International Journal of Computational Mathematics and Computer Science

Volume 1, Issue 2, July (2024)

ISSN: XXXX-XXXX

Hybrid Optimized Routing in RPL Protocol for Low Power and Lossy Networks

Dr. Mohamed Sithik M*

Assistant Professor

Department of Computer Science, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai-600062

Email: drmohamedsithikm@veltech.edu.in

Junath Naseer Ahamed,

Senior Faculty

Department of Information Technology, College of Computing and Information Sciences, University of Technology and Applied Science, IBRI. Sultanate of Oman.

Email: Junath.Naseer@utas.edu.om,

Abstract – In this research work, a new load balancing mobility aware model was developed for Routing protocol for Low Power Lossy networks (RPL). In the first stage we keep a simple OF that considers expected transmission count (ETX) and cumulative path delay to rank candidate parents as per RPL specification. This model has the capacity to identify mobile nodes and pass this information to possible child nodes so that the mobile node will not be preferred while identifying the next parent. Finally, a hybrid model was developed called Load balancing Mobility aware Hybrid-RPL (LMH-RPL) which can provide load balancing energy conscious route to static nodes (SN) and proactive handoff mechanism for mobile nodes in the network. The executed results prove that the proposed work performs better when compared with other prominent RPL algorithms.

Keywords: RPL, Objective Function, static nodes, Mobile Node, Load Balancing.

1. INTRODUCTION

Routing Protocol for Low power Lossy Network is a distance vector based proactive routing protocol that works on IEEE 802.15.4 standard network. This protocol was developed by IETF-ROLL work group and drafted in RFC 6550 (Winter et al., 2012). It is optimized to work in low power wireless environment susceptible to packet loss and retransmission. RPL has the capability to conserve energy while maintaining route proactively and quickly share routing knowledge and adapt to topology changes. RPL constructs topology like a tree and the root node of the tree acts as the controller for the tree. Each node is ranked in the tree which advertises its relative ability to become an

*Corresponding Author

ideal parent to its sub nodes. The most common open-source operating system implementing RPL is Contiki. RPL is also implemented in LiteOS, TinyOS, T-Kernel, RIOT etc. RPL forms nontransitive / non broadcast multiple access (NBMA) network topology upon which it computes routes (Winter et al., 2012). An RPL network can have one or more sink nodes both of which are connected to a common backbone link.

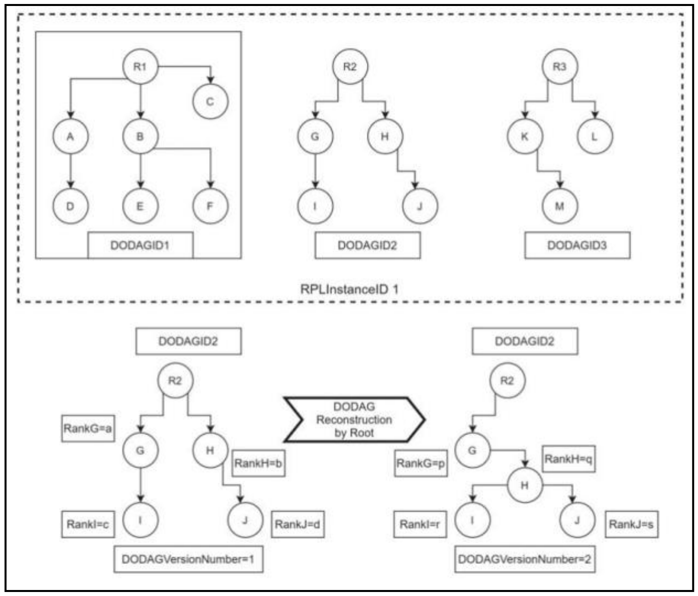

Figure 1 Different identifiers used by RPL

1.1 RPL Limitations and Challenges

As the de facto standard routing protocol for IoT networks, RPL has been successful in addressing many key issues in LLN routing. But still, it requires improvements in terms of energy conservation and mobility support. Below are the few areas in which RPL needs improvement to provide better routing services in various IoT network

1.2 Limitations with Objective Functions

RPL relies on single path routing. Once a node identifies a preferred parent, then the node continues to utilize it for all data transfers until it becomes inaccessible. This can lead to a problem where all children use the most preferred parent in the neighborhood and eventually exhausting it. This can reduce the lifetime of the network and can contribute to poor QoS. RPL does not provide any load balancing feature to counter this problem. Researchers have developed various models on RPL which imparts Load balancing features. Many of them delivers promising result. The approach differs on the fact that some models use the secondary path identified actively for load balancing while others use it only as a backup if primary route fails or its resource fall short of a specific threshold. Identifying and keeping a backup route becomes even more challenging when we have MN changing the topology of the network.

1.3 Metrics Composition

RPL uses multiple metrics to evaluate the fitness of route. It is left to the implementation on how to utilize these metrics and rank the nodes in the network. This provides greater flexibility in adapting RPL to different domain networks. Relying on single routing metric can provide excellent results in one network but could be a complete failure in another (Altwassi et al., 2018). For example, hop count metric will create DODAG with less depth but will exhaust resources of high child count parent. These results in a situation where the performance of RPL completely depends on the implementation specification. An ideally performing RPL implementation cannot be moved to another network without understanding the OF used in it and its fitness into the new domain. This situation arises because RPL standard does not give any specification on how the metrics should be combined or used to rank nodes (Karkazis et al., 2012). The impact of MinHopRankIncrease on RPL will make longer path in network less preferable to a node even though it may be a better quality path to BR. This effect sometimes supersedes the effect desired by the OF to prefer path based on its metric settings. To find an ideal RPL setting in terms of OF for a domain stands as a research challenge.

1.4 Mobility support

RPL is developed to provide good performance by keeping itself resource constraint as it must work on LLNs. The implementation of trickle algorithm helps a lot to achieve faster route convergence with low routing overhead. Trickle archives this by reducing routing updates once the network stabilizes. But the MNs in network always have high routing requirement throughout its lifetime. Trickle's forced routing update reduction in the network will render MN parentless in the network. Increasing routing update altogether in network is unnecessary for SNs and will incur more cost in terms of network resources. Mobility support should treat MN separately and cater to its need in LLNs. This makes imparting mobility support for RPL an open research challenge (Ghaleb et al., 2018).

1.5 Downward Routes

As per standard, RPL expects MP2P data transfer to be the dominant traffic pattern in LLNs. RPL packets, node states and routing overhead are all optimized to achieve MP2P traffic pattern. But this has an adverse effect on P2P and P2MP traffic pattern (Ko et al., 2015).

2. RELATED WORK

In (Farag et al., 2019), a reliable mobility aware routing (RMA-RPL) enhancement for RPL is proposed. In this paper, dynamic motion detection and link quality are combined to update network nodes on topological changes. To save energy, an adaptive timer is used to limit the number of routing changes in the network. The RMA-RPL is divided into two stages. A dynamic rank updating system is built in the first phase, which may follow MN's movement and relay changes to its prospective parents. This step also guarantees that the network does not generate routing loops. The RSSI value is used by RMA-RPL to determine node mobility in respect to its neighbors. A dynamically managed timer is employed in the second phase to minimize the DIO transmission rate in the network while also detecting topology changes with minimal latency. RMA-RPL produced greater PDR, lower network overhead, and lower overall latency in the network, according to simulation results. To increase RPL performance with MN, the authors introduced a smart-HOP method (mRPL) in (Fotouhi et al., 2015). The reactive method based on the trickling algorithm, according to the authors, is insufficient to manage MN's route requests in the network. This paper proposes smart-HOP, a proactive handoff method that reduces packet loss and keeps handoff time under a tenth of a second. The results were evaluated in the Cooja simulator, and it was discovered that smart-HOP achieves nearly 100% packet delivery at a cost of less than 1% more network routing overhead. The access point notifies children of their relative distance from it by responding with the average RSSI it received in the past n broadcasts. A node that transmits data continuously is informed of its mobility in this way. However, the stated outcome may be influenced by differences in network density and data transfer rate.

Murali and Jamalipour (2018) suggested a mobility-aware, energy-efficient parent selection approach. ETX, RSSI, ELT, and the Euclidean distance between PP and MN are the metrics considered in this model. For each node in the candidate parent set, rank and distance are determined based on the above metrics. The PP is then identified by selecting the lowest rank and distance value, with rank value taking precedence over distance value. As a result, PP is chosen based on a low ETX, a high ELT, and a high RSSI value, all of which gives superior MN to QoS in the network. The authors also suggest D-Trickle, a dynamic trickle algorithm that adjusts the listen only duration according on the density of nearby MNs. The investigation was carried out using the Cooja simulator with regular RPL and several different RPL versions. It was shown that this model works better in terms of PDR, the number of nodes alive, and lowers end-to-end latency and energy consumption in the network under various traffic scenarios. To adapt RPL to automotive IoT networks, authors created the Enhanced Mobility-based Content Centric Technique (EMCC) in (Kumar and Hariharan, 2020). For high-speed mobile nodes in the VANET, EMCC tries to give rapid routing changes. In this architecture, MN notifies its neighbours of its mobility capabilities by including a flag in the DAO message. The content-centric OF reduces latency and network energy consumption. According to how serious it is, data in the network is classified in this paradigm as A or B. This strategy is known as content centric since each node defines various channels for transferring different types of data. In Cooja simulation, EMCC outperforms them RPL algorithms in terms of packet reception ratio and transmission latency.

Manikannan and Nagarajan (2020) presented a firefly algorithm-based RPL optimization for IoT networks with MN. This research is based on mRPL that has been optimized using the firefly algorithm. RouteRQ packets are used to correct the initial brightness assigned to each source node in this work. The attractiveness and brightness parameters utilized in the algorithm are sent through the RouteRQ packet header. There are five steps to the model. The default values for the performance parameters are initially applied. The firefly method is then used to evaluate the OF. Each node then communicates its energy value to its neighbors. Now, depending on the energy received, neighbour nodes assess a node's attractiveness value. The best parent node is then selected based on the data gathered. According to the study's findings using the Contiki OS, the mRPL + firefly method outperformed other algorithms in terms of PDR, network latency, and power consumption. The authors of (Shin and Seol, 2020) proposed a solution for decreasing power consumption in RPL networks utilizing MN. The proposed model is flexible enough to accommodate different mobility patterns seen in various IoT network domains. The amount of movement experienced by a node's NN is indicated by its mobility level (ML). The ML of a node determines the rate at which it sends routing control messages.

This allows MN to get timely route updates while also reducing routing packet transmission in highly sta ble network areas. In this work, ML controls the trickling timer's interval, reducing the amount of superfluous DIO message transmission and resulting in energy savings. The Cooja simulator assessment asserts a 70% boost in PDR over RPL-VANET. The comparison is only valid for one protocol, thus it may be worthwhile to look at how well the model performs in networks with higher densities.

For mobile Internet of Things devices, Theodorou and Mamatas (2020) suggested a Software Defined Networking (SDN) solution (SD-MIoT). Using an intelligence algorithm and the connection graph provided by the SDN controller, SD-MIoT can detect network movement passively. The recommended mobility detecting module for use in this study is MODE. Based on the mobility behaviour identified by MODE, this model may alter routing tactics to reduce control overhead while delivering robust connections. A node transition matrix is created to better understand node mobility in the network. This matrix interacts with MODE to provide MN with the optimum route across the network. With the help of Contiki OS and a human movement pattern, the evaluation was conducted in a real-world context. The results show that the SD-MIoT performed better in terms of PDR and reduced network routing overhead. In order to improve RPL performance, Gaddour et al. (2014) introduced a fuzzy logic optimization employing OF. The proposed OF-FL uses four criteria to determine the optimum path through the network: end-to-end latency, hop count, connection quality, and remaining node energy. Residual energy, the last statistic, is used as a restriction to prevent designating a network node with lesser energy as a parent node. This strategy reduces network energy consumption while enhancing PDR.

Altwassi et al. (2018) introduced a load balancing traffic aware composite metric called as ETXPC-RPL. This metric is the ratio of total parent count to ETX. The load balancing approach of ETXPC-RPL ensures that nodes with lesser number of children and lower ETX value is preferred over other nodes to become a parent. The result shows that the proposed model improved PDR while reducing power consumption.

EL-RPL (Energy and Load Aware RPL) was proposed by authors in (Sankar and Srinivasan, 2018a). EL-RPL uses BDI (Battery Depletion Index), ETX and current load for ranking parents. Current load is calculated by counting the number of children for a parent and BDI is the ratio between initial energy and current energy level. The metrics are combined with appropriate weights to generate the OF called OF-EL. The node which exhibits the minimum value for OF-EL is selected as the preferred parent. The EL-RPL shows a 10% improvement on network lifetime and 4% improvement in PDR when compared to standard RPL.

Alvi et al. (2017) proposed a new OF using the combination of Hop Count and ETX. Both metrics are checked with respective threshold values to reduce frequent parent change and improve routing stability. These threshold values are used to select preferred parent from candidate parent set. The ranking of parents in candidate parent set is done on ETX, Hop Count metrics combined with the calculated threshold values to identify the best parent. The result shows reduced number of parent changes, improved stability, and reduced energy consumption. The model however resulted in reduced PDR when compared to RPL-ETX.

3. SYSTEM METHODOLOGY

3.1 DODAG Construction

RPL provides route for each node to reach the root of DODAG. This is done by comparing the rank of each node in the network generated by Objective Function (OF), and each node selecting its preferred parent. OF defines how nodes in a network selects its route to reach the Border Router (BR) within a RPL instance. OF is defined as Object Code Point in DIO packet configuration option and is advertised by each DODAG's root in an RPL Instance. The OF defines how the metrics obtained in DIO packet is interpreted and converted to Rank of the node. OF also defines how a node selects its preferred parent (PP). Rank can also be viewed as a node's relative position in the network with respect to BR.

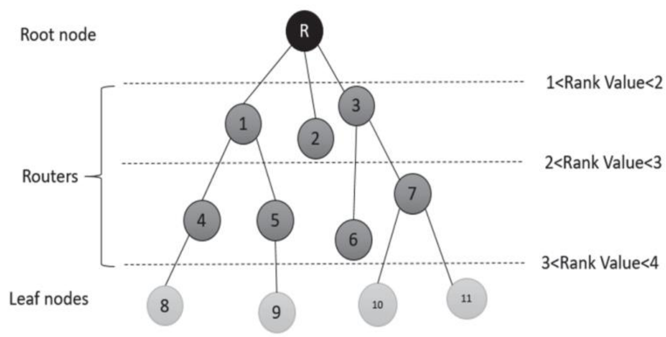

Figure 2 Sample DODAG

The identification and rectification of loops is a key characteristic of RPL. The direction of data flow is compared to the rank of the sender and recipient to identify the loop. For upward data transmission, the sender should be ranked higher than the recipient, and vice versa for downward data transmission. If this connection is not kept up, RPL recognises a loop. A node can sabotage the path to BR by announcing its rank as INFINITE RANK. Following that, every one of its children will select a different node as their preferred parent. By setting and sending a new DODAG-version, Root can initiate a complete DODAG reconstruction. All nodes must reidentify rank and pick new parent based on updated metrics and its location to reconstruct DODAG. The version number indicates how many times DODAG has been recreated in order to transmit data effectively. A node that gets numerous DIO from neighboring DODAGs might use this parameter to determine network's stability.

3.2 Objective Function (OF)

As there is OF to conduct the additional iteration to eventually decide on the most preferred route, it kept OF straightforward. This included the link level metrics in FOF and the node level metrics in OF in this study since they produced encouraging results. In this case OF ranks the parents based on expected transmission count (ETX) and cumulative path delay. ETX reflects the stability of the link between a node N and its parent P (Kim et al., 2016) and is determined as shown below.

$$ETX(node_n, node_p) = \frac{\text{Total number of transmissions from n to p}}{\text{Total number of successful transmissions from n to p}}$$

The total route delay between N and BR through P is the second metric of significance to FSOF. This metric has been considered in order to minimize network latency. The overall time delay in transmitting a packet from node N to BR is determined by the queue delay ($T_Q(N)$) at each forwarding node as well as the number of forwarding nodes required to reach BR (Jurdak et al., 2010). Because the channel is occupied, data transmission might be delayed at a forwarding node owing to the time it takes to go from sleep to active mode ($T_{sw}(N)$) and the back-off time (Tb_off (N)) the node needs to wait before transmission. Using below Equation, a node N in the network estimates the time delay through parent P to reach BR.

$$T_d(N,P) = T_Q(N) + T_{trans}(N) + T_{b_off}(N) + T_{sw}(N) + T_d(P)$$

The overall delay of the total route from parent P to BR is $T_d(P)$. Node N calculates its routing metric OF (N,P) through its candidate parent P, as shown below.

$$OF(N,P) = \alpha.ETX(N,P) + \beta.T_d(N,P) + MinHopRankIncrease$$

The minimal rank value that must be raised from the rank of parent to the rank of child is represented by MinHopRankIncrease. While calculating the rank, we add weights α and β to the equation to represent the relative significance we give to ETX and T_d. After identifying P_i as its parent, node N's rank is computed as shown below.

$$Rank(N,P_i) = Rank(P_i) + FS - OF(N,P_i)$$

After OF has ranked all of the potential parents, we choose the best n number of parent nodes for the parent set P_{OF} {P_1, P_2, P_3, ..., P_n}, where the OF must find the optimal parent $P_i \in P_{OF}$. The node N will advertise the Rank(N,P_i) in Eqn. 4.4 as its rank in the DIO packet once the OF determines the optimum parent P_i.

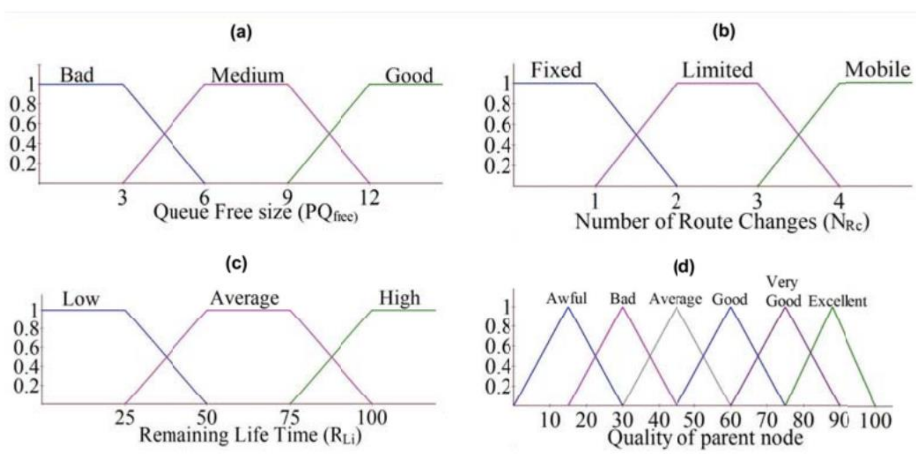

Figure 3 OF function

Figure 3.2(a), the linguistic variables for $P_Q$free of a parent are set to bad, medium, or good, indicating a node's relative queue free size available for accommodating a child's data. As illustrated in Figure 3.2(a), the parent node with a queue free size of 3 packets or less after accommodating traffic from the child node is the least favored option and hence belongs to the 100% bad set. Beginning at 6, a parent node is no longer in the poor set and belongs exclusively to the medium set until 9. Nodes with a queue free size of 9 to 12 packets belong to the medium and good sets, whereas those with a queue free size of more than 12 packets are deemed good. As indicated in Figure 3.2(b) and 3.2(c), we have divided N_{Rc} and R_{Li} into three linguistic variables.

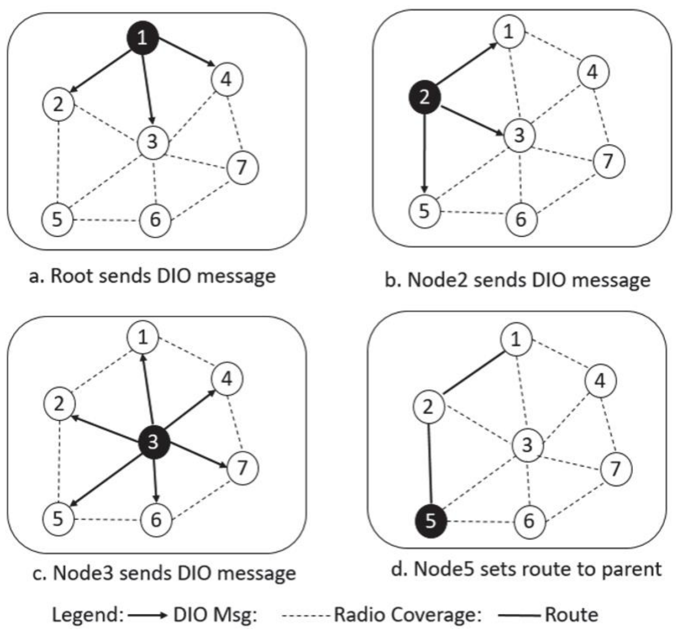

Figure 4 RPL establishing route using DIO message

Figure 4 shows how to create a route in RPL using the DIO message. It demonstrates how a DIO message originates at Root (node 1) and is passed on to succeeding networks. When a node receives DIO from a parent, it can only alter the Rank and DTSN values in the packet and should pass it to any children it may have. DODAG root sets the other fields in DIO as indicated before, and this value should be unaltered by all nodes. DIO may transmit an optional message called DIO Metric Container, which provides routing metrics and limitations that the recipient can use to determine which parent is preferred.

3.3 Load balancing Mobility aware Hybrid routing enhancement for RPL (LMH-RPL)

Static Nodes (SN) and Mobile Nodes (MN) are the two types of nodes in a DODAG. MNs are not allowed to be parent nodes in LMH-RPL to prevent major route failures in the network owing to its mobility. First, we will look at how SNs choose their parents, and then we will discuss about how MNs in the network select its parents. It is assumed that the MNs are preconfigured to join DODAG as leaf nodes and there are no malicious nodes in the network.

3.3.1 Static Node parent selection

In a two-stage procedure, SNs in the network identify preferred parents. To begin, we rank the parents using the OF and the metrics expected transmission count (ETX) and cumulative route delay, as specified by the RPL specification. The ETX of the child-parent relationship is determined using the Eqn. 1

$$ETX(node_c, node_p) = \frac{\text{Total number of transmissions from c to p}}{\text{Total number of successful transmissions from c to p}}$$

where $node_p$ is the parent node and $node_c$ is the child node, then ETX will ensure that the link between them is reliable. Since link quality is critical for reliable data transmission in the network, we employ ETX as a routing metric in this model. The overall time a packet takes to travel from a node to a BR is referred to as cumulative route delay. Inducing route delay as a metric aids in maintaining acceptable QoS for data transfer in the network. Three components have a role in the overall path delay. i) the time it takes for each node in the route to move from sleep to active state (T_{sw}(N)). ii) the backoff time (Tb_{off} (N)) that each node must wait to avoid network contention, and iii) the total amount of time a packet must spend in the wait queue (T_Q(N)) for each node in the path. The below equation a child N calculates the cumulative delay of the path through candidate parent P.

$$T_d(N,P) = T_Q(N) + T_{trans}(N) + T_{b_off}(N) + T_{sw}(N) + T_d(P)$$

The cumulative delay of the path provided by the parent node P to reach BR is T_d(P). Finally, as seen in Eqn. 3, the SN in our model ranks each of its parents.

$$Rank_{SN} = \alpha \times ETX(N,P) + \beta \times T_d(N,P) + MinHopRankIncrease$$

To regulate the influence of ETX and Td on identifying the rank of a parent node, by selectively adding α and β as weights to Eqn. next stage of route identification, Node N rates all of the potential parents and chooses the n best ranked parent P_{SN} = {P_1, P_2, P_3,...,P_n}.

3.3.2 SN Multipath Route

The child node at this step transmits and receives two customised packets from every Pi PSN, referred to as Child Query (Cquery) and Parent Response (Presp), respectively, in order to obtain more information for selecting the best route. The child node sends the Cquery with the flag value set to 1 to the top n parents. Parent nodes must answer to a Cquery request with a Presp that includes the following metrics: remaining energy, current data rate via it, and average queue free size. This reply Presp packet's flag value is set to 1.

The data rate of a parent node is calculated by summing the data rates of each of its children d_i and its own data rate d_p. We employ an EWMA filter with a 65% weightage on the current value to eliminate the transient surges in data rate. A parent node with q children determines its data rate D_p as given in Eqn. 6.4

$$D_p = \sum_{i=0}^{q} d_i + d_p$$

The Queue free size of parent P (P_{Qf}) is calculated by using transmission rate of parent (P_{trans}), queue size (Q_{sz}), and D_p as given in Eqn. 6.5.

$$P_{Qf} = Q_{sz} + P_{trans} - D_p$$

The child node calculates the lifespan RL of each parent assuming the child starts transmitting via it after obtaining these three metric values from a subset of parents. The child node employs the following to determine remaining lifespan, its own data rate Dc, the data rate handled by the parent node Dp, and the energy remaining Er in the parent node.

$$R_L = \frac{E_r}{D_p + D_c}$$

By sorting all of the parents $P_i \in P_{SN}$ in decreasing order of RL, the child node may now determine its primary path to BR. The preferred parent is determined as the parent node $P_i \in P_{SN}$ with the maximum R_L, which can live the longest after accepting the data transfer from the child node. Using Eqn. 6.7, child node N estimates its rank for dissemination in its DIO

$$Rank(N, P_e) = Rank(P_e) + Rank_{FN}(P_e)$$

In the event that the primary link fails or cannot deliver appropriate QoS, each child node retains an alternative path to BR. Nodes in set P_{SN} are sorted in decreasing order of P_{Qf} to determine alternative path. The first node in this list (P_q) is recognized as the alternative route since it has the largest queue free size. If $P_q = P_e$, the second-best parent node with P_{Qf} is chosen as P_q and designated as the secondary path to BR.

Algorithm 1 Parent identification of Mobile Node N

```
procedure IDENTIFY NEXT PP
    broadcast Cquery (flag = 0) to SNN
    if receive no Presp from SNNs then
        broadcast Cquery (flag = 3) to MN
        receive Presp from MNs
        nextPP = MN with best RSSI values
    else
        receive Presp from SNN
        store SNNs RSSI into RSSI_curr
        store SNN's Q_sz,R_en
        for each SNN compare(RSSI_curr[],RSSI_old[]) do
            Sort n SNN with highest RSSI gain to RSSI_sel[]
        end for
        while node in RSSI_sel[] do
            if nodeQ_sz and R_en > 10% available then
                nextPP=node
            end if
        end while
        if nextPP = null then
            nextPP = first node in RSSI_sel[]
        end if
    end if
    return(nextPP)
end procedure
```

4. RESULT AND DISCUSSION

All the experiments are done in NS2 simulation, 4 GB of memory and running on windows 10.

4.1 Ratio of Delivered Packets

The packet delivery ratio is calculated by dividing the total number of packets received at the BR by the total number of packets generated at the source nodes. The PDR of RPL, OF-FL, TSOF, mRPL, MEQA-RPL, and LMH-RPL for MN moving at 1m/s is shown in Figure 5. The LMH-RPL outperforms the other RPL variants in terms of PDR thanks to its abilities to handle mobility and balance load. Since congestion may force a node to use a different route to get to BR, it has no impact on the PDR in LMH-RPL. Our method can therefore offer enhanced PDR in a range of mobility scenarios. The suggested approach, MEQA-RPL, mRPL, TS-OF, OF-FL, and RPL have average PDRs for MN at 1 m/s of 91.2%, 86.7%, 83.8%, 76.4%, 71.8%, and 58.6%, respectively. The findings imply that the proactive seamless changeover support provided to MN in the network, along with the load balancing function of LMH-RPL, enhance the PDR of the network.

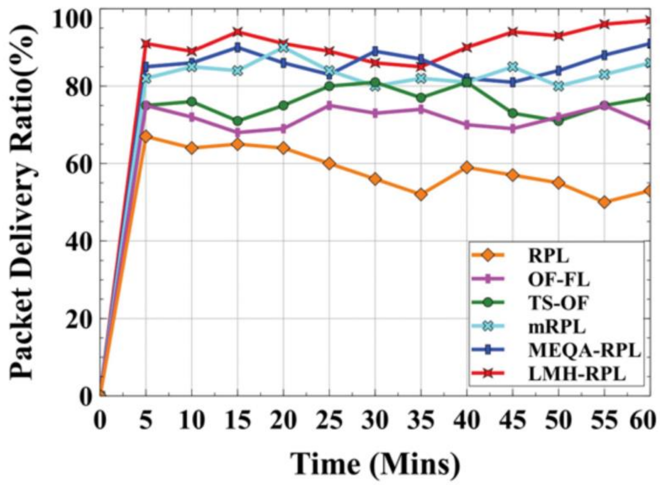

Figure 5 Packet Delivery Ratio MN at 1m/s

4.2 Average End to End delay

It is measures the typical amount of time a packet takes to travel through a network from source to recipient (EED). Figure 6 depicts the RPL, OF-FL, TS-OF, mRPL, MEQA-RPL, and LMHRPL EEDs with MN moving at a speed of 1 m/s. Due to the fact that route latency is one of the measures in OF used by LMH-RPL and MEQA-RPL to rank parents, these algorithms have lower EED than other RPL methods. The effectiveness of LMH-RPL is attributable to its capacity for load balancing. This ensures that packets are transmitted from every node without much delay, even while PP experiences congestion. Comparing our suggested model to MEQA-RPL and mRPL for MN at 1 m/s, we are able to lower EED by 11% and 15%, respectively.

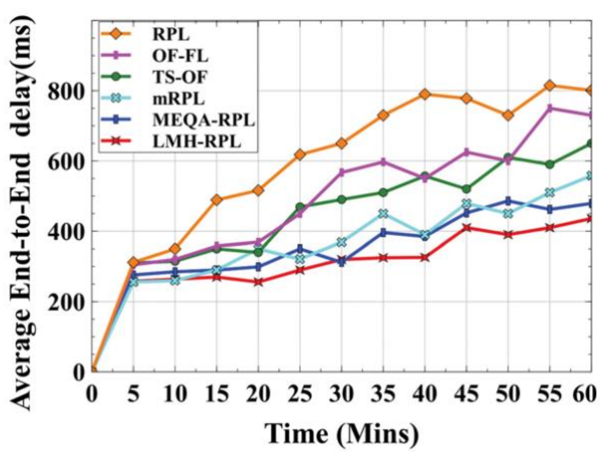

Figure 6 Average End to End delay MN at 1m/s

4.3 Energy consumption

Figure 7 shows the amount of energy used by nodes to maintain the network's pathways. This section examines the energy needed for DIS, DIO, DAO, and special packets used in the suggested model, such as P_{resp} and C_{query}, during transmission and receipt. MN is moving at 1 m/s. MN's decision to select a parent who can provide the most coverage is what led to the decreased energy use. As a result, the network experiences fewer route adjustments, which reduces the energy needed for route maintenance. The three models MEQA-RPL, LMH-RPL, and mRPL, which need MN to act as leaf nodes in the network, outperform earlier RPL algorithms as the degree of mobility increases. This is so that there won't be as many route delays brought on by MN. The proposed technique, mRPL, MEQA-RPL, OF-FL, TS-OF, and RPL had average energy consumption of 0.092J, 0.102J, 0.106J, 0.117J, 0.133J, 0.174J for MN at 1m/s respectively.

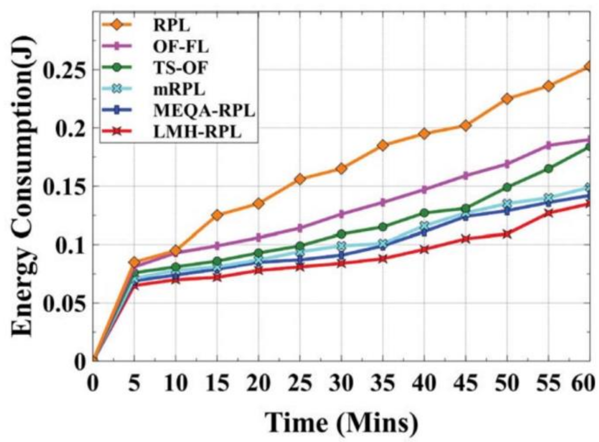

Figure 7 Energy consumption for MN at 1m/s

4.4 Number of living nodes

During the beginning phases of DODAG construction, each node initially uses a substantially larger amount of energy before returning to normal. Given that the remaining energy metric is provided as a threshold parameter in LMH-RPL, MN would not choose an SN with less remaining energy as their PP. By successfully creating the most stable channel for MN, LMH-RPL reduces the amount of network control packet exchange. By doing this, nodes with less leftover energy would live longer. By dividing energy use throughout the NN, the load balancing feature decreases network stress. A node's position, rank, movement, distance, and other criteria affect how much energy it uses. LMH-RPL provides more evenly distributed energy consumption in the network compared to other RPL algorithms, extending the life of network nodes. In all mobility scenarios, LMH-RPL was able to maintain a larger number of nodes in the network.

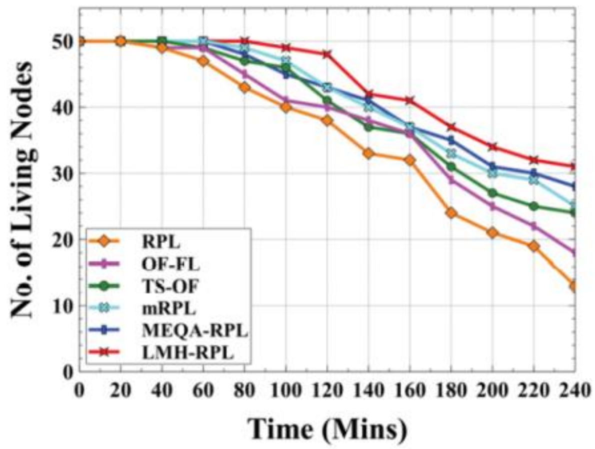

Figure 8 Number of living nodes for MN at 1m/s

5. CONCLUSION

The research work has contributed to developing a mobility module for RPL. The module also helps to find parent with minimum remaining energy for MN which can accommodate in its radio range for maximum amount of time. It is also noteworthy that ETX metrics is used by majority of OF compositions as good transmission link will reduce retransmission and help save energy. The OF can be tweaked to adapt to another situation by changing only the weight values. The Hybrid model developed over TSOF and MEQA-RPL called LMH-RPL provides excellent result in network. LMH-RPL's load balancing feature for SN greatly increases the resilience of the network towards momentary surge in congestion. LMH-RPL also provides MN the capability to relay data transfer for isolated MN in the network. The essence of this research work is to propose mobility support while conserving energy in RPL network.

References

[1] Altwassi, H. S., Pervez, Z., Dahal, K. and Ghaleb, B. (2018), The rpl load balancing in iot network with burst traffic scenarios, in '2018 12th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)', IEEE, pp

[2] Alvi, S. A., ul Hassan, F. and Mian, A. N. (2017), On the energy efficiency and stability of rpl routing protocol, in '2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)', IEEE, pp. 1927–1932.

[3] Bouaziz, M., Rachedi, A., Belghith, A., Berbineau, M. and Al-Ahmadi, S. (2019), 'Emarpl: Energy and mobility aware routing for the internet of mobile things', Future Generation Computer Systems 97(1), 247–258

[4] Farag, H., Osterberg, P., Gidlund, M. and Han, S. (2019), Rma-rp: A reliable mobility-aware ¨ routing protocol for industrial iot networks, in '2019 IEEE Global Conference on Internet of Things (GCIoT)', IEEE, pp. 1–6.

[5] Fotouhi, H., Vahabi, M., Rabet, I., Bjorkman, M. and Alves, M. (2019), Mobifog: Mobility ¨ management framework for fog-assisted iot networks, in '2019 IEEE Global Conference on Internet of Things (GCIoT)', IEEE, pp. 1–8.

[6] Gaddour, O., Koubaa, A., Baccour, N. and Abid, M. (2014), Of-fl: Qos-aware fuzzy logic ˆ objective function for the rpl routing protocol, in '2014 12th International symposium on modeling and optimization in mobile, ad hoc, and wireless networks (WiOpt)', IEEE, pp. 365– 372

[7] Harshavardhana, T., Vineeth, B., Anand, S. and Hegde, M. (2018), Power control and crosslayer design of rpl objective function for low power and lossy networks, in '2018 10th international conference on communication systems and networks (COMSNETS)', IEEE, pp. 214– 219.

[8] Kharrufa, H., Al-Kashoash, H. and Kemp, A. H. (2018), 'A game theoretic optimization of rpl for mobile internet of things applications', IEEE Sensors Journal 18(6), 2520–2530.

[9] Kharrufa, H., Al-Kashoash, H., Al-Nidawi, Y., Mosquera, M. Q. and Kemp, A. H. (2017), Dynamic rpl for multi-hop routing in iot applications, in '2017 13th Annual Conference on wireless on-demand network systems and services (WONS)', IEEE, pp. 100–103.

[10] Lalani, S. R., Salehi, A. A. M., Taghizadeh, H., Safaei, B., Monazzah, A. M. H. and Ejlali, A. (2020), Refer: A reliable and energy-efficient rpl for mobile iot applications, in '2020 CSI/CPSSI International Symposium on Real-Time and Embedded Systems and Technologies (RTEST)', IEEE, pp. 1–8.

[11] Lazarevska, M., Farahbakhsh, R., Shakya, N. M. and Crespi, N. (2018), Mobility supported energy efficient routing protocol for iot based healthcare applications, in '2018 IEEE Conference on Standards for Communications and Networking (CSCN)', IEEE, pp. 1–5

[12] Manikannan, K. and Nagarajan, V. (2020), 'Optimized mobility management for rpl/6lowpan based iot network architecture using the firefly algorithm', Microprocessors and Microsystems 77(1), 103193.

[13] Murali, S. and Jamalipour, A. (2018), 'Mobility-aware energy-efficient parent selection algorithm for low power and lossy networks', IEEE Internet of Things Journal 6(2), 2593–2601

[14] Murali, S. and Jamalipour, A. (2018), 'Mobility-aware energy-efficient parent selection algorithm for low power and lossy networks', IEEE Internet of Things Journal 6(2), 2593–2601.

[15] Sankar, S. and Srinivasan, P. (2018a), 'Energy and load aware routing protocol for internet of things', International Journal of Advances in Applied Sciences (IJAAS) 7(3), 255–264.

[16] Shin, Y. and Seol, S. (2020), Improvement of power consumption in rpl-based networks for mobility environment, in '2020 International Conference on Electronics, Information, and Communication (ICEIC)', IEEE, pp. 1–3

[17] Theodorou, T. and Mamatas, L. (2020), 'Sd-miot: A software-defined networking solution for mobile internet of things', IEEE Internet of Things Journal 8(6), 4604–4617

[18] Violettas, G., Petridou, S. and Mamatas, L. (2018), Routing under heterogeneity and mobility for the internet of things: A centralized control approach, in '2018 IEEE Global Communications Conference (GLOBECOM)', IEEE, pp. 1–7.

[19] Yadav, R. K. and Awasthi, N. (2020), A route stable energy and mobility aware routing protocol for iot, in '2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)', IEEE, pp. 942–948.

[20] Zhao, M., Chong, P. H. J. and Chan, H. C. (2017), 'An energy-efficient and cluster-parent based rpl with power-level refinement for low-power and lossy networks', Computer Communications 104(8), 17–33.