Performance Evaluation of ETXPC-RPL Routing algorithm in IoT Network

Zibuyisile Magubane
Department of Computer Science
University of Zululand
KwaDlangezwa 3886, South Africa
buyinkomose@gmail.com

Paul Tarwireyi
Department of Computer Science
University of Zululand
KwaDlangezwa 3866, South Africa
p.tarwireyi@unizulu.ac.za

Adnan. M Abu-Mahfouz
Council for Scientific and Industrial Research (CSIR)
Pretoria 0184, South Africa
a.abumahfouz@ieee.org

Mathew.O Adigun
Department of Computer Science
University of Zululand
KwaaDlangezwa 3866, South Africa
adigunm@unizulu.ac.za

***Abstract*— Nowadays physical objects have an ability to connect to the internet, process, analyze the data to the real world through the Internet of Things (IoT). These physical objects have a limited processing space, storage, memory, and bandwidth. Routing becomes a significant factor throughout the data transmission process in IoT devices. The network layer Routing Protocol for Low power and Lossy Networks (RPL) provides communication and the performance of information exchange among the IoT devices. The RPL protocol was mainly designed to perform data transmission effectively in the network with low dynamic traffic. Then, in large network traffic, heavy dynamic traffic scenarios it encounters network congestion problems due to load imbalance distribution. Several RPL load balancing routing algorithms have existed intended to improve network reliability. The Expected Transmission Count and Parent count (ETXPC) objective function is one of the load balancing objective functions that have been evaluated in a burst traffic network scenario. However, this study evaluates the performance of ETXPC objective function in the RPL general IoT network with regards to different network densities. For this study, we applied the parent count (PC), expected transmission count (ETX), and Buffer Occupancy (BO) routing metrics to guide the data packets. The performance metrics considered for evaluation were energy consumption, the ratio of packets delivered, control overhead, and end-to-end delay. İn large traffic network ETXPC provides unreliable links that decrease packet delivered successfully and increases control overhead. The algorithm balances the load distribution such a way that it reduces the network delay and energy consumption in the general IoT network.**

Keywords*— *IoT, Load balancing, RPL, ETX, Parent Count, Buffer Occupancy.

I. Introduction

Internet of things (IoT) remains a giant network that connects diverse, multiple objects (physical and virtual) to the internet [1]. These objects collect data in different environments and exchange data packets without human interaction. The data collected by the sensors embedded in various physical devices is forwarded to the IoT gateway. IoT gateway filters out the unnecessary data collected by the devices then forward the real data to the cloud for processing. The routing layer handles the data transmission from the initial node to the final node. Routing is important for decision making in which route can be used by nodes (devices) to send the data packets to the destination [2]. IoT applications use network layer routing protocols to optimize data transmission. The Routing Protocol for Low power and Lossy Networks (RPL) is based on a tree-like topology which was made by Routing Over Low power and Lossy Network (ROLL). It has been established by the Internet Engineering Task Force (IETF) group and standardized in the Request For Comment document (RFC 6550) [3]. RPL suites the needs of resource-constrained devices by providing reliable paths for transmitting the data among the devices. It is a proactive routing protocol that exchanges the data among the nodes in a form of acyclic graphs. RPL uses objective functions (OF) consist of routing metrics to compute the greatest path among the nodes. OFs minimize the cost of forwarding data among the nodes. The best link and the shortest path for transmission of data are permitted by the default RPLs objective functions. Whenever the network increases, the parent node of the best link and shortest path receive more packets than other nodes. Therefore it becomes congested, loses packets, and delays the processing due to unbalanced load distribution among the parent nodes[4]. Load balancing is a technique that distributes the traffic load equally on the nodes in the Wireless Sensor Network (WSN)[5]. The IoT applications produce a voluminous of data daily, therefore RPL is unable to optimize all the routes effectively in IoT network. Still, in RPL, the load balancing technique is required to improve the reliability of the network. Load balancing issue in the RPL protocol that has been studied by several authors in different network scenarios considering various performance metrics [4]–[9].

İn [9] the authors proposed ETXPC-RPL load balancing routing algorithm, the algorithm balanced the load through the parent count mechanism and the ETX expected transmission Count routing metric. Additionally, Minimum Rank with Hysteresis Objective Function (MRHOF) parameters has been applied to implement load balancing in RPL protocol. ETXPC algorithm was designed to handle traffic distribution in a burst traffic scenario. The study results depicted that the ETXPC-RPL method improved the ratio of packets delivered successfully and minimized the use of power consumption. However, the study was limited to the equal network density of (30 nodes) and different traffic scenarios (1, 20, 40 & 60) packet per minute. The performance metrics used to evaluate ETXPC-RPL in burst traffic scenario where the ratio of packets successfully delivered to the root node and the power consumed by the nodes during packet transmission. However, for this paper, the performance of the ETXPC-RPL algorithm is achieved by evaluating the algorithm in an increasing network density (initially includes ten (10) network nodes up to one hundred (100) nodes network nodes) in general IoT network. This evaluation provides the novelty of how this algorithm balances the load in small network size and large network size. While also giving the idea of how the network area affects the data exchange among the devices. The implementation of an algorithm has been processed in the Contiki Operating System. The network simulator Cooja was used to simulate the algorithm in the IoT network [10]. The evaluation is based on performance metrics such as network delay (End-to-End), the energy consumption of nodes, the control overhead, and packet delivery ratio.

This paper presents the work grounded on balancing the traffic distribution in IoT networks. This study considers implementing the existing load balancing algorithm ETXPC in RPL protocol to balance the traffic in the IoT network. However, ETXPC has been evaluated in a burst traffic scenario. This study introduces the buffer occupancy routing metric to detect congestion in parent nodes. Combine the parent count with Buffer occupancy to improve the quality of service. Highlights the effect of the network density in power consumption. Discusses the importance of choosing the best parent node when balancing the load distribution.

This paper is well-ordered into sections, the background study and Related are presented in section II. The Load balancing routing Algorithm is presented in section III while the second last section IV depicts the performance evaluation, and the results then section V concludes this work.

II. Background and Related work

A. Background

The network layer is the routing layer that arranges the performance of information exchange by providing the routing protocols that provide better routes for sending data over the internet. Proactive routing protocols update the routing table periodically according to the latest destination list and maintains the routing table. The Routing Protocol for Low power and lossy network (RPL) was standardized by ROLL-WG to provide efficient routing in Low power devices. The protocol obtains routing through objective functions. The Minimum Rank with Hysteresis Objective Function (MRHOF) and the Objective Function Zero (OF0) are the main routing objective functions for path optimization in RPL protocol. The OF0 allows the node to connect to the version of DODAG based on the best connectivity to the nodes without any specific routing metric for path optimization. OF0 chose the ideal parent node also backup the possible beneficiary if one is available. The routing of upward traffic is based on the ideal parent without considering any load balancing [11].

Minimum Rank with Hysteresis Objective Function (MRHOF) computes the route based on the Expected transmission count (ETX) metric. The objective function allows the nodes to compute the route of the least number of expected transmissions [12]. The nodes routing path is computed based on the available neighbor nodes. When a node fails to create the path to the root node, it becomes a leaf node after joining the candidate neighbor node. The IoT applications are increasing traffic every time, while RPL the objective functions are unable to balance the traffic load distribution efficiently. Therefore, the parent nodes become congested in large traffic networks, they lose packets in queues due to poor link quality. Network delay also occurs that results network failure to live longer.

RPL protocol has control messages such as Destination Information Object (DIO) that contains a rank of the root node and the DODAGID to other nodes in the network. DODAGs forms an upward route by releasing DIO control messages. The Destination Advertisement Object (DAO) is the control message that is used by the nodes to send a reply to the root node to join the DODAG and it forms a downward route. While the DODAG Information Solicitation (DIS) control messages are sent by the node when it did not receive a DIO message on time to solicit the DIO message from neighbors [13]. DIS control messages are responsible for adding nodes to the network and detecting new neighbors.

Fig. 1 describes the RPL tree topology in IoT networks. The DODAG root is indicated in blue color. The parent nodes indicated with red color while the yellow color indicates the child nodes. The DIO control message is sent by the root node to all its neighbor nodes asking them to join the network. The neighbor nodes to join the network they add the address of the DIOs sender at the end of their parent nodes list then computes its rank according to the Objective function used. And then all its neighbors receive a DIO control message with a new rank. If the nodes are stable connected to the network it can ignore the DIO message but if the node is not connected to the network it can process the DIO control message until all nodes in DODAG join the RPL network. The child nodes choose the parents grounded on the best links and the shortest path. Whenever the network increases the parent of the best link receives a massive volume of traffic that other nodes. Then that node becomes congested and fails to transmit all the packets on time.

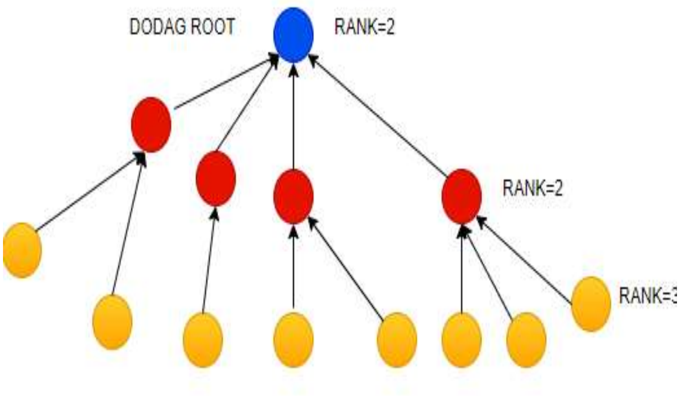

Fig.1 RPL tree topology

B. Related work

IoT devices generate massive data for different IoT applications. However, load balancing becomes an issue in RPL protocol and it has been evaluated in several prior studies recently like [6], [14]–[21]. İn [6] the scholars proposed the load balancing algorithm based on RPL called lbRPL. Their aim was to balance the load distribution in low power devices to improve RPL performance. The algorithm balanced the traffic through Load Balancing Indicator (LBI) which worked with the ETX, Remaining energy, and a Parent count to decide on which route is efficient to forward the data packets. The performance was evaluated in the Cooja network simulator. lbRPL outperformed the default RPL objective functions (MRHOF and OF0) in conditions of network stability and network lifetime.

Balancing the load distribution including the topology structure, the authors in [5] proposed the skewness and load balancing routing protocol based on RPL (SB-RPL) in IoT networks. SB-RPL used various routing metrics including Quality of the link and skewness among the subtrees of the network to construct the topology. The algorithm's performance was implemented in the Cooja simulator and tested in a real testbed. The SB-RPL achieved load balancing by reducing the child nodes among each subtree of the overloaded bottleneck node. The protocol improved tree structure, balanced child nodes on each parent node. This method mitigated the delay of the network and improved the packet delivered successfully than the RPL protocol. During the changes of parent nodes, the control overhead is increasing and deplete the energy of the nodes in the network.

Load balancing has been a huge problem in RPL protocol when it is supposed to provide communication among the nodes in dense networks, burst traffic scenarios. Expected Transmission Count and Parent Count (ETXPC) load balancing objective function was proposed by authors in [9] for RPL protocol to balance the load in a burst traffic network scenario. The objective function selects the best parent of the good link to guide the data packets to the root node. Furthermore, the algorithm was tested on the fixed network volume and the performance metrics considered only Packet delivery ratio and energy consumption. For burst traffic network ETXPC-RPL improved packet delivery ratio and minimized energy consumed by the nodes. However, this algorithm does not provide the volume of control messages generated during packets transmission. The time consumed by the packets to reach the destination.

The objective function to avoid congestion in the RPL protocol for IoT network proposed by authors in [22]. The routing algorithm CoAR alleviates network congestion by utilizing the alternative parent node. Whenever the reliable parent node is congested, the algorithm presents several routing metrics to choose an alternative parent node. The algorithm balanced the load distribution, used congestion detection mechanisms, and optimal parent selection. As the nodes changing the parent nodes and reporting congestion, many control packets generated which may degrade network performance in IoT scenarios.

III. Load Balancing Routing Algorithm

The load-balancing objective function ETXPC was initially proposed to provide the best load distribution in the RPL protocol for burst network scenarios. İt balanced the load distribution in parent nodes by recognizing the best parents to route the packets. The ETXPC-RPL utilizes the MRHOF objective function to be used during the burst traffic scenarios [9]. The scheme uses the ETX metric for link quality and the parent count values to balance the load distribution. Many nodes are used as parent nodes. The number of parent count was compared by selecting a single parent with a minimum number of child nodes as a preferred parent. Hence more nodes had an opportunity to be chosen as an ideal parent regardless of their link (ETX) value.

The algorithm applies a multi-hop data collection. Whenever it detects the best efficient route to the sink it starts with managing the traffic to update and analyze the traffic route to discover the traffic mode whether regular or burst traffic. After discovering the traffic mode, it employs the thresh hold value of the ETX with parent count to notice the effective route to the root node. The best parent is chosen according to the volume of packets available in the node. If the parent node has fewer packets than other parent nodes, the child node forwards the packets regardless of the link quality. If the chosen parent nodes have an equal volume of packets, the child nodes calculate the ETX value or change the path to find another parent node.

In this study, the ETXPC routing algorithm is deployed in the general IoT network. Since the algorithm used a single node metric, we were attracted to detect the congested nodes. Then we adopted the buffer occupancy as a node metric to detect the network congestion in parent nodes. Whenever the parent node has several child nodes, the buffer space available determines whether the parent is congested or is still have space to welcome more packets. The available buffer size is defined by subtracting the currently occupied buffer space from the total capacity of the buffer[23]. This algorithm successfully balanced the load distribution by optimizing parent selection. The parent selection is optimized by applying multiple node metrics including parent count, the buffer occupancy, and ETX. Every child node has multiple choices of parent nodes to elect the best parent node. Then, the child node counts the parent nodes available and choose its parent node among them. The best parent node is determined by the strong link between the nodes, the available buffer space on the parent node to avoid network congestion.

The nodes generate control messages (DIO, DAO, DIS, and DIO-ACK) that vary with the stability of the network. The nodes send more regular updates when changes occur in the network [12]. The nodes generate control messages for reliable communication and avoid the collision in the network. The control messages help to lessen the node's power consumption and mitigate the network latency. The nodes generate more control messages to acknowledge packet transmission, advertise available routes, and so on.

The algorithm in this study is evaluated in a regular IoT network based on different network densities. The performance of the algorithm is founded on the Packet delivery ratio, power consumption, End-to-end delay, and control overhead. The performance metrics are essential to provide the reliability and the stability of the routing algorithm to the network.

Performance Metrics: The performance evaluation of the ETXPC-RPL algorithm in this study is based on the below-mentioned performance metrics. The algorithms reliability to the network is determined by the performance metrics.

- Packet Delivery Ratio (PDR)

Packet delivery ratio (PDR) stands for the volume of data packets sent and successfully reached the desired destination over the overall volume of packets that are sent by a transmitter during the end-to-end communication [24]. PDR is considered as an important performance metric that also represents the reliability of the routing algorithm. To measure the strength of the network.

$$Average\ PDR = \frac{TotalPackets\ Received}{Total\ Packets\ Delivered} * 100 \ldots.(1)$$

- Energy Consumption

The performance metric Energy consumption is determined by the power consumed by the nodes to collect and transmit the data packets between the source and the desired root. The total power consumption is the power consumed by the nodes in all states of packet processing, receiving, and transferring. The node consumes power in CPU, LPM, Tx, and Rx mode. The average power consumption consists of the average sum of all states of the node.

$$P\_Components = P_{cpu} + P_{lpm} + \mathrm{P}_{\mathrm{tx}} + P_{rx} \ldots.(2)$$

- End-to-End delay

The performance metric indicates the network delay is referred to as End-to-end delay. It is influential to the network performance it provides the time spend by the data packets traveling from the source to the root node. The nodes choose the shortest path and less congested parent nodes to mitigate network latency. In the less congested parent nodes, the processing of packets is fast.

$$\text{End\_to\_end}\ Dleay = \frac{recvTime - SentTime}{Total\ Packet\ Received} \ldots (3)$$

- Control Overhead

Overhead is the total number of Internet Control Message Protocol version 6 (ICMPv6) packets transmitted by the nodes in the RPL protocol. Control overhead is when the nodes report about finding the route, error messages, thus it results in the congestion in the network [25]. The root node generates its information to the neighbor nodes for joining the network, the neighbor nodes calculate the rank according to the objective function implemented in RPL DODAG then join the network. The neighbor nodes propagate their information for other nodes to join the network. If the nodes all joined the network, they ignore DIO messages but those who are not yet joined they will join the DODAG network based on the rank of their parent nodes. The nodes generate more packets during the redirection of packets to the root node. The higher volume of control overhead determines the instability of the links also depletes the energy in the nodes. The instability of the links resulting in the loss of packets delivered to the network.

A. Implementation

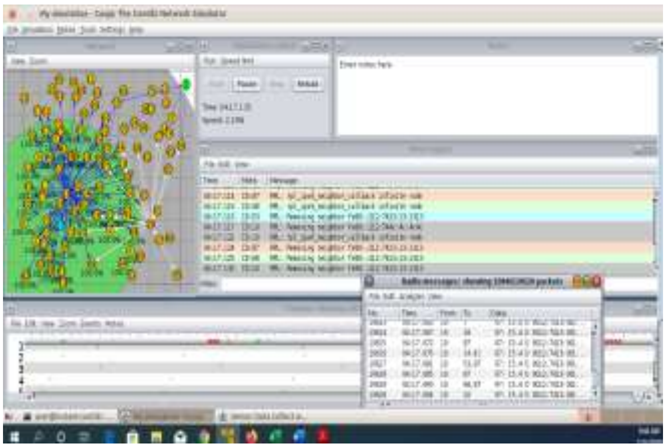

Fig.2 Cooja simulator

The fig.2 depicts the Operating System known as Contiki; It has the network simulator Cooja to evaluate the performance of routing algorithms [26].

The different windows of the network simulator displayed above were used to simulate the load balancing algorithm in RPL. The network window displays the scattered nodes where the nodes exchange the messages. The green node in the right corner represents the sink (root) node while the yellow-colored nodes represent the sender nodes. The nodes (node 37) transmission range represented by the green circle and the grey circle represents where the range where nodes interfere. The simulation control window is where the simulation is being controlled to start, pause, stop, and reload. The mote output window displays the simulation output. The radio messages show the packets delivered out of the packets sent (packet delivery ratio). The timeline window shows the LEDs of the nodes running on the simulator.

The programming language used to develop the routing metric is C language. RPL is flexible for modifications therefore, to implement the objective function ETXPC we used the client-server programming model on top of the existing RPL protocol.

The UDP protocol has been used for the implementation of client and server nodes. However, the sink node was represented by the server node at the same time as the clients represented by the scattered sender nodes in the simulation environment. The UDP sink sends the invite and waits for any incoming link in predefined port and then it receives the data packets from available sender then processes the packets received. The UDP clients join the UDP sink via IP and port of the sink which is known by all the nodes. The data packets were sent periodically form client nodes to the sink node. The sender nodes used multi-hop or single-hop data transmission while monitoring the distance between the nodes to choose the best link.

IV. PERFORMANCE EVALUATION

The performance of ETXPC-RPL compared to RPL is evaluated considering different network densities in regular IoT networks. The ETXPC-RPL load balancing algorithm was implemented in the Contiki 2.7 Operating System. The network simulator Cooja was used to evaluate the performance of the routing algorithms performance. To implement ETXPC-RPL, there are algorithms modified in RPL to fulfill the goal of this study. The rpl config.h, mrhof.c, udp-sink.c, and udp sender.c are the open-source codes that were modified for load balancing distribution in RPL Protocol. The collect-view shell was used to collect the results for energy consumption and packet delivery ratio. The Wireshark was used to collect the delay results and the control overhead results. The evaluations have been made in comparison with the default RPL objective function. Several simulations tests were conducted to verify its performance in the IoT environment.

A. Simulation setup

The ETXPC-RPL is designed for the general IoT network, it has an aim to distribute data over a random topology and maintain the balanced load distribution of data to improve network performance and avoid network congestion. The Contiki operating system has been used to implement the routing algorithm. Then the network simulator Cooja was used to conduct the simulations. Contiki is an open-source operating system reliable for connecting the microcontrollers with low power and tiny low-cost to the internet [26]. Cooja is a simulator embedded in Contiki OS, a flexible java-based cross-layer. The IETF protocols for Ipv6 network including RPL and 6LoWPAN are supported by Contiki OS.

These are the parameters were set for every simulation. For radio messages, Unit Disc Graph Model-Distance loss (UDGM-Distance loss) was used as the basic radio model of the Cooja network simulator. Every node has interference and transmission that are considered as a disk. For this algorithm, we used the RX success ratio variable for packet reception ratio and TX ratio variable for packet transmission ratio to minimize packet loss.

During simulations, at first, we used ten (10) nodes (Tmote sky) with a single sink node plus 9 sender nodes which were randomly distributed in the $100m^2$ network area. The simulations were repeated for ten iterations and move to the next network density of twenty (20) until 100 nodes. Every simulation run lasts for twenty (20) minutes. The summary of the simulation parameters is tabulated below.

TABLE I. SIMULATION PARAMETERS

Parameter	Value
Operating System	Contiki 2.7
Simulator	Cooja
The mote type	Sky mote
Radio environment	UDGM (Unit Disc Graph Model)-Distance loss
Number of Nodes (network density)	10, 20, 30, 40, 50, 60, 70, 80, 90, 100
Topology	Random
RX success ratio	100%
TX success ratio	100%
Simulation area	100*100
Simulation time	20 minutes

B. Results

The parameters measured for evaluation are the ratio of packets successfully delivered to the root node (packet delivery ratio), control overhead, end-to-end delay, and power consumption. The packets delivered to the root over the packet sent were collected in the radio messages window. Wireshark was used to collect the results for network delay, and the control overhead. The collect view shell embedded in the Cooja simulator was used in data collection of the nodes power consumption during the information exchange among the nodes. The results were collected to evaluate the performance of ETXPC-RPL in the general IoT network.

The performance analysis of the ETXPC-RPL routing algorithm depicted in graphical form based on the performance metrics was measured. Jupyter notebook was used to analyze the simulation results of the RPL and ETXPC-RPL routing algorithms. The ETXPC-RPL performance was compared with the default RPL in different network densities.

1) Packet Delivery Ratio (PDR)

The illustration in Fig. 3, illustrates the packets successfully delivered to the root node in percentage. The ratio of packets sent is increasing as the network usage increases. In a large network, the child nodes have many alternative parent nodes they can choose the reliable parent to route the data packets to the root (sink) node for processing. However, the nodes end up transmitting packets using poor links that lose packets along the way. In a low traffic network, the algorithm needs a reliable link to forward packets successfully. ETXPC-RPL delivers the same packets as RPL in a large network. ETXPC-RPL does not deliver 100% packets to the destination in the IoT network due to the poor link quality. RPL uses link quality as a metric to forward packets to the parent node. However, some packets are stuck on the queue and some get lost because it does not check the parent node's availability before it forwards the packets.

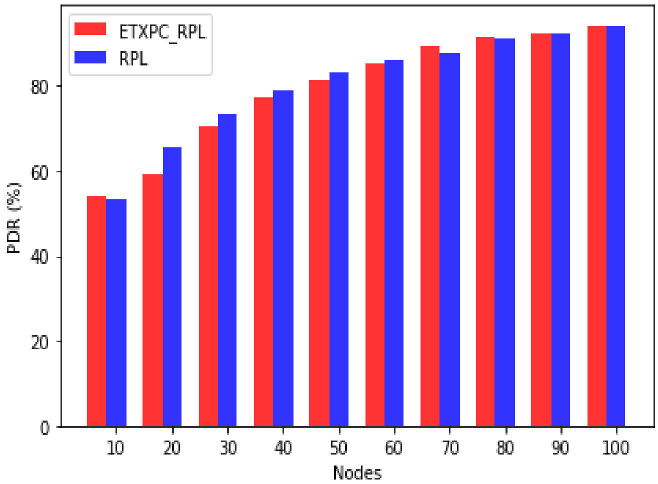

Fig.3 Packet Delivery ratio

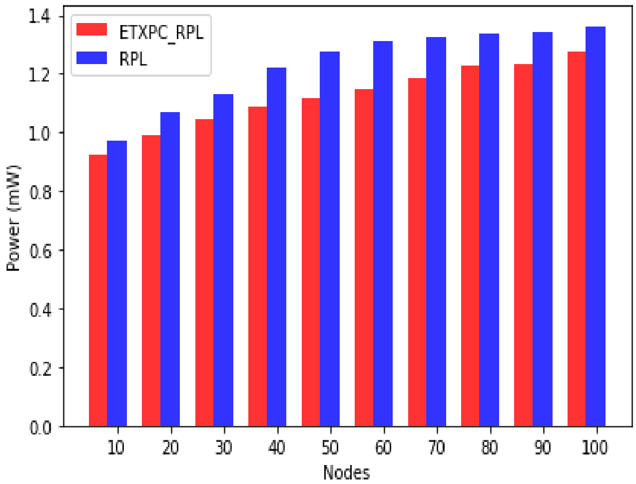

Fig.4 Power consumption

2) Energy consumption

Fig. 4 illustrates the results of the energy consumed by the nodes during the process of data packets transmission among the nodes. Most of the energy is consumed by the radio lossy links when they broadcast also retransmit the data packets. Whenever the network increases, the power consumed by nodes is also increasing. Most of the nodes they consume more power during the transmission of the packets. As the network increases more preferred parent nodes are processing a volume of packets from many child nodes, therefore they consume more power in high network traffic to ensure all the packets are processed and transmitted successfully. However, ETXPC-RPL consumes less power compared to RPL.

3) End-to-end delay

The latency of the nodes is referred to as the end-to-end delay performance metric. The metric determines the average time interval among the packets generated at the sender nodes until they are successfully delivered to the root node. The parent count and buffer occupancy metrics helped to minimize the delay of the packets. Fig. 5 depicts the delay which is increasing with the usage of the network. The Delay of packets is high in the large network because the processing is taking time in parent nodes. The ETXPC-RPL has less delay than RPL because the child nodes select the reliable parent nodes. In RPL more packets delay while the node is changing routes due to the unbalanced load distribution.

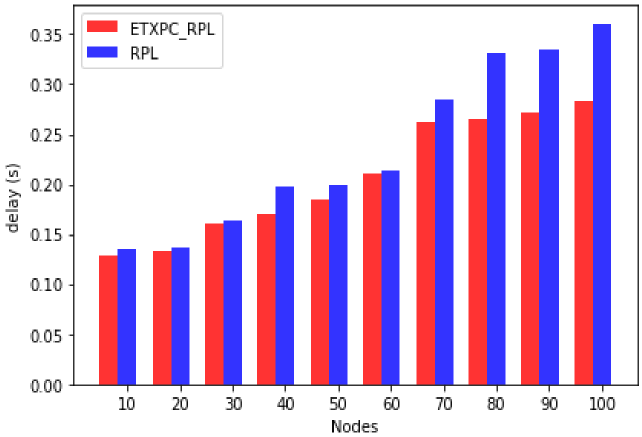

Fig.5. End-to-End Delay

4) Control overhead

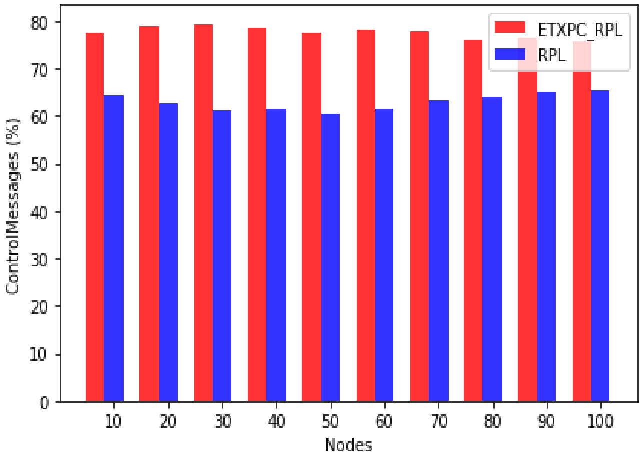

Fig.6 Control plane overhead

Fig. 6 depicts the total control messages that were generated during the transmission of packets. ETXPC-RPL algorithm generates more control messages. While RPL generates fewer control messages because it selects reliable links to forward the packets. İn ETXPC-RPL the nodes end up using many links to find the best path to the parent node. RPL does not consider the parent selection mechanism but the best link that makes the packets to stuck in long queues.

C. Discussion

The above graphs determine the performance analyses of the routing algorithm ETXP-RPL in IoT network different network densities. İn Fig. 3, the ratio of data packets delivered to the sink node over the data packets sent is rapidly increasing with the increase of the network usage. However, the ETXPC-RPL algorithm has unreliable links to forward packets to the preferred parent. The unreliability of the links is illustrated by the high control plane overhead of ETXPC-RPL in Fig. 6. İn general IoT ETXPC-RPL performs poorly in IoT networks than in burst traffic network based on PDR.

The node's power consumption during the data packets transmission is increasing as the network size increases. ETXPC-RPL consume less energy in general IoT network than in Burst scenarios. The congestion mitigation also reduces network delay. The algorithm mitigated the congestion through the buffer occupancy metric. Therefore, the network delay is reduced in Fig. 5. However, congestion mitigation increases the control overhead in the network.

İn RPL delay of the packets is high because of unbalanced traffic forwarded to parent nodes. The delay is high in the large network because all the parent nodes forward the packets to a single sink node. However, the waiting time to deliver the packets is increasing as the network usage increases.

V. CONCLUSION

The evaluation of the ETXPC-RPL load-balancing algorithm is conducted to analyze the performance of the algorithm in regular IoT networks regarding different network densities (sizes). The algorithm was previously evaluated and performed well in a burst traffic network of 30 nodes based on PDR and Energy consumption. Its improved packet delivery ratio and minimized the energy consumed by the nodes in burst traffic.

For this study, ETXPC-RPL utilized parent count, buffer occupancy, and ETX routing metrics. It minimized the energy consumption and the delay of the network. The algorithm balances the traffic distribution perfectly in child nodes. Due to a single link metric, the algorithm decreases the packet delivery ratio in the general IoT network. The single link routing metric ETX makes it difficult for the nodes to choose the reliable links for packet transmission in large network traffic. Therefore, more control messages being generated during the changing of paths and acknowledgment of the packets. This results in high network overhead.

ETXPC-RPL load balancing algorithm degraded the packet delivery ratio in general IoT network unlike in burst traffic where it improved the packet delivery ratio. However, the volume of the network in the IoT environment is increasing while in the burst traffic scenario there was a fixed volume of the network. Therefore, in the IoT network, ETXPC-RPL needs to improve packets delivered and mitigate the control overhead. Load balancing is an ongoing problem in the RPL protocol. ETXPC-RPL balances the load in IoT network although it generates high control messages that cause low data packet delivered. İn future this algorithm can be compared with other load balancing objectives functions that are currently proposed for RPL protocol. The ETXPC-RPL algorithm could be improved by introducing another link metric to provide a better packet delivery ratio. The time intervals should be considered to minimize the control overhead.

ACKNOWLEDGMENT

The authors acknowledge the support from the Center of Excellence, WMN cluster. We also acknowledge the support from the Council for Scientific and Industrial Research, Pretoria, South Africa, through the Smart Networks collaboration initiative and IoT-Factory Program (Funded by the Department of Science and Innovation (DSI), South Africa.

REFERENCES

[1] P. Internet, “IoTs – Overview of the Ecosystem Design and Planning requirements,” no. September, 2018.

[2] S. Santiago and L. Arockiam, “Energy Efficiency in Internet of Things : An Overview Energy,” *Int. J. Recent Trends Eng. Res.*, no. December, pp. 475–482, 2016.

[3] T. Winter *et al.*, “Rfc6550: Rpl,” *Internet Eng. Task Force (IETF), Req. Comments 6550, ISSN2070-1721*, pp. 1–157, 2012, doi: 10.1017/CBO9781107415324.004.

[4] S. Taghizadeh, H. Bobarshad, and H. Elbiaze, “CLRPL: Context-Aware and Load Balancing RPL for Iot Networks under Heavy and Highly Dynamic Load,” *IEEE Access*, vol. 6, pp. 23277–23291, 2018, doi: 10.1109/ACCESS.2018.2817128.

[15] L. Code, “Towards Efficient Load Balancing Strategy for RPL Routing Protocol in IoT Networks Graduate School of Seoul National University.”

[6] A. Sebastian, S. Sivagurunathan, and .., “Load Balancing Metric Based Routing Protocol for Low Power and Lossy Networks (lbRPL),” *Int. J. Eng. Technol.*, vol. 7, no. 2.22, p. 39, 2018, doi: 10.14419/ijet.v7i2.22.11806.

[7] B. Ghaleb, A. Al-Dubai, E. Ekonomou, W. Gharib, L. Mackenzi, and M. Bani Khala, “A New Load-Balancing Aware Objective Function for RPL’s IoT Networks,” *Proc. - 20th Int. Conf. High Perform. Comput. Commun. 16th Int. Conf. Smart City 4th Int. Conf. Data Sci. Syst. HPCC/SmartCity/DSS 2018*, no. January 2019, pp. 909–914, 2019, doi: 10.1109/HPCC/SmartCity/DSS.2018.00151.

[8] S. Id, "Evaluating the Effect of RPL- Based Load balancing Routing Protocols in IoT Distributed networks by Zibuyisile Magubane," 2019.

[9] H. S. Altwassi, Z. Pervez, K. Dahal, and B. Ghaleb, “The RPL Load Balancing in IoT Network with Burst Traffic Scenarios,” *Int. Conf. Software, Knowl. Information, Ind. Manag. Appl. Ski.*, vol. 2018-Decem, pp. 1–7, 2019, doi: 10.1109/SKIMA.2018.8631520.

[10] C. Thomson, “Cooja Simulator Manual,” no. C, pp. 2015–2016, 2016, doi: 10.13140/RG.2.1.4274.8408.

[11] 2019 Haines et al, “済無No Title No Title,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013, doi: 10.1017/CBO9781107415324.004.

[12] S. Kharche and S. Pawar, "Stability Model for RPL with Minimum Rank Hysteresis Objective Function in 6LoWPAN," *Int. J. Pure Appl. Math.*, vol. 118, no. 5, 2018.

[13] H. A. A. Al-Kashoash, Y. Al-Nidawi, and A. H. Kemp, “Congestion-aware RPL for 6L0WPAN networks,” *Wirel. Telecommun. Symp.*, vol. 2016-May, pp. 1–6, 2016, doi: 10.1109/WTS.2016.7482026.

[14] M.Qasem, “Load Balancing Objective Function in RPL draft-qasem-roll-rpl-load-balancing-01,” *Stand. Track*, no. October, pp. 1–10, 2018.

[15] B. Ghaleb, A. Al-Dubai, E. Ekonomou, W. Gharib, L. Mackenzi, and M. Bani Khala, “A New Load-Balancing Aware Objective Function for RPL’s IoT Networks,” *Proc. - 20th Int. Conf. High Perform. Comput. Commun. 16th Int. Conf. Smart City 4th Int. Conf. Data Sci. Syst. HPCC/SmartCity/DSS 2018*, no. June, pp. 909–914, 2019, doi: 10.1109/HPCC/SmartCity/DSS.2018.00151.

[16] Z. Magubane, P. Tarwireyi, O. A. Oki, and M. O. Adigun,

"Evaluating the Effect of RPL-Based Load balancing Routing Protocols in IoT Distributed networks," in *Southern Africa Telecommunication Networks and Applications Conference (SATNAC)*, 2019.

[17] P. Yin, S. Yang, J. Xu, J. Dai, and B. Lin, "Efficient Traffic Load-Balancing via Incremental Expansion of Routing Choices," *ACM Trans. Model. Perform. Eval. Comput. Syst.*, vol. 4, no. 1, pp. 1–35, 2019, doi: 10.1145/3243173.

[18] S. Santiago, A. Kumar, and L. Arockiam, "EALBA:

Energy-Aware Load Balancing Algorithm for IoT Networks," *Proc. 2018 Int. Conf. Mechatron. Syst. Robot.*, pp. 46–50, 2018, doi: 10.1145/3230876.3230880.

[19] B. K. Tarcisio Bruno Oliveira, Pedro Henrique Gomes, Danielo G. Gomes, "ALABAMO : A LoAd BAlancing MOdel for RPL ALABAMO : A LoAd BAlancing MOdel for RPL," no. May, 2016, doi: 10.13140/RG.2.1.1336.6648.

[20] H. S. Altwassi, Z. Pervez, K. Dahal, and B. Ghaleb, "The RPL Load Balancing in IoT Network with Burst Traffic Scenarios," *Int. Conf. Software, Knowl. Information, Ind. Manag. Appl. Ski.*, vol. 2018-Decem, no. February 2019, 2019, doi: 10.1109/SKIMA.2018.8631520.

[21] A. Sebastian and D. S. Sivagurunathan, "Load Balancing Optimization for Rpl Based Emergency Response Using Q-Learning," *MATTER Int. J. Sci. Technol.*, vol. 4, no. 2, pp. 74–92, 2018, doi: 10.20319/mijst.2018.42.7492.

[22] K. S. Bhandari, A. S. M. S. Hosen, and G. H. Cho, "CoAR: Congestion-aware routing protocol for low power and lossy networks for IoT applications," *Sensors (Switzerland)*, vol. 18, no. 11, 2018, doi: 10.3390/s18113838.

[23] K. S. Bhandari and G. H. Cho, "Resource oriented topology construction to ensure high reliability in IoT based smart city networks," *Int. J. Syst. Assur. Eng. Manag.*, 2019, doi: 10.1007/s13198-019-00861-2.

[24] L. Nguyen and C.-K. Kim, "Towards Skewness and Balancing of RPL Trees for the Internet of Things," pp. 1–12, 2019, [Online]. Available: http://arxiv.org/abs/1903.01839.

[25] X. Liu, Z. Sheng, C. Yin, F. Ali, and D. Roggen, "Performance Analysis of Routing Protocol for Low Power and Lossy Networks (RPL) in Large Scale Networks," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2172–2185, 2017, doi: 10.1109/JIOT.2017.2755980.

[26] Thingsquare, "Contiki: The Open Source OS for the Internet of Things," p. 1, 2014, [Online]. Available: http://www.contiki-os.org/.