumay, N. Jhanjhi, U. Ghosh, and P. Sharma, "Energy efficient optimal parent selection based routing protocol for internet of things using firefly optimization algorithm," Transactions on Emerging Telecommunica- tions Technologies, vol. 32, no. 8, p. e4171, 2021.

[3] Z. A. Almusaylim, A. Alhumam, and N. Jhanjhi, "Proposing a secure RPL based internet of things routing protocol: A review," Ad Hoc Networks, vol. 101, p. 102096, 2020.

[4] H. Farag and C˘. Stefanovic˘, "Congestion-aware routing in dynamic iot networks: A reinforcement learning ap- proach," in 2021 IEEE Global Communications Confer- ence (GLOBECOM). IEEE, 2021, pp. 1–6.

[5] H. Shreenidhi and N. S. Ramaiah, "Improving lifetime of IoT network by improvising routing protocol on low power and lossy network by using Contiki Cooja tool," in 2020 International Conference on Computational Intelli- gence for Smart Power System and Sustainable Energy (CISPSSE). IEEE, 2020, pp. 1–4.

[6] A. Touzene, A. Al Kalbani, K. Day, and N. Al Zidi, "Performance analysis of a new energy-aware RPL rout- ing objective function for internet of things," in 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE). IEEE, 2020, pp. 1–6.

[7] M. Mahyoub, A. S. H. Mahmoud, M. Abu-Amara, and T. R. Sheltami, "An efficient RPL-based mechanism for node-to-node communications in IoT," IEEE internet of things journal, vol. 8, no. 9, pp. 7152–7169, 2020.

[8] Y. Kim and J. Paek, "NG-RPL for efficient P2P routing in low-power multihop wireless networks," IEEE Access, vol. 8, pp. 182 591–182 599, 2020.

[9] N. Azman, A. Syarif, J.-F. Dollinger, S. Ouchani, L. Idoumghar et al., "Performance analysis of RPL proto- cols in LLN network using Friedman's test," in 2020 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS). IEEE, 2020, pp. 1–6.

[10] *Pancaroglu and S. Sen, "Load balancing for RPL- based internet of things: A review," Ad Hoc Networks, vol. 116, p. 102491, 2021.*

[11] *Safara, A. Souri, T. Baker, I. Al Ridhawi, and M. Alo- qaily, "PriNergy: A priority-based energy-efficient rout- ing method for IoT systems," The Journal of Supercom- puting, vol. 76, no. 11, pp. 8609–8626, 2020.*

[12] M. Conti, P. Kaliyar, and C. Lal, "A robust multicast com- munication protocol for low power and lossy networks," Journal of Network and Computer Applications, vol. 164, p. 102675, 2020.

[13] L. C. Mutalemwa and S. Shin, "Secure routing protocols for source node privacy protection in multi-hop commu- nication wireless networks," Energies, vol. 13, no. 2, p. 292, 2020

[14] T. ul Hassan, M. Asim, T. Baker, J. Hassan, and N. Tariq, "CTrust-RPL: A control layer-based trust mechanism for supporting secure routing in routing protocol for low power and lossy networks-based internet of things appli- cations," Transactions on Emerging Telecommunications Technologies, vol. 32, no. 3, p. e4224, 2021.

[15] S. S. L. Preeth, R. Dhanalakshmi, R. Kumar, and S. Si, "Efficient parent selection for RPL using ACO and coverage based dynamic trickle techniques," Journal of Ambient Intelligence and Humanized Computing, vol. 11, pp. 4377–4391, 2020.

[16] A. Seyfollahi and A. Ghaffari, "A lightweight load bal- ancing and route minimizing solution for routing protocol for low-power and lossy networks," Computer networks, vol. 179, p. 107368, 2020.

[17] K. Manikannan and V. Nagarajan, "Optimized mobility management for RPL/6LoWPAN based IoT network architecture using the firefly

algorithm," Microprocessors and Microsystems, vol. 77, p. 103193, 2020.

[18] Chiti, R. Fantacci, and L. Pierucci, "A green rout- ing protocol with wireless power transfer for internet of things," Journal of Sensor and Actuator Networks, vol. 10, no. 1, p. 6, 2021.

[19] Z. Bidai, "RPL enhancement to support video traffic for IoMT applications," Wireless Personal Communications, vol. 122, no. 3, pp. 2367–2394, 2022.

[20] A. Karami and N. Derakhshanfard, "RPRTD: Routing protocol based on remaining time to encounter nodes with destination node in delay tolerant network using artificial neural network," Peer-to-Peer Networking and Applications, vol. 13, pp. 1406–1422, 2020.

[21] Z. Royaee, H. Mirvaziri, and A. Khatibi Bardsiri, "De- signing a context-aware model for rpl load balancing of low power and lossy networks in the internet of things," Journal of Ambient Intelligence and Humanized Computing, vol. 12, pp. 2449–2468, 2021.

[22] S. Sahraoui and N. Henni, "SAMP-RPL: secure and adaptive multipath rpl for enhanced security and reli- ability in heterogeneous iot-connected low power and lossy networks," Journal of Ambient Intelligence and Humanized Computing, vol. 14, no. 1, pp. 409–429, 2023.

[23] M. B. Yassien, S. A. Aljawarneh, M. Eyadat, and E. Eay- dat, "Routing protocol for low power and lossy network– load balancing time-based," International Journal of Ma- chine Learning and Cybernetics, vol. 12, no. 11, pp. 3101–3114, 2021.

[24] A. Musaddiq, Y. B. Zikria, Zulqarnain, 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, pp. 1–23, 2020.

[25] N. N. Hadaya and S. A. Alabady, "Improved rpl protocol for low-power and lossy network for iot environment," SN Computer Science, vol. 2, no. 5, p. 341, 2021.