INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2022, VOL. 68, NO. 2, PP. 187-191
Manuscript received November 28, 2021; revised April, 2022. DOI: 10.24425/ijet.2022.139866

Enhancing IoT Performance via Using Mobility Aware for Dynamic RPL Routing Protocol Technique (MA-RPL)

Ahmed R. Zarzoor

***Abstract*—Nodes' aware-mobility in the Internet of Things (IoTs) stills open defy for researchers, due to the dynamic changing of routing path and networks' resource limitations. Therefore, in this study a new method is proposed called Mobility Aware - "Routing Protocol for Low power and Lossy Networks" (MA-RPL), that consists of two phases: in the first phase splitting the entire network into sub areas based on reference nodes with "Time Difference of Arrival" (TDoA) technique. While, the second phase, is about managing mobile nodes (MNs) in RPL according to the sub areas' ID. The Cooja simulator software has been used to implement and assess MA-RPL method performance, according to the data packet metrics (lost packet, packet delivery ratio PDR), latency and nodes' power usage in comparison with two methods: Corona (Co-RPL) and Mobility Enhanced (ME-RPL). The simulation results have been shown that the MA-RPL method consumes less nodes' energy usage, gives less latency with minimum data packet loss in comparison with Co-RPL and ME-RPL**

***Keywords*—Internet of Things (IoTs); Routing Protocol for Low power and Lossy Networks (RPL); Mobility Aware (MA-RPL); Time Difference of Arrival (TDoA)**

I. INTRODUCTION

THE Internet of Things (IoTs), is considered the next generation of the internet, that based on the idea of communication from Machine to Machine (M2M) [1]. Also, in this network, each machine has limited capabilities (low battery capacity, tiny memory storage) that enabling it to communicate with other machines and exchanged data without any human intervention. The RPL routing protocol for IoTs has developed by "Internet Engineering Task Force" (IETF) [2] as a standardized routing protocol for M2M communication in IoTs [3]. In RPL, the machines are utilized the tree structure for routing data between child and root node. The tree structure in RPL is called as "Destination Oriented Directed Acyclic Graphs" (DODAG). Where, the root node is considered as border router or sink node. Whilst, the parent node is elected dynamically via utilizing objective function (OF) based on metrics such as distance between (node and sink), number of hop and Expected number of Transmissions (ETX). The ETX is the number of retransmissions that needed to submit packet to the target node. Thus, small distance, minimum number of hop and low ETX value means target node is the ideal node to be elected as parent. [4-6].

In the dynamic network, each node moves freely from one DODAG to another DODAG. Which, required aware about the new node's location in the network. Besides, there are many challenges that need to be considered in the dynamic network. The first challenge, is the dynamic changing of routing path between the (child and sink) node in the DODAG, since nodes are not static and can be moved from one location to another one. The second challenge, is the time that required to specify the new MNs' position and to deliver data packet between pair of nodes (i.e., latency). The third challenge, is nodes' resource limitation such as small battery power, low calculation speed [7-9]. So, to overcome these challenges, some researchers utilized neighbor location information and searching for DIS message in order to detect the new location of MN [10]. Nevertheless, this technique has been used only on the child level without considering the parent level in the IoT network. While, others researchers, divided the entire network into regions. Where, there is unique code to each region. So, based on the code the new position of mobile node (MN) is specified. However, this technique does not consider the small number of nodes in each region (i.e., low density) [11-12].

Therefore, in this study a new technique is proposed to handle MNs for RPL based on dividing the network into sub areas (i.e., regions). Each region is specified via utilizing reference nodes, Received Signal Strength Indicator (RSSI) and trickier time threshold and region id. Also, the mobile node in each region is managed by using region ID. In another word, routing data between source node and target node is managed based on the region information. However, the rest of this paper is organized as follows: section II, explores related works, section III describes the study methods. While, the section IV discusses the implementation results. Finally, the last section includes the study's conclusion.

A. Standard RPL Protocol

The tree scheme is utilized in RPL protocol, that is called (DODAG) for routing data between source node (child) and root (sink) node. In a DODAG, the topology is defined via utilizing four control messages which are [4-6]: 1. DIS "DOGAG Information Solicitation", 2. DAO, "DODAG Information Object" 3. DIO, "DODAG Information Object" 4. DAO-ACK "Destination Advertisement Object Acknowledgement".

Ahmed R. Zarzoor is with Directorate of Inspection, Ministry of Health, Baghdad, Iraq, (e-mail: Ahmed.Arjabi@gmail.com).

However, the messages are used in two directions (upwards and downwards) to convey message in RPL, see figure 1. In downwards direction, the root (sink) node broadcast the DIO message that contains it id and OF information (hop count and ETX) to neighbors. Consequently, each neighbor node that gets the DIO message will add its' rank (which is the distance between its location and root node location) to the message and pass it to the neighbors. The DIO message is broadcasted periodically to ensure it reach to all the nodes in the DODAG structure. While, in the upward direction each node broadcast DAO message by utilizing the shortest route to reach the sink node. So, each node may need to send the DAO-ACK to guarantee the received of the DAO message. Whereas the DIS message, is used in case of a new node joins DOGAG structure.

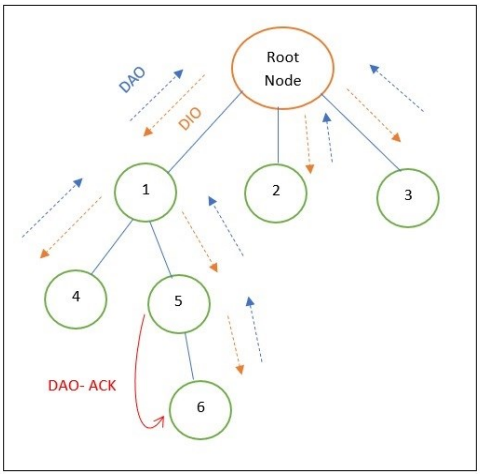


Fig. 1. Illustrates RPL DODAG structure [4].

II. Related Work

The RPL protocol is originally designed for static network. Therefore, many researchers have been worked on developing RPL structure in order to deal with mobile nodes (MNs). Sharma et al, is extended the standard RPL structure to enhance Energy usage in the network based on the regions ER-RPL [10] by dividing the network into sub areas. Where, each sub area is identified by id that is used to find for availability of MNs in the region. In the study [11] authors are proposed Corona-RPL (Co-RPL) method to deal with MNs. In Co-RPL the entire network is divided into groups. Each group forms a circle and called "Corona" (i.e., DODAG structure). So, the DAG root node position is allocated in the middle of circle and have a unique ID. The ID is used for identifying MNs among circles and to activate nodes neighbor discovery process in each corona. In [13] researchers are proposed Mobility Enhanced (ME-RPL) method to handle mobility of child node only. In ME-RPL they extended the structure of the DIO message by adding a new the filed called "mobility option" to identify status of node in the DODAG structure. Thus, the field option takes value (1 for MN or 0 for static node). However, in their method they considered only the child nodes are MN and the rest of the nodes are static. While, in [14] authors are used EMEER method to split the network into sub areas, each with code in order to provide an ideal point to point (p2p) paths for MNs. The EMEER is worked same as RE-RPL, but the only different that EMEER utilizes a set of nodes, instead of using all the nodes in the path discovery process. Also, they used the availability of MN location to minimize energy consumption. Besides, they handled the problem of non-uniform density regions, that appeared in RE-RPL via utilizing IRCM matrix that includes information (such as area code, number of nodes) of the source area and destination area.

Whilst in [15], the authors utilized "Trickle Timer" (TT) to develop RPL in the method called dynamic RPL (D-RPL) for IoT applications. The (TT) is the time that required to manage the recurrent sending of the DIO message. So, when the nodes' rank (i.e., distance between its' position and the target node' position) value is not change then TT value is reset. The main idea of D-RPL is about controlling TT period based on a comparison of the current RSSI value with RSSI value of the last received packets from MN, to minimize handover time latency of MN. Also, the [16] are presented "mobility compliant RPL" (mRPL) method to handle MNs in RPL via utilizing "smart-Hop" algorithm. Where, the MNs' position is detected according to average RSSI value. For instance, when the MN is moved through the overlapping area that is constructed from three nodes' coverage range. In this case, the MN make handover only to the node that have high average RSSI value in the overlapping area. So, the basic idea of mRPL is to minimize the handover times based on the average RSSI value.

In another study [17], authors are used "Kalman Position" KP-RPL technique to provide durable and trustworthy routing, considering the positioning inexactness and node isolation cases. Also, in KP-RPL each node creates its own confidence area based on anchor node that have most of a probably positions place. The anchor node position is used in Kalman filtering to reduce the error of position localization based on the previous routing decision. So, the MN can discard communication with a node that isolated due to the error of its' position allocation. While, in [18] a cross-layer method is used for IoTs based on neighbours' variability to prop MNs in RPL. In their method, the most static node is selected as parent according to the time differences of receiving RSSI value. Also, in [20] utilizes the same technique, but they used a timer called "T_ monitoring", which is the time that required for the sensor to detect packet period (i.e., packet recurrence measurement time). Consequently, they used T_monitoring to update routing table of neighbours based on RSSI value. So, if there is no preferred parent of MN and T_ monitoring is expired, then the MN sends DIS message to discover available parent and reset T_monitoring value

III. MA-RPL Method

The study method is designed in order, to improve the performance of RPL protocol in IoT that consisted of MNs, reference nodes and static sink node. The proposed method is divided in two phases which are:

A. Splitting network into sub-areas

In this phase, the whole network is divided into sub areas (i.e., regions) based on the knowing locations of static reference nodes, see figure 2. Each region has a unique code called ID and identified by utilizing four reference nodes. While, the location of the MN is specified in each region via using "Time Difference of Arrival" (TDoA) technique [20-22]. In TDoA, the unknown MN coordinate location (x,y) is allocated from the overlapping area of the hyperbole circles (i.e. the foci of the reference nodes used). Subsequently, the differences are computed based on the propagation times. To illustrate see figure 3, Where (R1, R2 and R3) are reference nodes and MN is the mobile node. The coordinate location of R1, R2 and R3 are (x2, y2), (x3, y3) and (x4, y4). The unknown coordinate location of MN is (x,y) and calculated by using equation 1 and 2. Where, T is the arrival times to reference node, V is the variation between the two arrival times and d is this distance between reference node and MN. Also, the range of variation (r) between MN and two reference nodes is calculated by using equation 3 and 4. While, the spread speed of the signal (s) is calculated by using equation 5. Where, Zi is the range variation error.

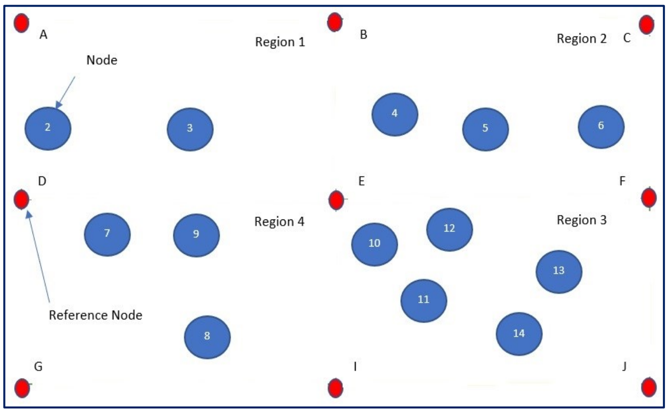


Fig. 2. Splitting network according to reference nodes

$$\sqrt{(x_2-x)^2+(y_2-y)^2}\text{-}\sqrt{(x_3-x)^2+(y_3-y)^2}=V(T_2-T_3) \quad (1)$$

$$\sqrt{(x_2-x)^2+(y_2-y)^2}\text{-}\sqrt{(x_4-x)^2+(y_4-y)^2}=V(T_2-T_4) \quad (2)$$

$$r_{i,1} = S|T_1 - T_i| \; i \in \{2,3,\dots,N\} \quad (3)$$

$$r_{i,1} = d_{i,1} + x_{i,1} \; i \in \{2,3,\dots,N\} \quad (4)$$

$$S|T_1 - T_i| = d_{i,1} + Z_{i,1} \; i \in \{2,3,\dots,N\} \quad (5)$$

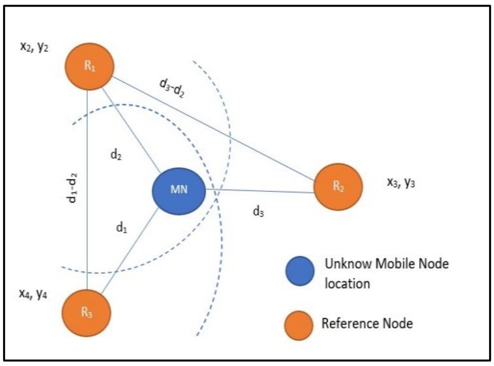


Fig. 3. Illustrates TDoA methods to allocate mobile node position in region

B. *Managing the MNs in RPL*

In the second phase, after splitting the network into regions, each region owns a unique ID and contains at least three MNs. The DOG is formed in each region as follow:

1. The root node of each DODAG structure will be considered as sink node.
2. The root node sends DIO message to all reference nodes that form the region.
3. Each reference node stores the region ID and MNs coordinate location with their timestamp that allocated in region.
4. The parent is selected, according to the ETX value and RSSI value.
5. The node that has low ETX value and strong RSSI value in comparison with other nodes in the same region will be elected as a parent.

In case of MN moving from one region to another, then the monitoring area (area, that formed from four reference nodes of each region) is used in that case. Thus, the MN location is specified according to the monitoring area ID (i.e., region ID). In this study, the DIS message content is extended by adding two new fields which are: region ID and timestamp, see figure 4. The region ID field, is updated with a new value by using reference nodes information. Consequently, a new nodes' position is identified via utilizing TDoA mechanism. To illustrate, let assume

1. A new node in the monitoring region sends a request message to all reference nodes that formed the region.
2. If the region is (empty or number of MNs less than two), then the message will be ignored until at least three MNs are allocated in this region.
3. Otherwise, each reference node will send ACK message to confirm the joining of new node in the region.

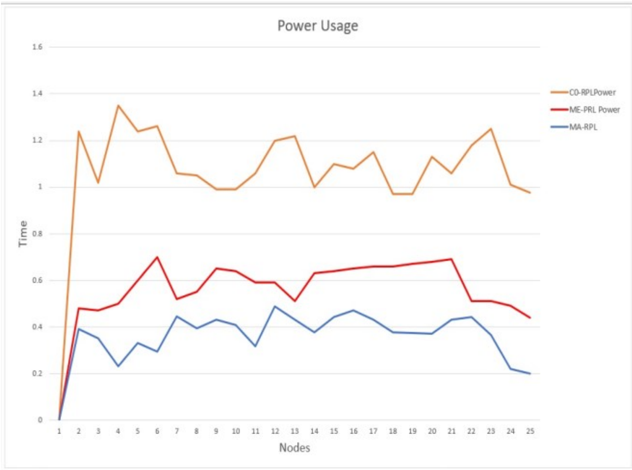
Region ID	Timestamp	Reserved	Option(s)

Fig. 4. Structure of extended DIS Message

IV. Result and Discussions

The study method is implemented by using Cooja simulator software. Three scenarios are created in Cooja simulator: MA-RPL, ME-RPL and Co-RPL. The first scenario is created for the MA-RPL method, in which number of nodes 25 used as follow: (static sink node=1, MNs =24) and reference nodes =9. While, in the second and third scenario for Co-RPL and ME-RPL, number of nodes 25 which are used as follow: number of static sink node=1, MNs=12 and static node=12). Also, all the nodes in the three scenarios are deployed in 100x100 meter area, see Table 1. Besides, the performance of MA-RPL, ME-RPL and Co-RPL is measured according to the route quality metrics: Packet Delivery Ratio (PDR), the number of drop packets, latency and node energy usage.

The implementation results of the three scenarios have been shown that: the nodes' energy usage in the Co-RPL method is higher in comparison with MA-RPL and ME-RP, see figure 5. For PDR, the MA-RPL gives heigh values (value=79 Kbps) in contrast with Co-RPL (value=74 Kbps) and ME-RPL (value=72 Kbps). Whilst, for Dropping packet both ME-RPL (value=120 packet) and Co-RPL (value=129 packet) are greater than the MA-RPL (value=103 packet). Even though, the number of MNs in both methods are half than the number of MNs in the MA-RPL, see figure 6. For latency (i.e., End to End Delay), the MA-RPL gives less latency for delivering the data packets to the destination node in comparison with ME-RPL and CO-RPL, see figure 7.



Fig, 5. Illustrates the power consuming for Co-RPL, ME-RPL and MA-RPL

TABLE I
SIMULATOR PARAMETERS

Parameter	Value
Operating System (OS)	Contiki OS Version 3.0
Area in Meters (m)	100 X 100 m
Total number of mobile nodes for MA-RPL	24
Total number of mobile nodes for ME-RPL	12
Total number of mobile nodes for CO-RPL	12
Total number of reference nodes for MA-RPL	9
Total number of static nodes for MA-RPL	1 sink node
Total number of static nodes for ME-RPL	12 node + 1 sink
Total number of static nodes for Co-RPL	12 node + 1 sink
Transmission Packet Ratio (TX)	100%
Received Packet Ratio (RX)	100%
TX and RX Range	100m
Network protocol	Contiki RPL
Start Delay	5 seconds
Simulation Time	60 minutes
Link failure model	UDGM with distance

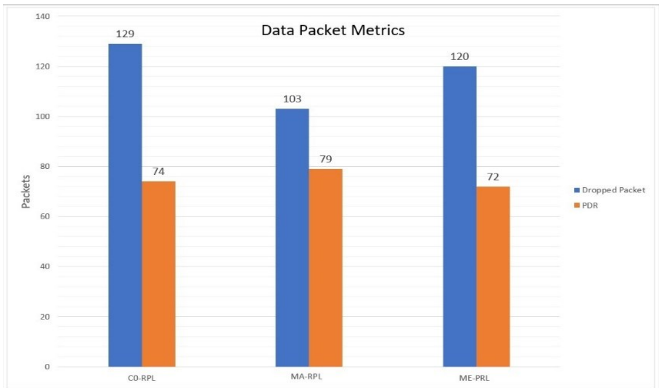


Fig. 6. Illustrates the data packet metrics: Dropped packet and PDR for ME-RPL, MA-RPL and Co-RPL

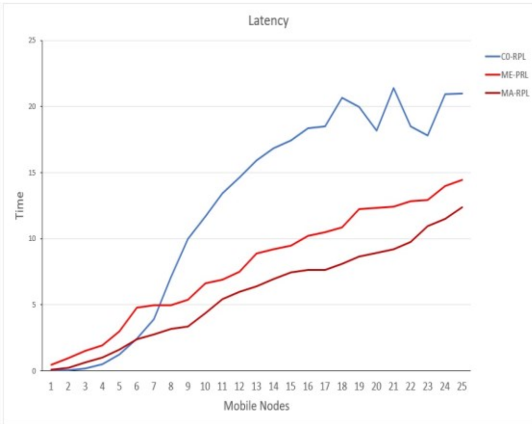


Fig. 7. Illustrates the latency for ME-RPL, MA-RPL and Co-RPL.

CONCLUSION

This study has been performed to aware about the MNs in RPL, in order to enhance the network performance based on the technique called MA-RPL that consisted of two phases: Splitting network by using reference nodes and Managing MNs in RPL. In the first phase, the network is divided into regions based on reference nodes information such as MNs' position information, timestamp and region ID. In the second phase, the MNs in the dynamic DODAG structure, is managed according to the region ID and ETX values.

However, three scenarios have been implemented in Cooja simulator software for the three methods: MA-RPL, RE-RPL and Co-RPL. The performance for the three methods has been done according to the network performance metrics: data packets metrics (lost packet, PDR), latency and nodes' energy usage. However, the evaluation process has shown that MA-RPL gives a smaller number of lost packets and higher PDR in comparison with ME-RPL and Co-RPL. Also, in the MA-RPL method, the network consumed less energy in comparison with ME-RPL and Co-RPL. Whilst, the MA-RPL gives lower delay in comparison with Co-RPL and ME-RPL.

ACKNOWLEDGEMENTS

I would like to appreciate all the excellent suggestions of anonymous reviewers to enhance the quality of this paper.

REFERENCES

[1] R. Prasad and V. Rohokale, "Internet of Things (IoT) and Machine to Machine (M2M) Communication," Cyber Security: The Lifeline of Information and Communication Technology, *Springer Series in Wireless Technology*, pp. 125-141, 2020. https://doi.org/10.1007/978-3-030-31703-4_9

[2] H. Tschofenig. and E. Baccelli, "Cyberphysical Security for the Masses: A Survey of the Internet Protocol Suite for Internet of Things Security," *In: IEEE Security and Privacy Magazine, Institute of Electrical and Electronics Engineers*, vol. 17, no. 5, pp.47-57, 2019. https://doi.org/10.1109/MSEC.2019.2923973

[3] [H. Kharrufa, H. Al-Kashoash and A Kemp, "RPL-Based Routing Protocols in IoT Applications: A Review, " *In: IEEE Sensors Journal*, vol.19, no.15, pp 5952-5967, 2019. https://doi.org/10.1109/JSEN.2019.2910881

[4] A. Zarzoor, "Optimizing RPL performance based on the selection of best route between child and root node using E-MHOF method," *International Journal of Electrical and Computer Engineering (IJECE),* vol. 11, no. 1, pp. 2088-8708, 2021. http://doi.org/10.11591/ijece.v11i1.pp224-231

[5] M. Farooq and D. Pesch, "Reduced Overhead Routing in Short-Range Low-Power and Lossy Wireless Networks,"u *Sensors*, vol. 19, no. 5, pp. 1-10, 2019. https://dx.doi.org/10.3390%2Fs19051240

[6] K. Haque, A. Abdelgawad, V. Yanambaka and K. Yelamarthi, "An Energy-Efficient and Reliable RPL for IoT," *in 2020 IEEE 6th World Forum on Internet of Things (WF-IoT),* pp 1-2, 2020. https://doi.org/10.1109/WF-IoT48130.2020.9221450

[7] P. Satanasaowapak and C. Khunboa, "The improvement of node Mobility in RPL to increase transmission efficiency," *International Journal of Electrical and Computer Engineering*, vol. 5, no. 9, pp. 4238- 4249, 2019. http://doi.org/10.11591/ijece.v9i5.pp4238-4249

[8] D. Bendouda, L. Mokdad and H. Haffaf, "Exploiting node mobility for fault management in RPL-based wireless sensor networks," International Journal of High Performance Computing and Networking (IJHPCN), vol. 12, no. 1, pp. 26-38, 2018. http://doi.org/10.1504/IJHPCN.2018.093839

[9] A. Musaddiq, Y. Zikria,, Zulqarnain and S. Won Kim, "Routing protocol for Low-Power and Lossy Networks for heterogeneous traffic network," *EURASIP Journal on Wireless Communications and Networking,* pp. 1-23, vol. 2020, no. 21,pp. 1-23, 2020. https://doi.org/10.1186/s13638-020-1645-4

[10] M. Zhao, I. W. Wang-Hei Ho, and P. Chong. "An energy-efficient region based rpl routing protocol for low-power and lossy networks," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 1319–1333, 2016. https://doi.org/10.1109/JIOT.2016.2593438

[11] O. Gaddour, A. Koubâa, R. Rangarajan, O. Cheikhrouhou, E. Tovar, and M. Abid, "Co-RPL: RPL routing for mobile low power wireless sensor networks using Corona mechanism," *in IEEE Industrial Embedded Systems (SIES), 2014 9th IEEE International Symposium*, pp. 200-209, 2019. https://doi.org/10.1109/SIES.2014.6871205

[12] Z. Latib, A. Jamil, N. Alduais, J. Abdullah, L. Audah, and R. Alias, "Strategies for a better performance of RPL under mobility in wireless sensor networks," *in Advances in Electrical and Electronic Engineering*: From Theory to Applications AIP Conference Proceedings, pp. 1-8, 2017. https://doi.org/10.1063/1.5002020

[13] I. El Korbi, M. Ben Brahim, C. Adjih and L. Saidane, "Mobility Enhanced RPL for Wireless Sensor Networks," in 2012 Third International Conference on The Network of the Future (NOF), 2012: 1-8. https://doi.org/10.1109/NOF.2012.6463993

[14] P. Sharma, V. Jain and A. Kumar Uprawal, "EMAEER: Enhanced Mobility Aware Energy Efficient Routing Protocol for Internet of Things," *in 2018 Conference on Information and Communication Technology (CICT)*, pp.1-6, 2018. https://doi.org/10.1109/INFOCOMTECH.2018.8722396

[15] H. Kharrufa, H. Al-Kashoash, Y. Al-Nidawi, M. Mosquera and A. Kemp, "Dynamic RPL for multi-hop routing in IoT applications," *in 2017 13th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, Jackson, WY, pp. 100-103, 2017. https://doi.org/10.1109/WONS.2017.7888753

[16] M. Radhesh Anand and M. Tahiliani, "mRPL++: Smarter-HOP for optimizing mobility in RPL," *in 2016 IEEE Region 10 Symposium (TENSYMP).* IEEE, pp. 36–41, 2016. https://doi.org/10.1109/TENCONSpring.2016.7519374

[17] M. Barcelo, A. Correa, J. Vicario, A. Morell and X. Vilajosana, "Addressing Mobility in RPL With Position Assisted Metrics," *IEEE Sensors Journal*, vol. 16, no. 7, pp.2151-2161, 2016. https://doi.org/10.1109/JSEN.2015.2500916

[18] J. Kniess and V. De Figueiredo Marques, "MARPL: A crosslayer approach for Internet of things based on neighbor variability for mobility support in RPL," *Trans Emerging Tel Tech*, pp. 1-17, 2020. https://doi.org/10.1002/ett.3931

[19] V. de Figueiredo Marques, J. Kniess, "Mobility Aware RPL (MARPL): Mobility to RPL on Neighbor Variability," *in Green, Pervasive, and Cloud Computing. GPC 2019. Lecture Notes in Computer Scienc*e, vol. 11484, pp.59-73, 2019. https://doi.org/10.1007/978-3-030-19223-5_5

[20] P. Wu, S. Su. Z. Zuo, , X. Guo, B. Sun, X. Wen, "Time Difference of Arrival (TDoA) Localization Combining Weighted Least Squares and Firefly Algorithm," *Sensors*, vol.19, no.2554, pp. 1-14, 2019. http://doi.org/10.3390/s19112554

[21] M. Bai, S. Lan and J. Huang, "Time Difference of Arrival (TDOA)-Based Acoustic Source Localization and Signal Extraction for Intelligent Audio Classification," *In: 2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 632-636, 2018. https://doi.org/10.1109/SAM.2018.8448583

[22] F. Khelifi, A. Bradai, A. Benslimane, R. Priyanka and A. Mohamed, "A Survey of Localization Systems in Internet of Things," *Mobile Networks and Applications*, vol. 24, pp. 761–785, 2019. https://doi.org/10.1007/s11036-018-1090-3