Optimized Dynamic Trickle Algorithm for Low Power and Lossy Networks

Muneer Bani Yassein[1], Muntaha Al-as'ad[1] and Dragana Krstic[2]

Abstract –**The industry is now converging towards the incorporation of Internet of Things (IoT) connectivity to their devices. Thus, a huge effort is invested towards the implication of well-established and efficient IoT devices. Due to the dramatic increase in the number of IoT devices, efficient network algorithms are required to manage and coordinate the communication between those devices. Routing Protocol for Low Power and Lossy Networks (RPL) is one of the routing algorithms that are used as routing protocol in these diverse networks. RPL suffers from several limitations. One of the major limitations is the short listening period. This limitation causes a lot of inconsistencies in the network such as high delay and dropping rate in the network. In this work, we introduce an adjustment of the trickle time that dynamically allocates the listening time based on the dropping rate in the network. The results of this work show significant improvement over other enhancement implementation on this protocol.**

Keywords –**RPL, IoT, Protocol, Dynamic Trickle Algorithm, Cooja, Objective Function, DAG, DODAGs, PDR.**

I. Introduction

Everything that we use is connected to the Internet. From this point, the concept of the Internet of Things (IoT) came up [1]. This concept consists of two parts: Internet and things. In IoT, things are represented by nodes such as laptops, devices, cars, fridges, etc. Each node in IoT has a unique address that is completely different from the other nodes in the network. In IoT, Network uses the protocol to operate as stander a routing protocol that is Routing Protocol for Low Power and Lossy Networks (RPL) to reduce energy use and save time. RPL is distance vector protocol, and it used in the Internet Protocol version 6 (IPv6) environment. It consists of five components: control massage, duty cycle, Objective Function (OF) [2], the routing metrics and the trickle timer. It uses three different types of the traffic: Multi Path Process Model (MP2M), point-to-multipoint communication (P2MP), and peer-to-peer (P2P) [3], [4].

The RPL uses the construction of the acyclic graph as a map representing the path between the nodes in the IoT network. It is built on the Directed Acyclic Graph (DAG) using the Objective Function (OF). In the RPL may be more DAGs. Each DAG contains more Destination Oriented DAGs (DODAGs) and is distinguished by giving an ID to each of them. Each DODAG contains only one root. Each node in the DODAG calculates its rank and chooses the preferred parent. It also represents a level in the graph. It is a value measured by the distance of the node from the root and is calculated by a specific OF previously specified in the DODAG. This is presented in Fig. 1 [5].

The DODAG is built as follows: First, a DODAG Information Object (DIO) broadcast message is sent to all neighbors. This message contains important parameters including DODAG ID, rank value, OF. This message is sent by the root. Each node in the DODAG receives the Information Solicitation (DIS). After receiving the message, the rank is calculated, and the preferred parent is selected. A reply to the DIS message from the bottom is called the Destination Advertisement Object (DAO). DAO is unicast because each node sends a reply to the DIS message. If you want a node to join the DODAG, you send a DODAG DIS broadcast to get the DIO message from neighbors in DODAG. You get this message from the nearest nodes as shown in the graph in Fig. 2 in [6].

A routing protocol that is used widely in the IoT environment is the RPL [7–9]. One of the main components of RPL is the trickle timer algorithm. This algorithm intends to make the information exchanging between the IoT nodes more simple, scalable, and energy efficient. This algorithm has a downflow that limits its usage. The listening time for this algorithm is short, in such a way that makes the nodes have no sufficient time to hear each other. Thus, this reduces the network performance. Several enhancements were suggested to overcome this short flow. This work addresses those variants and intends to enhance over the current trickle algorithm and its variants/enhancements [10-14] to achieve a more scalable and less power consumption RPL version to meet the needs for IoT low power/lossy network environment.

One interesting solution that addresses the limitation of the trickle algorithm is a dynamic version of this algorithm [15].

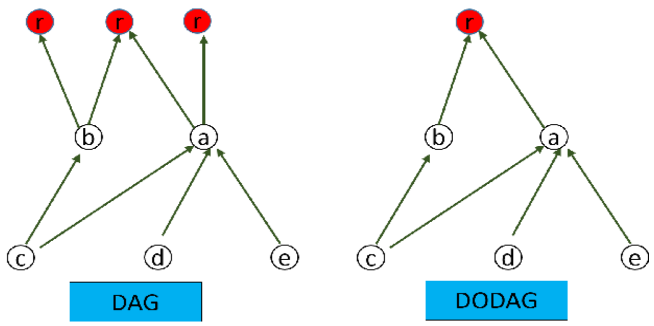

Fig. 1. DODAG and DAG graphs

[1]Muneer Bani Yassein and Muntaha Al-as'ad are with Department of Computer Science, Jordan University of Science and Technology, Irbid, Jordan, E-mails: masadeh@just.edu.jo, maalasaad17@cit.just.edu.jo

[2]Dragana Krstić is with University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Serbia, E-mail: dragana.krstic@elfak.ni.ac.rs

This approach dynamically allocates the listening period based on the number of neighbors for each node. We will extend this algorithm to incorporate an important factor to the dynamic trickle algorithm. One of the main factors that we will consider measuring the congestion in a network is the Packet Dropping Rate (PDR) [16], [17]. PDR gives sufficient information to dynamically adjust the waiting time based on the actual dropping rate that occurs in the network [18 – 21]. An Online Packet Dropping Rate (OPDR) estimation is sufficient to produce an optimal solution for trickle algorithm in IoT network. We will enhance the dynamic trickle algorithm by incorporating the OPDR. This approach is called Over-the-Air (OTA) testing.

To measure the performance and accuracy of our suggested algorithm, we need to develop the OTA in a real-life testbed. Due to the difficulty of doing this, we will implement OTA algorithm in a simulated network. Severa network simulators exist like Cooja, QualNet, OMNET, OPNET, and NS-3.

The remainder of this paper is present as follows: Section II shows the background of the study. Section III sheds the light on related works. Section IV describes the methodology proposed in this research. Sections V and VI give evaluation and results discussion, whereas Section VII concludes this research and provides possibilities for future work.

II. Background Information

The rise of the Internet of Things has revolutionized the legacy internet from conventional computers that are interconnected in a network with a plethora of devices, those devices are known as smart devices. Those devices include different physical entities (i.e., things) such as vehicles, home appliances, and wearable devices, and empowered with networking and computing capabilities [22]. Due to the pervasiveness property of the IoT network, caused by the big number of interconnected devices, such networks require a well-crafted algorithm to control the flow of messages between the devices. This algorithm must incorporate the needs for a protocol that tackle a low power lossy networks infrastructure. Different routing protocols exist for the IoT network infrastructure such as RPL [8].

The trickle timer algorithm is an important component in RPL protocol [23–28]. This algorithm aims to enhance the information exchanging between the IoT nodes. Due to the limitations of this algorithm, such as the short listening time, several enhancements were suggested to overcome this limitation. One of the solutions to this problem was suggesting the addition of the listening period only period in the first half of each interval. However, the solution causes another problem, which is an increment of delay in propagating transmissions to resolve the inconsistency situation as fast as possible. Another problem caused by listening to the only one period is the load balancing problem reported in [15].

III. Related Work

Yassein et al introduced in [15] the enhancement in a dynamic algorithm to calculate the waiting period. When calculating the time based on the number of neighbors, the calculation is long if the number of neighbors is large.

Ghaleb et al. showed in [28] the version of the algorithm trickle (E-Trickle) was a solution to the problem of short listening without a listening period. The results showed that the E-Trickle reduced the convergence time by 43% while maintaining the same energy efficiency and reliability. This did not entail the addition of any cost in energy consumption and loss of data.

In [29], the gap that appeared previously was bridged by dealing only with listening to period problems only. This is reflected in power consumption and time convergence. E-Trickle is integrated with the RPL routing protocol. Random topology was used in simulations, and the results regarding energy consumption were very impressive.

In [30], an amendment to the E-Trickle algorithm to become Trickle-plus is showed. The goal is increasing the flexibility of the protocol to build the network within the best time of convergence and energy consumption. The number of sent control traffic messages and confirmed experiments and simulations proves that.

Vallati et al. in [31] did a deep analysis and evaluation of RPL performance with a very special focus on Trickle algorithm and its parameter to improve network build, configuration time and power consumption. The nature of Trickle work can lead to a sub-optimal path, especially when the cancellation of messages is of a large extent. This problem is solved with a new algorithm which became Trickle-F to be effective in the collection of the best and more effective paths while maintaining the same energy consumption.

Djamaa et al presented in [32] a simple improvement that can significantly reduce the delay of the trickle. This is done during the period of listening only with no accompanying additions affect this work. The result showed that this improves the time of consistency of Trickle with results greater than a factor of 10 in reduction of the time of spread.

Meyfroyt et al analyzed in [33] the mathematical algorithm and a modified version of the Trickle algorithm was proposed with the addition of a parameter to determine the length of the listening period only. The addition of parameter increased the speed in the deployment of updates and control of the number of transmissions and affected the distribution of the transport load between nodes in the network. A mathematical model describing how the message count and inter-transmission times of the Trickle algorithm depend on its various parameters. The study also showed how constant frequency, network size, and length of hearing were used. The transmission process in a single network is similar to that of a Markov series.

Becker et al presented in [34] models for analyzing algorithm behavior that pertains to the time of consistency and to compare models and simulations in a different topology for the network to effectively detect the algorithm in the network.

Levis et al used in [35] three different work frames to evaluate the trickle. The first is abstract algorithmic simulator, written especially for this study. The second is TOSSIM which compiles directly from TinyOS code. Finally, TinyOS mica-2 motes for empirical studies are used to validate our simulation results and prove the real-world effectiveness of Trickle.

IV. Methodology

Given the intrinsic dynamic nature of the IoT, there is a need for an efficient algorithm to manage the routing of data between the nods. RPL is one solution, however, the trickle algorithm needed to be better optimized to cope with the needs of the IoT environment. The Dynamic trickle algorithms ignores the factors in the surrounding of network environment. When calculating the time based on the number of neighbors, it can take a long time if the number of neighbors is large.

To measure the PDR, we need a specific time windows to do the calculation of the dropping rate. To do this, we will experiment using different time windows and see how the solution converges based on the surrounding factors such as the number of networks, network latency, etc. This work requires an accurate measurement metrics to measure the performance of our algorithm. We will investigate the existing measurements and pick the best that meets our needs. Also, we will pick a most suitable network simulator.

The dynamic trickle algorithms ignore the factors in the surrounding network environment. Our work is intended to overcome this limitation by incorporating the packet dropping rate (PDR). PDR can give a real indication of the surround network environment. We want to calculate an online packet dropping rate. Based on the calculated value, we will calculate the new waiting time for the trickle algorithm.

There are two nodes exchanging information: node (A) and node (B). Node (A) sends some packets (S) = 1, ..., n packets to the node (B). Node (B) receives number of packets (R) = 1, ... , i=n from node (A). To calculate the dropping rate (D) between nodes (A) and (B)), the next formula is used:

$$D = S / R. \tag{1}$$

Based on the calculated value of the dropping rate, we will calculate the trickle time (T) as follows:

$$T = initial\ time * D \tag{2}$$

Initially, the nodes transmit at a specific rate. With increasing the number of nodes, some packets will be dropped. This dropping can measure the congestion in the network. Each node will calculate the dropping rate and will use this dropping rate to calculate the listening time in the trickle algorithm.

V. Evaluation

We evaluate two approaches using Cooja simulator [36], [37]. The two main implementations that were compared are: (1) our implantation for the dropping rate (optimized dynamic approach), and (2) the dynamic trickle algorithm that was proposed in [15]. We evaluate our work based on the average power consumption measured in mW.

We also investigate the packet delivery ratio, since it is the main indicator for the effectiveness of the network. The packet delivery rate measures the number of received packets over the entire packet that were sent between two nodes. We also use the inter-packet arrival time, which measures the time gap between the receiving of two consecutive packets.

Table I
Simulation Parameters

Parameter Name	Values
Simulator	Cooja 2.7
Number of Nodes	20,40,60,80
Simulation Time	900 seconds
Imin	2^{12}
Imax	2^{20}
Redundancy Factor (K)	1
Data Packet Rate	60 second
Transmission Range	30 m
Reception Success Ratio	20,40,60,80,100 %
Interference Range	30 m
Objective Function	MRHOF
Network Topology	Random
Radio Medium	Unit Disk Graph Medium (UDGM) Distance Loss

For the previous measurement, we calculated the average of those measurements among the entire number of nodes. Table 1 summarizes the main simulation parameters that were used in our experiments.

VI. Discussion of Results

The results of the comparison between the two models are discussed in this section. Average power consumption versus number of nodes is presented in Fig. 2. This figure shows the comparison between two approaches, dynamic and optimized. It is visible that the optimized approach exceeds the dynamic algorithm in terms of power consumption, i.e. shows better performance.

Fig. 3 shows the average packet delivery ratio comparison between these two approaches. It shows that optimized approach again over-performs the dynamic algorithm, now in terms of average packet delivery ratio. This can be depicted since the optimized approach has a higher delivery ratio.

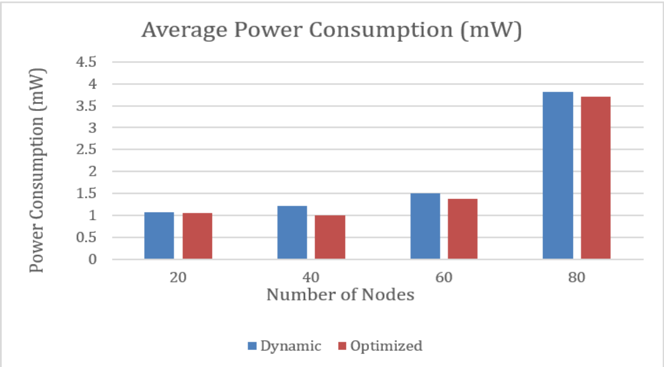

Fig. 2. Average Power Consumption

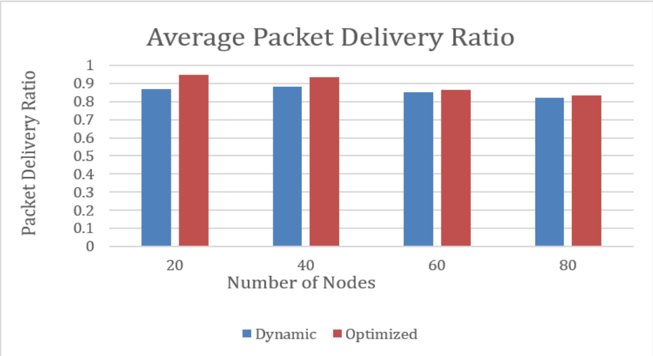

Fig. 3. Packet Delivery Ratio

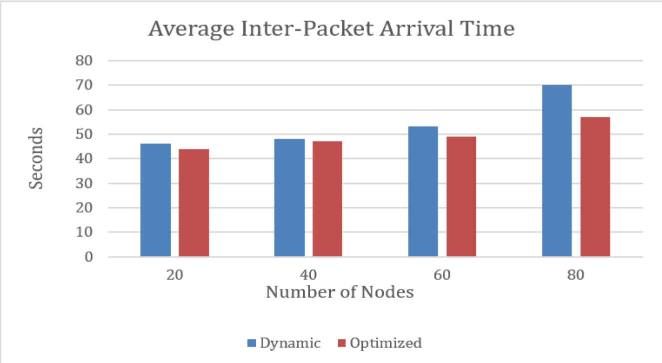

Fig. 4. Average Inter-Packet Arrival Time

Fig. 4 shows the average inter-packet arrival time comparison between these two approaches. It shows that the optimized approach over-performs the dynamic algorithm according this criterion. It is possible to see from this figure that inter-packet arrival time is less for the optimized approach.

VII. Conclusion and Future Work

In this paper, we propose an adjustment of the trickle time according to the drop-rate. The drop-rate is the best indicator of the network density and the propose surrounding factor around the network. If the network was dense, the dropping rate would be high, and thus, the timer should be reduced and vice versa. If there were a lot of obstacles in the network, then the dropping rate will be high, and thus we need to increase the listening time and vice versa. Thus, the improvement that we introduce in this work covers the dynamic condition around the network. This work achieves the best accuracy when compared with the other available enhancement on the RPL protocol.

As a future direction, the optimal dynamic trickle algorithms can be evaluated using different network topology, and more congested network condition. We are planning to evaluate it under different loss model algorithms (that represent different network environment) to prove its robustness. We also plan to include the packet dropping in the beacon data so that all nodes in the network will know all the dropping rate in the surrounding nodes. So that a better and accurate model can be constructed for the listening time calculation.

References

[1] M. C. Domingo, "An Overview of the Internet of Things for People with Disabilities," Journal of Network and Computer Applications, vol. 35, no. 2, pp. 584–596, 2012.

[2] W. Xiao, J. Liu, N. Jiang, and H. Shi, "An Optimization of the Object Function for Routing Protocol of Low-Power and Lossy Networks," in The 2014 2nd International Conference on Systems and Informatics (ICSAI 2014), pp. 515–519, 2014.

[3] H. Fotouhi, "Reliable Mobility Support in Low-Power Wireless Networks," Ph.D. dissertation, Faculdade de Engenharia, Universidade do Porto, 2015.

[4] M. Qasem, H. Altawssi, M. B. Yassien, and A. Al-Dubai, "Performance Evaluation of RPL Objective Functions," in 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, pp. 1606–1613, 2015.

[5] P. K. Singh, K. Kar, J. H. Nguyen, and D. Ku, "Mass Configuration with Confirmation in Tactical Networks," in Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications, pp. 99–106, 2017.

[6] L. Wallgren, S. Raza, and T. Voigt, "Routing Attacks and Countermeasures in the RPL-based Internet of Things," International Journal of Distributed Sensor Networks, vol. 9, no. 8, p. 794326, 2013.

[7] J. Hui and J. Vasseur, "The Routing Protocol for Low-Power and Lossy Networks (RPL) Option for Carrying RPL Information in Data-Plane Datagrams," RFC 6553, March, Tech. Rep., 2012.

[8] T. Winter, P. Thubert, A. Brandt, J. W. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J.-P. Vasseur, and R. K. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", rfc, vol. 6550, pp. 1–157, 2012.

[9] J. Tripathi, J. C. de Oliveira, and J.-P. Vasseur, "A Performance Evaluation Study of RPL: Routing Protocol for Low Power and Lossy Networks," in 2010 44th Annual Conference on Information Sciences and Systems (CISS), pp. 1–6, 2010.

[10] P. O. Kamgueu, E. Nataf, and T. D. Ndie, "Survey on RPL Enhancements: a Focus on Topology, Security and Mobility," Computer Communications, vol. 120, pp. 10–21, 2018.

[11] Y. B. Zikria, M. K. Afzal, F. Ishmanov, S. W. Kim, and H. Yu, "A Survey on Routing Protocols Supported by the Contiki Internet of Things Operating System," Future Generation Computer Systems, vol. 82, pp. 200–219, 2018.

[12] T. Clausen, J.-A. Cordero, J. Yi, and Y. Igarashi, "Use'em or Lose'em: On Unidirectional Links in Reactive Routing Protocols," Ad Hoc Networks, vol. 73, pp. 51–64, 2018.

[13] C. Pu and S. Hajjar, "Mitigating Forwarding Misbehaviors in RPL-based Low Power and Lossy Networks," in 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), pp. 1–6, 2018.

[14] D. Sasidharan and L. Jacob, "A Framework for the IPv6 Based Implementation of a Reactive Routing Protocol in ns-3: Case Study Using LOADng," Simulation Modelling Practice and Theory, vol. 82, pp. 32–54, 2018.

[15] M. B. Yassein, S. Aljawarneh, B. Ghaleb, E. Masa'deh, and R. M. T. Masadeh, "A New Dynamic Trickle Algorithm for Low Power and Lossy Networks," 2016 International Conference on Engineering & MIS (ICEMIS), pp. 1–6, 2016.

[16] V. Gupta, S. K. Devar, N. H. Kumar, and K. P. Bagadi, "Modelling of IoT Traffic and Its Impact on LoRaWAN," in

GLOBECOM 2017-2017 IEEE Global Communications Conference, pp. 1–6, 2017.
[17] Q. Shi, W. Shao, B. Fang, Y. Zhang, and Y. Zhang, "Reinforcement Learning-Based Spectrum Handoff Scheme with Measured PDR in Cognitive Radio Networks," Electronics Letters, vol. 55, no. 25, pp. 1368–1370, 2019.
[18] A. S. Shafigh, B. L. Veiga, and S. Glisic, "Cross Layer Scheme for Quality of Service Aware Multicast Routing in Mobile Ad Hoc Networks," Wireless Networks, vol. 24, no. 1, pp. 329–343, 2018.
[19] H. B. Liaqat, F. Xia, Q. Yang, Z. Xu, A. M. Ahmed, and A. Rahim, "Bio-inspired Packet Dropping for Ad-Hoc Social Networks," International Journal of Communication Systems, vol. 30, no. 1, p. e2857, 2017.
[20] M. A. Alsmirat, Y. Jararweh, I. Obaidat, and B. B. Gupta, "Automated Wireless Video Surveillance: an Evaluation Framework," Journal of Real Time Image Processing, vol. 13, no. 3, pp. 527–546, 2017.
[21] K. Balachander, V. Dionee, S. N. Suganya, and T. J. Christina, "A Viable IP Traceback VIA Dynamic Deterministic Packet Marking," International Journal of Engineering Science, vol. 5607, 2017.
[22] H. Aksu, L. Babun, M. Conti, G. Tolomei, and A. S. Uluagac, "Advertising in the IoT Era: Vision and Challenges," IEEE Communications Magazine, vol. 56, no. 11, pp. 138–144, 2018.
[23] P. Thubert, J.-P. Vasseur, E. M. Levy-Abegnoli, and P. Wetterwald, "Parent Device Allocation of Retransmit Slot to Child Network Device on Behalf of Peer Child Device in a Deterministic Network," uS Patent 9, 859, 970, Jan. 2018.
[24] M. Du, M. Zheng, and M. Song, "An Adaptive Preamble Sampling Based MAC Protocol for Cognitive Radio Sensor Networks," IEEE Sensors Letters, vol. 2, no. 1, pp. 1–4, 2018.
[25] X. Zhong and Y. Liang, "Scalable Downward Routing for Wireless Sensor Networks and Internet of Things Actuation," in 2018 IEEE 43rd Conference on Local Computer Networks (LCN), pp. 275–278, 2018.
[26] J. V. Sobral, J. J. Rodrigues, R. A. Rabelo, J. C. Lima Filho, N. Sousa, H. S. Araujo, and R. Holanda Filho, "A Framework for Enhancing the Performance of Internet of Things Applications Based on RFID and WSNs," Journal of Network and Computer Applications, vol. 107, pp. 56–68, 2018.
[27] M. Elappila, S. Chinara, and D. R. Parhi, "Survivable Path Routing in WSN for IoT Applications," Pervasive and Mobile Computing, vol. 43, pp. 49–63, 2018.
[28] B. Ghaleb, A. Al-Dubai, and E. Ekonomou, "E-trickle: Enhanced Trickle Algorithm for Low-Power and Lossy Networks," in 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, pp. 1123–1129, 2015.
[29] M. B. Yassein, S. Aljawarneh, and E. Masad'eh, "A New Elastic Trickle Timer Algorithm for Internet of Things," Journal of Network and Computer Applications, vol. 89, pp. 38–47, 2017.
[30] B. Ghaleb, A. Al-Dubai, E. Ekonomou, B. Paechter, and M. Qasem, "Trickle-plus: Elastic Trickle Algorithm for Low-Power Networks and Internet of Things," in 2016 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), pp. 103–108, 2016.
[31] C. Vallati and E. Mingozzi, "Trickle-f: Fair Broadcast Suppression to Improve Energy-Efficient Route Formation with the RPL Routing Protocol," in 2013 Sustainable Internet and ICT for Sustainability (SustainIT), pp. 1–9, 2013.
[32] B. Djamaa and M. Richardson, "Optimizing the Trickle Algorithm," IEEE Communications Letters, vol. 19, no. 5, pp. 819–822, 2015.
[33] T. M. Meyfroyt, S. C. Borst, O. J. Boxma, and D. Denteneer, "On the Scalability and Message Count of Trickle-Based Broadcasting Schemes," Queueing Systems, vol. 81, no. 2-3, pp. 203–230, 2015.
[34] M. Becker, K. Kuladinithi, and C. Gorg, "Modelling and Simulating the¨ Trickle Algorithm," in International Conference on Mobile Networks and Management, Springer, pp. 135–144, 2011.
[35] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks," in Proc. of the 1st USENIX/ACM Symp. on Networked Systems Design and Implementation, vol. 25, 2004.
[36] A. Mahmud, F. Hossain, T. A. Choity, and F. Juhin, "Simulation and Comparison of RPL, 6LoWPAN, and CoAP Protocols Using Cooja Simulator," in Proceedings of International Joint Conference on Computational Intelligence, Springer, pp. 317–326, 2020.
[37] M. Laaouafy, F. Lakrami, O. Labouidya, N. Elkamoun, and R. Iqdour, "Comparative Study of Localization Methods in WSN Using Cooja Simulator," in 2019 7th Mediterranean Congress of Telecommunications (CMT), pp. 1–5, 2019.