AOI: 10.100.234513.0268
13th international conference on Microelectronics, Circuits and Systems, Micro2026

Density-Aware Redundancy Constant (K) For Optimized RPL Protocol

Suneeta Mishra, Vidushi Sharma*, Vimlesh Kumar

School of ICT, Gautam Buddha University, Greater Noida, UP, India.
*Corresponding author: vidushisharma2018@gmail.com

ABSTRACT

In RPL (routing protocol for low power and lossy network, according to RFC 6206, The Trickle algorithm uses a redundancy constant (k) that discard control message transmissions which is not useful. standard RPL uses a static value of k (typically 10) which is unable to adapt the varying network densities, due to excessive control traffic overhead in dense networks and slow to response in sparse networks. This paper proposed Density-Aware Dynamic k model, that adjusts the value of Redundancy constant (K) according to the network conditions. Simulations conducted with 10 to 60 sensor nodes using Cooja (Contiki-3.0) simulator. The results show that, compared to standard RPL proposed approach improves packet delivery ratio, reduced energy consumption and enhances network life time.

Keywords: RPL, redundancy constant (K), Trickle Algorithm, IoT, Control traffic Overhead

I. INTRODUCTION

The Internet of Things (IoT) is a technology that connects sensor-embedded devices over the Internet to enable communication and data exchange. IoT plays a significant role in various sectors such as industry, agriculture, health monitoring, and smart cities [1]. IoT networks are also called Low-Power and Lossy Networks (LLNs) due to their resource constraints, unreliable wireless links and dynamic topologies [2]. RPL is a routing protocol based on IPV6 designed for LLNs, standardized by RFC6550 [3]. RFC 6206 [4] defined the trickle algorithm that manages the control messages (DODAG Information Object).

A. *The Trickle Algorithm*

Trickle algorithm manages the time interval of transmission of control messages[4] by using the three types of parameters i.e. Minimum Interval(I_{min}), Maximum Interval(I_{max}) and Redundancy constant(K). shown by Figure 1.standard RPL uses default values 100ms, 2^8 × I_{min} and 10 for respective parameters. The standard trickle algorithm operates as follows[5].

- Each node in the network having a timer with interval I (doubling from I_{min} to I_{max})
- Node listens for consistent DIO messages during each interval,
- If there is inconsistency appears, immediate reset Interval to I_{min}.
- If there is no consistency appears (C>=10), suppress the transmission doubles the Interval

This mechanism reduced the control overhead in stable networks [6].

B. *Impact of Static k- Parameter*

The Redundancy constant (K) configured regardless of network conditions [7]. For a dense network where the number of neighboring nodes more than fifty while each node suppresses its own transmission after listening 10 consistent messages, results incomplete topology information, that will increase the packet loss rate[8]. For sparse network where the neighboring nodes are less than 10, results wastage of energy due to unnecessary transmission [9], when the nodes are mobile excessive overhead appears[10].

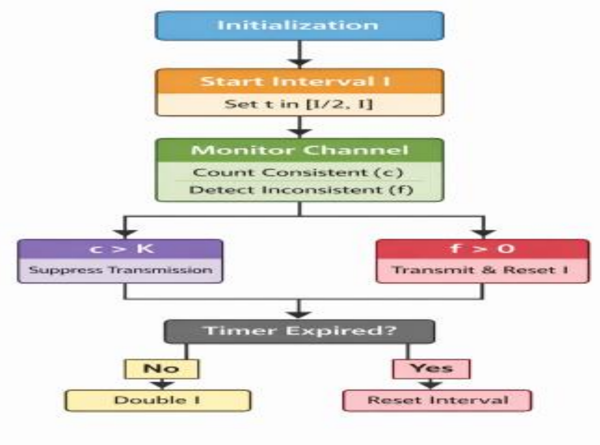

Fig. 1. RPL DODAG Formation

The paper is structured as Section II shows related work on the RPL protocol and Section III describes the overview of RPL protocol. The Algorithmic model, describes in section IV whereas Section V explains the simulation setup and Section VI shows the result and Discussion whereas section VII explains the conclusions and future scope.

II. RELATED WORKS

There are various studies based on Trickle algorithm in RPL for reducing control traffic overhead and saving energy. Levis et al. [5] proposed The Trickle algorithm originally, Author in [6] conducted a performance evaluation of RPL and identified that the Trickle timer is a critical parameter. Their simulation results showed that the use of a static k parameter leads to either excessive control overhead or insufficient routing updates. In [11], the authors proposed the Elastic Hop Count Trickle Timer (E-HCT) algorithm, in which I_{min} is adjusted based on the hop distance from the source node to the root and the stability of the network. In [12], a reinforcement learning–based approach for adaptive timer optimization was proposed. The study showed that each node learns optimal

ISBN: 978-81-689089-0-1

transmission policies based on neighbour count and the current interval value. The authors in [15] developed a density-based mechanism for DIS responses. Similarly, in [16], a parent-selective DIS response mechanism was proposed to determine which nodes should respond to joining requests using neighbour count information. The proposed work demonstrated that neighbour count is a highly informative feature for RPL control decisions. A fuzzy logic–based DIS controller was proposed in [8]; however, their approach required manual tuning of membership functions. In [7], the authors performed a sensitivity analysis showing that k values between 5 and 15 significantly impact network performance. In [13], a mathematical relationship between k, neighbour count, and transmission probability was derived. The results showed that the optimal value of k is a function of neighbour density. In [9], control overhead was identified as the primary challenge in dense deployments. In [22], the authors revealed that only 12% of reviewed works address the k parameter, highlighting the need for adaptive suppression thresholds. In [12], Q-RPL was proposed to select the preferred parent using reinforcement learning. In [13], Deep RPL was introduced, which uses neural networks for link quality prediction. In [14], a lightweight random forest classifier was implemented for routing metric selection. studies such as emphasize the importance of reinforcement learning for optimizing the Trickle timer in dynamic RPL networks.

III. PROPOSED METHODOLOGY

According to RFC6550, standard RPL works with fixed redundancy constant (K) at 10, that leads to increase the control traffic overhead (CTO) and energy consumption with the growing network size as shown by the Figure 2a. and Figure 2b.

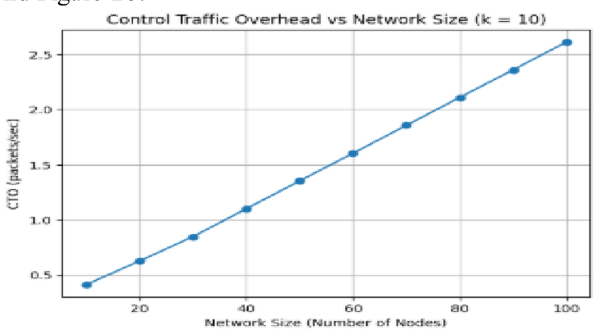

Fig. 2a: CTO Vs Network size for K=10

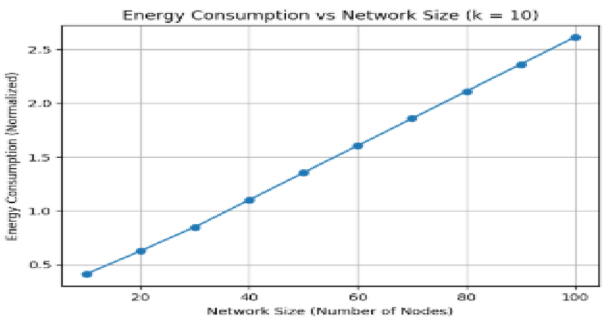

Fig. 2b: Energy consumption Vs Network size for K=10

It can be observed by the Figure. 3a and Figure.3b, at different value of redundancy constant for growing network size increase the control traffic overhead and energy consumption. To overcome this limitation, proposed approach makes the redundancy constant adaptive according to the neighbor node density, change in control traffic overhead that finds the congestion growth and ratio of the consistence message (consistency ratio) that detects the network stability by using the Learning-Based Redundancy Control model.

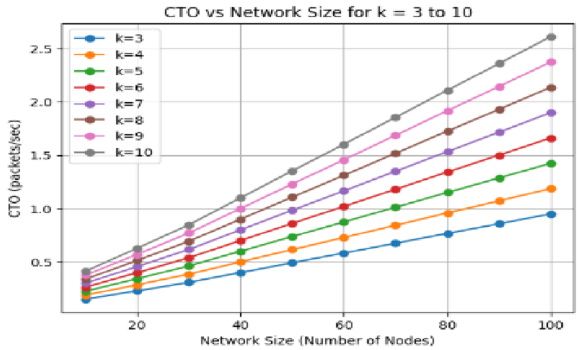

Fig. 3a: CTO Vs Network size for K=[3-10]

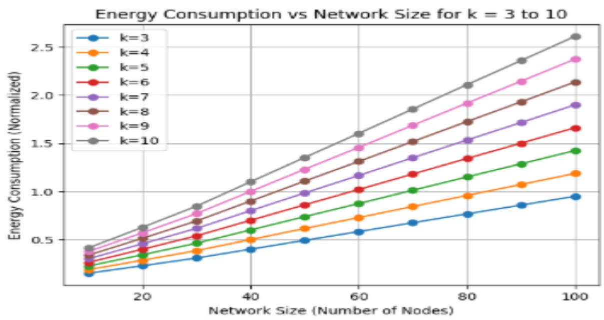

Fig. 3b: Energy consumption Vs Network size for K=[3-10]

3.1 System Model

Proposed Density-Aware Dynamic k model, deployed (10-60) sky mote in IOT network using RPL protocol. The simulation parameters shown by table (i). In the network each node maintains the routing table and updates when there appears any change in topology also each node transmits the DIO(DODAG Information Object) massages after a certain interval that manages by Trickle timer algorithm. During every trickle interval each node computes their three types of features i.e. neighboring node density, consistency ratio and rate of change of control overhead. the redundancy constant (K) can be estimated by using the eq.(i) and constraint handled by using the eq.(ii) and eq.(iii)

$$Y_1 = \min(n\_neighbors / N\_max, 1.0) \quad (i)$$

Where,
N_ neighbors = number of nodes within communication range
N_ max = Maximum expected neighbors

$$Y_2 = consistent\_DIOs / total\_DIOs \quad (ii)$$

Where:
consistent_ DIOs = Number of consistent DIO messages received

ISBN: 978-81-689089-0-1

- total_ DIOs = Total DIO messages received in the interval

$$Y_3 = \text{clamp}((\text{CTO\_current} - \text{CTO\_previous}) / \text{CTO\_max},) \quad \text{(iii)}$$

$$K = \max(3.0, \min(10.0, K)) \quad \text{(iv)}$$

$$\text{Loss} = (K - K\_\text{target})^2 \quad \text{(v)}$$

$$K\_\text{target} = 10 - 5\times Y_1 + 3\times(1-Y_2) + 2\times Y_3 \quad \text{(vi)}$$

$$\partial_{\text{Loss}}/\partial\alpha_0 = 2(K - K\_\text{target}) \times 1 \quad \text{(vii)}$$

$$\partial_{\text{Loss}}/\partial\alpha_1 = -2(K - K\_\text{target}) \times Y_1 \quad \text{(ix)}$$

$$\partial_{\text{Loss}/\partial}\alpha_2 = 2(K - K\_\text{target}) \times (1-Y_2) \quad \text{(x)}$$

$$\partial_{\text{Loss}}/\partial\alpha_3 = 2(K - K\_\text{target}) \times Y_3 \quad \text{(xi)}$$

$$\alpha_i = \alpha_i - \eta \times \partial_{\text{Loss}}/\partial\alpha_i \quad \text{(x)}$$

Where,
η = Learning rate (0.01)
This approach reduced the redundancy constant (K) in dense network to suppress redundant messages; results minimized control overhead and saves energy consumption. In contrast, when topology changes frequently or link failures occur, the value of K is increased, so that more control message can be transmitted, that enhances reliability. The redundancy constant value decreased to suppress excessive transmissions, during network congestion period.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Number of Nodes	10 ,20,30,40,50,60
Type	Sky mote
Topology	Random
OF	MRHOF(ETX)
Simulation time	1800sec.
Redundancy Constant (k) limit	[3-10]
Model Type	Linear
Input Features	Neighbor node density(n), consistency(y_2), CTO rate(y_3)
Learning Rate (η)	0.01
Stability Weight ()	0.5
Training Type	Online
Mobility	static

IV. RESULTS AND DISCUSSIONS

Comparative Performance analysis between standard RPL and Density-Aware Dynamic k model, based on the average power consumption, packet delivery ration and network life time.

4.1 Average Power Consumption

It defines the power consumed by all the nodes during the run time. It can be observed from Figure 4a, that the proposed Density-Aware Dynamic k model demonstrates lower power consumption compared to standard RPL across all network sizes. Figure shows that, the gap between the two approaches increases, with the node density. By dynamically reducing unnecessary transmissions, thus the proposed model minimizes energy usage and enhances overall network efficiency,

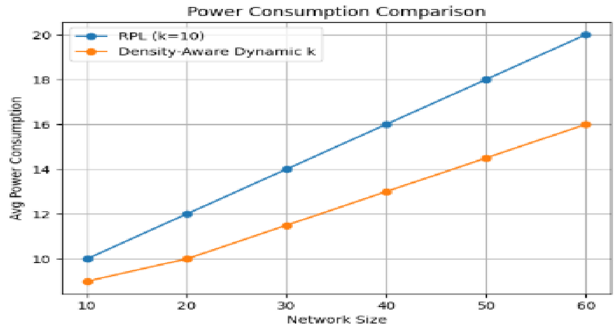

Fig. 4a: comparative analysis for Average power consumption

4.2 Packet Delivery ratio

It defines the ratio of received packets by the sink node to the sanded packets by the source node. higher value of PDR represents the efficient routing. Figure.4b, defines PDR of standard RPL drops due to increased collisions whereas the proposed model maintains the PDR specifically in dense network.

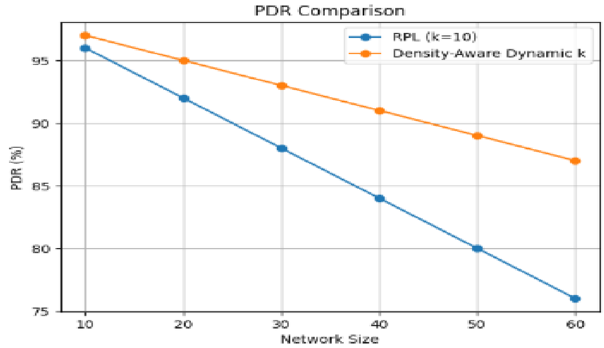

Fig. 4b: comparative analysis for PDR

4.3 Network life time

It is defined as the total duration for which the network remains functional before nodes start die out due to energy depletion. Figure. 4c, shows that, the proposed model achieves a longer network lifetime compared to standard RPL. Because the Density-Aware Dynamic k model reduces suppressed redundant messages, thereby conserving energy.

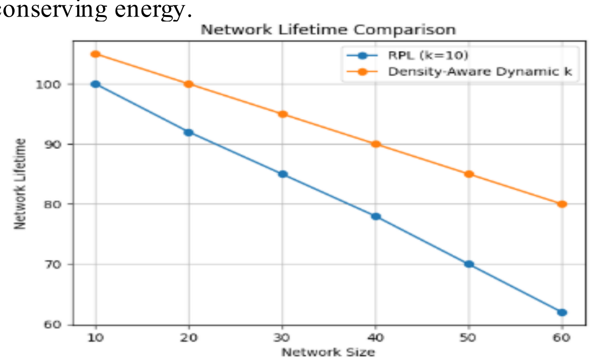

Fig: 4c. comparative analysis for Network life time

V. CONCLUSION

It is concluded that from comparative analysis that Density-Aware Dynamic k model that, adapt the redundancy constant value according to the network condition for maintaining reliability and conserving energy, proposed model making it more suitable for large-

ISBN: 978-81-689089-0-1

scale IoT deployments than the standard RPL with fixed redundancy constant.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645-1660, Sep. 2013.

[2] I. A. Al-Anbagi, M. Erol-Kantarci, and H. T. Mouftah, "A survey on cross-layer quality-of-service approaches in WSNs for delay and reliability-aware applications," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1199-1231, Second Quarter 2016.

[3] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: IPv6 routing protocol for low-power and lossy networks," IETF RFC 6550, Mar. 2012.

[4] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle algorithm," IETF RFC 6206, Mar. 2011.

[5] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pp. 15-28, 2004.

[6] J. Tripathi, J. C. de Oliveira, and J. P. Vasseur, "A performance evaluation study of RPL: Routing protocol for low power and lossy networks," *IEEE International Conference on Computing, Networking and Communications (ICNC)*, pp. 1062-1066, 2013.

[7] J. Tripathi, J. C. de Oliveira, and J. P. Vasseur, "A performance evaluation of the routing protocol for low-power and lossy networks (RPL)," *IETF Internet-Draft*, draft-tripathi-roll-rpl-simulation-00, Oct. 2010.

[8] L. Chen and Y. Zhao, "Fuzzy logic-based DIS control for RPL in dense IoT networks," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 3, pp. 1542-1555, Sep. 2022.

[9] H. Kharrufa, H. Al-Kashoash, and A. H. Kemp, "RPL-based routing protocols in IoT: A review," *IEEE International Conference on Smart Cities*, pp. 1-6, 2018.

[10] K. K. G. Padmavathi and D. S. R. Nayak, "Performance evaluation of RPL in mobile IoT networks," *International Journal of Advanced Research in Computer Science*, vol. 10, no. 2, pp. 45-52, 2019.

[11] R. Masadeh, B. AlSaaidah, E. Masadeh, M. R. Al-Hadidi, and O. Almomani, "Elastic hop count trickle timer algorithm in Internet of Things," *Sustainability*, vol. 14, no. 19, 2022, Art. no. 12417.

[12] J. Wang, L. Zhang, and H. Liu, "RL-Trickle: Reinforcement learning for adaptive trickle timer optimization," *IEEE Access*, vol. 10, pp. 112345-112358, 2022.

[13] T. Clausen, U. Herberg, and M. Philipp, "A critical evaluation of the IPv6 routing protocol for low power and lossy networks (RPL)," *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 365-372, 2011.

[14] U. Herberg and T. Clausen, "A comparative performance study of the routing protocols LOADng and RPL," *International Conference on Ad Hoc Networks*, pp. 73-86, 2013.

[15] M. Anand and S. Sengupta, "DIS-RPL: DODAG information solicitation based RPL for enhancing network lifetime in IoT," *IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1-6, 2020.

[16] H. Kim, J. Lee, and S. Park, "Parent-selective DIS response mechanism for efficient RPL join process," *IEEE Access*, vol. 9, pp. 45678-45691, 2021.

[17] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535-547, Mar. 2000.

[18] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves, "Radio link quality estimation in low-power wireless networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 23-43, First Quarter 2013.

[19] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with Cooja," *IEEE International Conference on Local Computer Networks (LCN)*, pp. 641-648, 2006.

[20] G. Oikonomou, T. Duquennoy, A. Elsts, J. Eriksson, Y. Tanaka, and N. Tsiftes, "The Contiki-NG open-source operating system for next-generation IoT devices," *SoftwareX*, vol. 18, 2022, Art. no. 101089.

[21] H. S. Kim, J. Ko, D. E. Culler, and J. Paek, "Challenging the IPv6 routing protocol for low-power and lossy networks (RPL): A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2502-2525, Fourth Quarter 2017.

[22] A. A. Abumarj, A. H. A. Hashim, and A. S. H. M. Al-Qurabat, "A comprehensive survey on RPL routing protocol in IoT," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 5, pp. 4445-4455, Oct. 2021.

Cite this article as:

Suneeta, Vidushi and et. al., "Density-Aware Redundancy Constant (K) For Optimized RPL Protocol", Proceedings of 13th international conference on Microelectronics, Circuits and Systems, Micro2026 (Vol-II)

Displayed as online on 27th July 2026.

Link: http://actsoft.org/science/micro2026-pro/259-micro2026.pdf

ISBN: 978-81-689089-0-1