Adaptive Parent Change and Cell Usage Aware Objective Function for RPL in 6TiSCH Network

Dzaky Zakiyal Fawwaz
Department of Information Convergence Engineering
Pusan National University,
Busan, South Korea
dzakybd@gmail.com

Sang-Hwa Chung*
Department of Information Convergence Engineering
Pusan National University,
Busan, South Korea
shchung@pusan.ac.kr

***Abstract*—The 6TiSCH wireless sensor network architecture combines the time-slotted channel hopping (TSCH) medium access layer (MAC) with the routing protocols for low-power and lossy networks (RPL). However, most RPL studies overlook the integration with TSCH MAC or focus solely on carrier-sense multiple access (CSMA) MAC. Existing RPL methods use an objective function (OF) based on an expected transmission count (ETX) link metric, yet this fails to fully capture network conditions and traffic load representation. Moreover, the RPL hysteresis method uses a static threshold without considering dynamic network conditions. Thus, it may not hold excessive parent changes and prevent better parent selection. To address these issues, we propose an adaptive parent change and cell usage aware objective function (AC) RPL method, which utilizes Q-learning to decide the optimal policy to change parent and an RPL OF based on traffic and cell usage. Experiments were conducted over simulation using the 6TiSCH simulator and real hardware tests using a testbed prepared by FIT IoT-Lab with OpenWSN firmware. The result demonstrates that AC-RPL outperforms the benchmarks. Compared to the standard RPL, AC-RPL can increase the packet delivery ratio and received packets by 9% and 13%. Also, it reduces energy consumption and latency by -21% and -8%.**

***Index Terms*—6TiSCH, RPL, Objective function, Wireless sensor network, Q-learning**

I. Introduction

The 6TiSCH working group is responsible for standardizing IPv6 protocols for low-power industrial networks by utilizing the TSCH mode of the IEEE802.15.4-2015 standard [1]. The TSCH MAC layer adopts a TDMA-based medium access and channel hopping approach, dividing time into uniform slots for packet exchanges between nodes and forming a repetitive pattern, known as a slotframe [2]. This working group facilitates low-power IoT applications, such as those used in smart industries and home appliances. 6TiSCH employs 6LoWPAN to facilitate IPv6 transmission over the network through header compression and packet fragmentation [3]. RPL manages the routing at the network layer, and a minimal scheduling function (MSF) is used for dynamic scheduling. MSF enhances the minimal scheduling setup and modifies child-parent links via the 6top protocol (6P) [4]. MSF utilizes a minimal cell for control packets and dedicated cells for data transmission. An example of a slotframe schedule can be seen in Figure 1.

* Corresponding Author

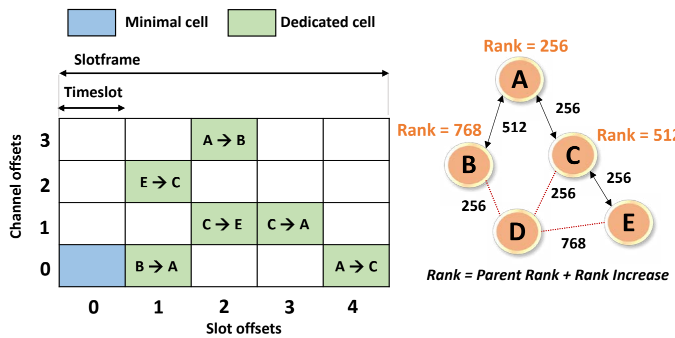


Fig. 1: Illustration of a TSCH schedule and RPL route in 6TiSCH network.

RPL establishes routes towards a border router [5]. The protocol uses DIO and DAO messages to create a downward and upward route. Nodes join the network by sending DIS and obtaining their DODAG rank using OF. OF0 [6] provides the mechanism for rank calculation and relies on hop count metric. Minimum Rank with Hysteresis OF (MRHOF) [7] enhance OF0 with hysteresis function and apply ETX metric, which represents the required number of re-transmission to send a packet. ETX metric depicted in Equation 1. MRHOF is designed to find the shortest path cost and prevent excessive parent churn in the network. Hysteresis function set a node to change path only if the rank difference exceeds a certain rank increment threshold following Equation 3.

The use of reinforcement learning (RL) algorithms in tasks like MAC scheduling, routing protocol development, and congestion control is progressively becoming more prevalent in IoT networking applications [8]. RL aims to maximize rewards through optimal state-action interactions involving an agent and its environment [9]. Q-learning (QL) algorithm [10] is a prominent RL method that operates off-policy. A Q-value represents the potential reward for a state-action pair, $Q(s, a)$, updated after each action based on a reward $r(s, a)$, expressed in Equation 4. QL utilizes learning, discount, and epsilon rate parameters, $\alpha, \beta, \varepsilon \in [0, 1]$. α balances the learning trade-off between existing knowledge and new values. β balances the importance of immediate versus future rewards. ε balances between exploration and exploitation of the learning based. In the exploitation phase, optimal action will be $a_{\text{opt}} = \text{argmax}_a$. In the exploration phase, any random a will be performed.

This study proposes AC-RPL, an adaptive parent change algorithm based on QL and cell usage aware objective function for RPL routing. AC-RPL aims to solve two problems. First, existing RPL routing is not designed specifically for TSCH in the 6TiSCH network, since MAC layer traffic information may help to improve routing quality. Second, the static threshold policy to change parent in MRHOF is not suitable for dynamic network conditions since we set the high threshold to a low traffic node, which may prevent the selection of a better parent from improving its quality; if we set the low threshold to a high traffic node may create unstable, excessive parent change. Therefore, AC-RPL considers MAC transmission load by cell usage along with queue load and packet transmission quality, which represent the network traffic and load condition. AC-RPL also formulated adaptive parent change so the node can proactively seek a better parent or prevent change to maintain stability.

$$\begin{aligned} \text{PRR} &= N_{Ack}/N_{Tx} \\ \text{ETX} &= 1/\text{PRR} \end{aligned} \tag{1}$$

$$\begin{aligned} &\text{Sp} = \lfloor (3 \times \text{Metric}) - 2 \rceil \\ &\text{rank}_{inc} = (\text{Rf} \times \text{Sp} + \text{Sr}) \times \text{MHRI} \\ &\text{rank}_i = \min\{\text{rank}_p + \text{rank}_{inc} \mid p \in \text{Pc}\} \end{aligned} \tag{2}$$

$$\text{Decision} = \begin{cases} \text{Change} & rank_{current} - rank_{new} \geq \text{threshold} \\ \text{Stay} & \text{Otherwise} \end{cases} \tag{3}$$

$$\begin{aligned} &\Delta Q(s,a) = r(s,a) + \beta \times \max_a Q(s_{next}, a) \\ &Q(s,a)' = (1-\alpha) \times Q(s,a) + \alpha \times \Delta Q(s,a) \end{aligned} \tag{4}$$

II. Related Work

ARMOR [11] utilizes a mobility-aware routing metric known as time to reside (TTR) along with a corresponding parent replacement strategy. TTR estimates the duration for which nodes will remain within each other's transmission range. TTR allows a node to choose a parent with a more prolonged connection time, enhancing reliability. PLC-OF [12] makes use of power line communication (PLC) physical (PHY) layer diversity as a routing metric, identifying the optimal path while taking sudden interference or congestion in power lines into account. EN-RPL [13] introduced a composite efficient routing (CER) metric that takes into account latency, queue utilization, node lifetime, link quality, and the number of bottleneck nodes, with weights that can be adjusted according to the needs of the application. ELITE [14] offers a routing metric called the Strobe Per Packet Ratio (SPR), which captures transmission operations at the MAC layer with the goal of selecting a route that minimizes strobe transmissions. FL-HELR-OF [15] presents an objective function that incorporates four metrics using fuzzy logic. The metrics are received signal strength indicator, hop count, latency and energy consumption. TA-RPL [16] leverages the cell allocation information from TSCH MAC, where the number of allocated TSCH cells reflects the bandwidth each node reserves and incorporates link quality and traffic information measured at the MAC layer. BEEX [17] proposes a multiarmed bandit (MAB)-based approach for predicting energy usage, enabling nodes to adapt and optimize total network energy consumption.

TABLE I: List of notations.

Notation	Description
Rf, Sp, Sr	Rank factor, Step of rank, Stretch of rank
SL	Slotframe length
MHRI	Minimal hop rank increase
aC	Autonomous cell
Pc, p	Parent candidates, Parent
mC	Minimal shared cell
nC^p_{used}	Negotiated cells used by node to parent
nC_{avail}	Negotiated cells available from parent
UC	Utilized cells from slotframe
nTx_p	Number of negotiated Tx cells to parent
nRx_p	Number of negotiated Rx cells to parent

However, none of the works consider adaptive parent changes and study RPL routing incorporated with TSCH MAC, except for TA-RPL. Thus, this study proposed AC-RPL that incorporated traffic load information from the TSCH MAC layer and formulated an adaptive parent change triggered using QL.

III. Proposed Method

Based on the earlier problems, we proposed AC-RPL as an adaptive parent change function using QL and cell usage-aware RPL OF. The related notations are listed in Table I.

A. Cell Usage aware Objective Function

We proposed RPL OF that considers negotiated cell usage that represents TSCH MAC traffic load, called cell available rate (CAR). CAR calculates from the reverse ratio of cell usage rate (CUR). CUR represents the number of negotiated cells used by the node to its parent nC^p_{used} with available negotiated cells from the candidate parent nC_{avail}, expressed in Equation 7. Furthermore, we also consider packet enqueue success rate (ESR), calculated from the reverse ratio of packet enqueue drop rate (EDR) because of the full load of the queue buffer or dropped by higher priority packet that described in Equation 5. We formulate ESR rather than queue load since it can represent the queue processing since the full queue buffer size was good as long as the node can maintain the transmission without any drop. Next, the packet reception rate (PRR) between the node and its parent indicates the number of packet transmission successes, shown in Equation 1. These metrics are combined as average reciprocal metrics following the calculation of ETX and then put into the step of the rank calculation shown in Equation 8. This OF has a broader route condition, starting from queue load condition, MAC layer traffic condition, and transmission link quality.

Node with low PRR indicates high re-transmission or higher failed transmission. Hence, we avoid selecting such connections and vice versa. ESR will promote the node to

Algorithm 1 AC-RPL

Input α, β, ε, nC_{avail}, Pc

function ParentCheck ▷ Periodic per N_{Tx} or N_{SL}
  Update parent Metric and rank_{inc}
  ▷ Learning end checkpoint
  Record SUR_{t+1}, s_{t+1}, p_{t+1}, Metric_{t+1}
  Update $r(s_t, a_t)$ and $Q(s_t, a_t)$
  Call $\mathrm{StartLearning}$
  Call $\mathrm{UpdateParent}$
function StartLearning ▷ Learning start checkpoint
  Record SUR_t, s_t, p_t, Metric_t
  if $rand < \varepsilon$ **then** random action a_t ▷ Exploration
  else action $a_t = a_t^{opt}$ ▷ Exploitation
function UpdateParent ▷ Update preferred parent
  if No parent or $\mathrm{Sp}_p > \mathrm{Sp}_{max}$ **then**
    Parent change ▷ Must change
  else if $p \neq top$ & $a_t = 1$ **then**
    Parent change ▷ Trigger change
  else Stay with current parent

Initialize empty Q-table
while Node is active **do**
  if First time joining **then** ▷ Start learning
    Call $\mathrm{StartLearning}$
  if Has parent and done N_{Tx} or N_{SL} **then** Call $\mathrm{ParentCheck}$

select the path with a loose queue buffer. Last, to illustrate CAR calculation based on Equation 7, let say we have $nC^p_{used} = 20$ with $nC_{avail} = 80$ from candidate parent 1 and we have $nC_{avail} = 40$ from candidate parent 2. So, CAR values are $1-(20/80) = 0.75$ and $1-(40/80) = 0.5$. Hence, the node will select candidate parent 1. The node reinforce to select a candidate parent that can handle the load while also providing more cells for future allocation. It will maximize bandwidth while also lowering traffic congestion.

$$\begin{aligned} \mathrm{EDR} &= Q_{drop}/Q_{enqueue} \\ \mathrm{ESR} &= 1 - \mathrm{EDR} \end{aligned} \tag{5}$$

$$\begin{aligned} \mathrm{UC} &= mC + aC + nC_{used} \\ nC_{avail} &= \mathrm{SL} - \mathrm{UC} \\ nC^p_{used} &= nTx_p + nRx_p \end{aligned} \tag{6}$$

$$\begin{aligned} \mathrm{CUR} &= \begin{cases} \dfrac{nC^p_{used}}{nC_{avail} + nC^p_{used}} & \text{Current } p \\ 1 & nC_{avail} \leq nC^p_{used} \\ \dfrac{nC^p_{used}}{nC_{avail}} & \text{Otherwise} \end{cases} \\ \mathrm{CAR} &= 1 - \mathrm{CUR} \end{aligned} \tag{7}$$

$$\mathrm{Metric} = \frac{(1/\mathrm{PRR}) + (1/\mathrm{ESR}) + (1/\mathrm{CAR})}{3} \tag{8}$$

B. Adaptive Parent Change using Q-learning

The proposed method utilizes Q-learning to enhance the performance of the RPL routing algorithm, with each node operating as an individual agent. We set the state s as a traffic class that categorizes traffic rate with size of $\mathrm{Tr}^{max}_{class}$, while the traffic rate itself calculated from the slotframe utilization rate, all depicts in Equation 9. Traffic class can represent how much transmission occurs to/from the node, which can show how many child or lower transmission depends on it. The higher class also represents higher traffic. The actions a are between staying with current parent $a = 0$ or allowing parent change $a = 1$. We want the node to select the optimal policy for each traffic class to define the tendency to actively seek a better parent or keep current conditions with its parent to maintain stability.

The reward is determined using Equation 10. When the node stays with the same parent ($p_{t+1} = p_t$), the reward would be highly positive if the parent maintains a stable or improved metric ($\mathrm{Metric}_{t+1} \leq \mathrm{Metric}_t$). Otherwise, the negative value is set if the metric is unstable or deteriorates ($\mathrm{Metric}_{t+1} > \mathrm{Metric}_t$). When the node changes with a new parent ($p_{t+1} \neq p_t$), the reward is subtracted by -1 to penalize the changing process. Changing the parent will result in more control packet exchange, leading to a higher energy charge and distracted transmission. This way, we want to promote nodes to change parents if the metric is stable or improved. We select metric as the comparison since it can represent parent and node connection quality, consisting their transmission quality, queue processing, and cell allocation. The higher metric value represents worse quality.

Within the exploration phase, AC-RPL will select random action on whether to change parent. In the exploitation phase, QL selects the optimal action for a specific state-action pair by opting for the highest Q-value, which signifies the cumulative reward over previous iterations. s_t, Metric_t, and a measurement triggered when the node first time joined the tree. Then, s_{t+1}, Metric_{t+1}, and reward updating triggered on the next periodic check and also for the following action. This periodic parent quality check is triggered after a certain number of transmissions N_{Tx} or slotframes N_{SL}. The proposed method is presented as a hysteresis function in Algorithm 1.

$$\begin{aligned} &\mathrm{SUR} = \mathrm{UC}/\mathrm{SL} \\ &s = min(\lfloor \mathrm{SUR} \times \mathrm{Tr}^{max}_{class} \rfloor, \mathrm{Tr}^{max}_{class} - 1) \\ &s \in \{0, 1, \ldots, \mathrm{Tr}^{max}_{class} - 1\} \end{aligned} \tag{9}$$

$$r(s,a) = \begin{cases} 2 & \mathrm{Metric}_{t+1} \leq \mathrm{Metric}_t \ \&\ p_{t+1} = p_t \\ -1 & \mathrm{Metric}_{t+1} > \mathrm{Metric}_t \ \&\ p_{t+1} = p_t \\ 1 & \mathrm{Metric}_{t+1} \leq \mathrm{Metric}_t \ \&\ p_{t+1} \neq p_t \\ -2 & \mathrm{Metric}_{t+1} > \mathrm{Metric}_t \ \&\ p_{t+1} \neq p_t \end{cases} \tag{10}$$

IV. Experiments

Experiments were carried out using the 6TiSCH simulator [18], a tool created by the 6TiSCH working group, programmed in Python. It employs the Pister-hack propagation

TABLE II: Experiment parameters.

Parameter	Value
Simulation tool	6TiSCH Simulator
Scheduling function	MSF
Queue size and Sp_{max}	10 and 9
Rf and Sr	1 and 0
SL and MHRI	101 and 256
Runs x Duration	3 x 60 minutes
Root position	Bottom-Left
α, β, ε	0.7, 0.7, 0.5
Packet interval	0.5 / 1 / 2 seconds
Topology (size)	5x5 (25) / 5x10 (50) / 10x10 (100)
Testbed settings	
Site	Lille, FIT IoT-Lab
Firmware	OpenWSN
Board	Arm Cortex M3
Topology (nodes)	5x5 (25)

TABLE III: Number of packet received in experiment.

Packet Interval	Network Size	Method	Packet Received
2	50	MRHOF	72702
		AC-RPL	74256
		BEEX	58996
		TA-RPL	70427
0.5	50	MRHOF	194099
		AC-RPL	228871
		BEEX	178892
		TA-RPL	207241
1	50	MRHOF	137769
		AC-RPL	146676
		BEEX	107371
		TA-RPL	134640
1	25	MRHOF	72352
		AC-RPL	73138
		BEEX	58883
		TA-RPL	68008
1	100	MRHOF	137594
		AC-RPL	191783
		BEEX	143314
		TA-RPL	137509

loss model to construct the node-link quality [19]. We configured data transmission based on an agricultural monitoring application [20] that set 20 bytes as the packet size and one second as the packet transmission interval. We doubled and halved the period into 0.5 and 2 seconds to mimic a range of packet transmission intervals. We set α, β, and ε with 0.7, 0.7, and 0.5, respectively. It focus more on exploitation phase while reinforcing learning new values and future rewards. We performed the scenario with 3 times runs to observe the variability in simulation results. The experiment parameters are listed in Table II.

We compared AC-RPL with several related benchmark algorithms: MHROF as the existing RPL OF standard, BEEX [17] due to its similar method of using RL for RPL routing, and TA-RPL [16] which also takes into account 6TiSCH cell usage to optimize RPL routing. Several evaluation metrics were used in this study. First, the packet delivery ratio depicts the rate of successful packet transmission. Second, the total packet received. Third, the latency shows the average delay in packet transmission from the source node to the root node and

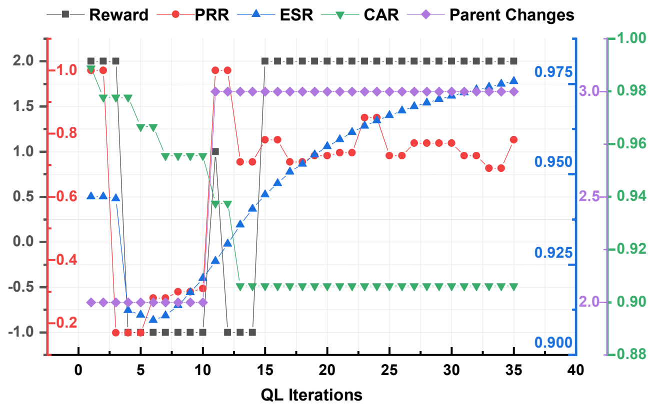


Fig. 2: Convergence evaluation on AC-RPL.

is expressed in seconds (s). Third, energy represents energy consumption expressed in milliampere per hour (mAh) and calculated based on the 6TiSCH energy model [21]. 6TiSCH set 2821.5 mAh as the node's battery capacity. Last, the total number of parent changes in the network, including the first time joining RPL.

A. *Evaluation on Q-learning convergence*

We monitored the variation in related variables of AC-RPL to evaluate their convergence across Q-learning iterations. We randomly chose one node for observation. Figure 2 shows that in the beginning, the reward is positive for a few moments but then negative, which depicts that it has worsened metric quality. It will promote the node to change the parent. Variable parent changes increment indicates the number of changes done by the node. After the change, the reward becomes stable. PRR and ESR were very low initially, indicating many failed transmissions occur and packet drops in the queue buffer. Later, when the node changes to the better parent, it can increase PRR and ESR. A higher rate of successful packet transmission will trigger MSF to request more negotiated cells to provide higher transmission requirements. Therefore, CAR also decreased after the node changed to a better parent, indicating the node has better link quality and allocated more cells to accommodate the traffic.

B. *Evaluation on benchmark algorithms*

Table III, Figure 3 and 4 show the evaluation result on different packet transmission intervals and network sizes. We can observe that lower intervals will have more traffic congestion, resulting in lower PDR and vice versa. The performance indicators deteriorate as the network size increases. There is barely any difference in the results when the network size is 25 nodes. The most notable difference was found at the largest network size of 100 nodes and the shortest packet transmission interval of 0.5 seconds. AC-RPL, in several scenarios, can have higher parent changes than other benchmarks.

In contrast, the other benchmarks can stick with their parent, which leads to performance stagnation. It indicates that AC-RPL can adaptively select parent change policy based on node traffic conditions and whether the change is stable

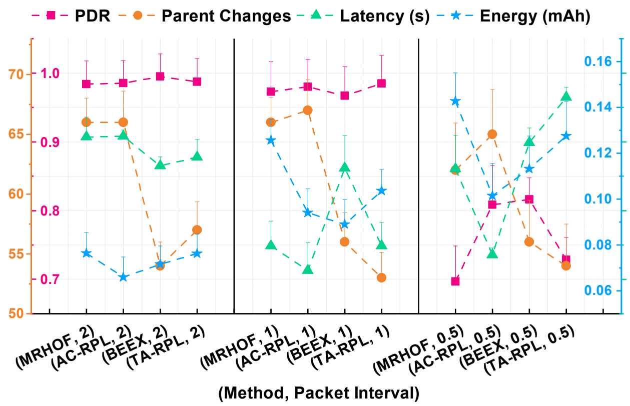


Fig. 3: Benchmark evaluation on different packet interval.

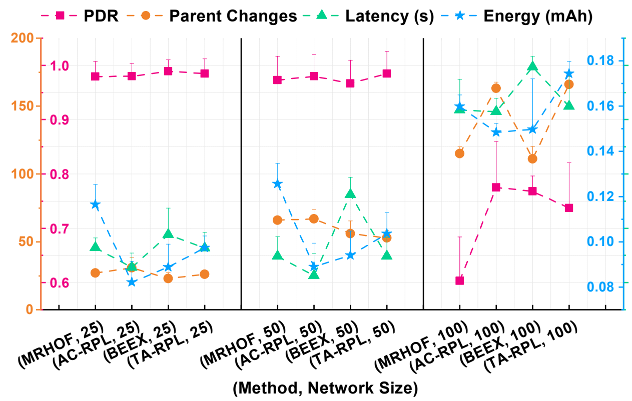
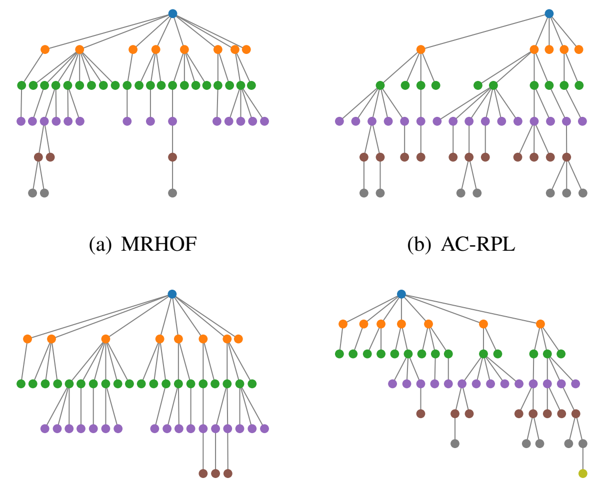


Fig. 4: Benchmark evaluation on different network size.

or unstable. Although more parent changes result in more control packet exchange to connect a node with a new parent, high changes do not mean it maintains lower performance. Otherwise, parent changes can give better performance. A better parent can prevent the network from maintaining traffic congestion or packet drops. In that case, the network can improve its packet reception, lower energy consumption and latency. Hence, AC-RPL maintains optimized parent selection based on the packet transmission quality, negotiated cell utilization representing MAC traffic, and queue load. Then, it can lower transmission failure, which is inherent in lower energy consumption needed for re-transmitting failed transmission and having more packet receptions. The other benchmarks result in lower performance since when the node has high traffic transmission and load that it cannot handle, it will result in congested traffic and higher latency. In comparison, AC-RPL can handle such conditions better since it considers the comprehensive metrics and has a proactive parent change that allows the node to seek a better parent.

The constructed topology of benchmark algorithms from 50 nodes network size is presented in Figure 5. The topology represents the RPL node connection. We discover that MRHOF and BEEX have dense connections upon the middle layers, though the distribution of child nodes also seems unbalanced. TA-RPL has quite a balanced topology, and it can distribute the load into several sub-trees but is quite dense in the middle layers and skewed in the lower layers. In comparison, AC-RPL has a more balanced topology than the other benchmarks by distributing the nodes into subhierarchies. By not being too deep, too wide or skewed, AC-RPL can achieve a level of resilience and manage fair cell utilization and queue allocation while minimizing congestion. AC-RPL reinforces better parent selection and results in a balanced connection since it considers network transmission, traffic, and load quality.



(a) MRHOF (b) AC-RPL

(c) BEEX (d) TA-RPL

Fig. 5: Benchmark network topology on 50 nodes.

C. *Evaluation on testbed environment*

We also test the proposed method AC-RPL with standard method MRHOF on 25 nodes over a real testbed on Lille, FIT IoT-LAB [22] that implemented using OpenWSN firmware [23]. The deployment site is shown in Figure 6. The parameters in the real testbed test were set as the same as the result of the previous experiment in the simulation. The evaluation result is shown in Figure 7. We discover that the real test validates our finding in simulation experiments that AC-RPL is better than MRHOF in terms of higher PDR, lower latency, and energy usage. Although it has more parent changes that also indicate more control packet exchange, those changes are necessary to prevent network traffic congestion from staying on bad parent quality and to improve network performance.

Fig. 6: FIT IoT-Lab testbed deployment in Lille site.

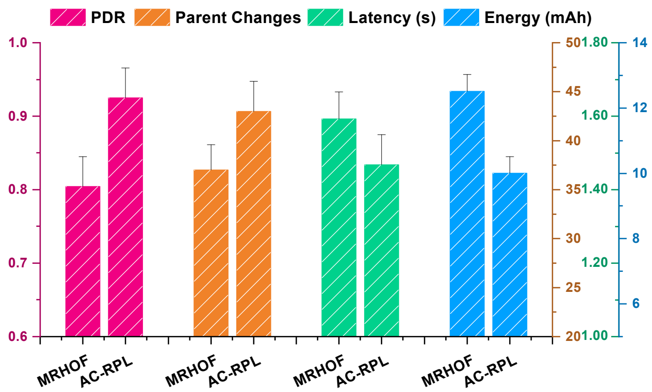


Fig. 7: Evaluation on testbed environment.

V. Conclusion

The challenge accompanied by existing RPL over a 6TiSCH network is not incorporating TSCH MAC layer information in the objective function to improve network routing. Furthermore, static parent change hysteresis or threshold restricts the node from seeking better parent conditions and also leading excessive changes, which can optimize overall network performance. Hence, the AC-RPL algorithm is proposed, which observes cell usage on TSCH, representing MAC traffic information along with packet transmission quality, queue load processing, and TSCH cell utilization. These metrics will give nodes a better decision policy in selecting parents. AC-RPL also formulates an adaptive parent change hysteresis function by considering the rank changes and step of rank that represents parent and node connection quality, which can reinforce the node to proactively change to a better parent. We evaluated AC-RPL using the 6TiSCH simulator and FIT IoT-Lab testbed on OpenWSN firmware with existing RPL routing algorithms. The results showed that AC-RPL could increase the packet delivery ratio and number of received packets while reducing latency and energy usage. In future work, we will optimize QL learning by using weight sharing to help newly joined nodes adapt faster and improve network performance.

Acknowledgement

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2023-RS-2023-00260098) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation).

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2023-RS-2023-00254177) grant funded by the Korea government(MSIT).

References

[1] P. Thubert, “An Architecture for IPv6 over the Time-Slotted Channel Hopping Mode of IEEE 802.15.4 (6TiSCH),” RFC 9030, May 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9030

[2] T. Watteyne, M. R. Palattella, and L. A. Grieco, “Using IEEE 802.15.4e Time-Slotted Channel Hopping (TSCH) in the Internet of Things (IoT): Problem Statement,” RFC 7554, May 2015. [Online]. Available: https://www.rfc-editor.org/info/rfc7554

[3] G. Montenegro, J. Hui, D. Culler, and N. Kushalnagar, “Transmission of IPv6 Packets over IEEE 802.15.4 Networks,” RFC 4944, Sep. 2007. [Online]. Available: https://www.rfc-editor.org/info/rfc4944

[4] T. Chang, M. Vučinić, X. Vilajosana, S. Duquennoy, and D. R. Dujovne, “6TiSCH Minimal Scheduling Function (MSF),” RFC 9033, May 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9033

[5] R. Alexander, A. Brandt, J. Vasseur, J. Hui, K. Pister, P. Thubert, P. Levis, R. Struik, R. Kelsey, and T. Winter, “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,” RFC 6550, Mar. 2012. [Online]. Available: https://www.rfc-editor.org/info/rfc6550

[6] P. Thubert, “Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL),” RFC 6552, Mar. 2012. [Online]. Available: https://www.rfc-editor.org/info/rfc6552

[7] O. Gnawali and P. Levis, “The Minimum Rank with Hysteresis Objective Function,” RFC 6719, Sep. 2012. [Online]. Available: https://www.rfc-editor.org/info/rfc6719

[8] M. S. Frikha, S. M. Gammar, A. Lahmadi, and L. Andrey, “Reinforcement and deep reinforcement learning for wireless internet of things: A survey,” *Computer Communications*, vol. 178, pp. 98–113, 2021.

[9] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[10] C. J. Watkins and P. Dayan, “Q-learning,” *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[11] A. Mohammadsalehi, B. Safaei, A. M. H. Monazzah, L. Bauer, J. Henkel, and A. Ejlali, “Armor: A reliable and mobility-aware rpl for mobile internet of things infrastructures,” *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1503–1516, 2021.

[12] M. Park, G. Jeong, O. Gnawali, and J. Paek, “Rpl objective function for multihop plc network,” *Journal of Communications and Networks*, vol. 25, no. 1, pp. 132–140, 2022.

[13] P. Singh and Y.-C. Chen, “Rpl enhancement for a parent selection mechanism and an efficient objective function,” *IEEE Sensors Journal*, vol. 19, no. 21, pp. 10 054–10 066, 2019.

[14] B. Safaei, A. M. H. Monazzah, and A. Ejlali, “Elite: An elaborated cross-layer rpl objective function to achieve energy efficiency in internet-of-things devices,” *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1169–1182, 2020.

[15] K. A. Darabkh, M. Al-Akhras, K. Ala’F, I. F. Jafar, and F. Jubair, “An innovative rpl objective function for broad range of iot domains utilizing fuzzy logic and multiple metrics,” *Expert Systems with Applications*, vol. 205, p. 117593, 2022.

[16] Y. Ha and S.-H. Chung, “Traffic-aware 6tisch routing method for iiot wireless networks,” *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 22 709–22 722, 2022.

[17] A. Musaddiq, R. Ali, S. W. Kim, and D.-S. Kim, “Learning-based resource management for low-power and lossy iot networks,” *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 16 006–16 016, 2022.

[18] E. Municio, G. Daneels, M. Vučinić, S. Latré, J. Famaey, Y. Tanaka, K. Brun, K. Muraoka, X. Vilajosana, and T. Watteyne, “Simulating 6tisch networks,” *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 3, p. e3494, 2019.

[19] H.-P. Le, M. John, and K. Pister, “Energy-aware routing in wireless sensor networks with adaptive energy-slope control,” *EE290Q-2 Spring*, pp. 1–6, 2009.

[20] K. S. Burman, S. Schmidt, D. El Houssaini, and O. Kanoun, “Design and evaluation of a low energy bluetooth sensor node for animal monitoring,” in *2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)*. IEEE, 2021, pp. 971–978.

[21] X. Vilajosana, Q. Wang, F. Chraim, T. Watteyne, T. Chang, and K. S. Pister, “A realistic energy consumption model for tsch networks,” *IEEE Sensors Journal*, vol. 14, no. 2, pp. 482–489, 2013.

[22] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele *et al.*, “Fit iot-lab: A large scale open experimental iot testbed,” in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE, 2015, pp. 459–464.

[23] T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, S. Glaser, and K. Pister, “Openwsn: a standards-based low-power wireless development environment,” *Transactions on Emerging Telecommunications Technologies*, vol. 23, no. 5, pp. 480–493, 2012.