Computers, Materials & Continua
DOI:10.32604/cmc.2021.014751
Article

Tech Science Press

Collision Observation-Based Optimization of Low-Power and Lossy IoT Network Using Reinforcement Learning

Arslan Musaddiq[1], Rashid Ali[2], Jin-Ghoo Choi[1], Byung-Seo Kim[3,*] and Sung-Won Kim[1]

[1]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan-si, 8541, South Korea
[2]School of Intelligent Mechatronics Engineering, Sejong University, Seoul, 05006, South Korea
[3]Department of Software and Communications Engineering, Hongik University, Seoul, 30016, South Korea
[*]Corresponding Author: Byung–Seo Kim. Email: jsnbs@hongik.ac.kr
Received: 13 October 2020; Accepted: 15 November 2020

Abstract: The Internet of Things (IoT) has numerous applications in every domain, e.g., smart cities to provide intelligent services to sustainable cities. The next-generation of IoT networks is expected to be densely deployed in a resource-constrained and lossy environment. The densely deployed nodes producing radically heterogeneous traffic pattern causes congestion and collision in the network. At the medium access control (MAC) layer, mitigating channel collision is still one of the main challenges of future IoT networks. Similarly, the standardized network layer uses a ranking mechanism based on hop-counts and expected transmission counts (ETX), which often does not adapt to the dynamic and lossy environment and impact performance. The ranking mechanism also requires large control overheads to update rank information. The resource-constrained IoT devices operating in a low-power and lossy network (LLN) environment need an efficient solution to handle these problems. Reinforcement learning (RL) algorithms like Q-learning are recently utilized to solve learning problems in LLNs devices like sensors. Thus, in this paper, an RL-based optimization of dense LLN IoT devices with heavy heterogeneous traffic is devised. The proposed protocol learns the collision information from the MAC layer and makes an intelligent decision at the network layer. The proposed protocol also enhances the operation of the trickle timer algorithm. A Q-learning model is employed to adaptively learn the channel collision probability and network layer ranking states with accumulated reward function. Based on a simulation using Contiki 3.0 Cooja, the proposed intelligent scheme achieves a lower packet loss ratio, improves throughput, produces lower control overheads, and consumes less energy than other state-of-the-art mechanisms.

Keywords: Internet of Things; RPL; MAC protocols; reinforcement learning; Q-learning

1 Introduction

Internet of Things (IoT) devices operate in a lossy environment and produce fluctuating and heterogeneous traffic patterns. The communication functions of these devices are based on the open systems interconnection (OSI) model layers. The device's communication capabilities can be enhanced by improving the lower OSI layers. The medium access control (MAC) and network layer mechanisms can be described with mathematical equations, but they require complicated algorithms due to device-constrained resources [1]. IoT nodes use carrier-sense multiple access with collision avoidance (CSMA/CA) to access a channel. The devices improve network efficiency by preventing collisions caused by nodes attempting to transmit at the same time. Collisions in the network represent one of the significant challenges of a wireless communication network. The collision probability increases as the network becomes denser [2]. The collision problem also worsens with a rise in the generation of network traffic.

The network layer uses the IPv6-based routing protocol for low-power and lossy networks (RPL) based on a destination-oriented directed acyclic (DODAG) graph. The RPL uses hop counts (i.e., objective function zero, OF0) [3] and the minimum rank with hysteresis objective function (MRHOF) [4], which determines an expected transmission count (ETX)-based link cost to route the packets. The RPL creates a ranking-based mechanism. These two OFs for rank calculations perform poor routing decisions. The ranks are updated using a control packet called a DODAG information object (DIO), which consumes computational and energy resources. The unbalanced traffic load, high relay burden, and limited communication capability eventually lead to poor performance. Therefore, designing an intelligent communication protocol for scarce resource devices is a challenging task [5].

Next-generation technologies need to focus on learning-based mechanisms to support numerous advanced fifth generation (5G) communication applications [6]. By utilizing intelligent protocols in low-power and lossy network (LLN) devices, the network can be more efficient and self-sustainable. To achieve this goal, the overall network performance can be enhanced by improving the device's capabilities at lower OSI model layers. The devices can learn the collision probability pattern at the MAC layer to perform efficiently and effectively at the network layer. The intelligent-learning models need to be extensively analyzed for efficient decision-making by the devices. Thus, machine learning (ML) is one of the most important artificial intelligence mechanisms for providing devices with the capabilities to learn and make decisions independently [7]. ML has recently attracted much attention due to its success in language processing, speech recognition, and big data analytics. Significantly, ML's features for cognitive radios have given wireless nodes the ability to recursively examine the wireless environment. With ML, IoT devices can develop an ability to learn a sequence of actions by utilizing data patterns [8]. Therefore, ML can enhance the device's performance by exploring future actions and learning from previous experiences. The integration of learning capabilities for a new generation of smart applications is essential for a self-sustainable optimized network. In this paper, we propose utilizing a reinforcement learning (RL)-based intelligent algorithm for LLN nodes. The proposed protocol uses a Q-learning mechanism to optimize the network. The proposed Q-learning-based protocol uses collision probability at the MAC layer and improves the network layer operation using this information. The proposed method guarantees a low packet loss ratio (PLR), high throughput, lower total control overheads, and lower energy consumption by utilizing collision information, node-ranking states, and a reward function.

In this paper, Section 2 explains the related research work, and Section 3 presents the preliminaries related to the machine learning-based RPL mechanism. The proposed protocol is defined

in Section 4, and Section 5 provides the performance evaluation results. Finally, Section 6 presents our conclusions and future directions.

2 Related Work

In recent years, several protocols have been introduced to enhance the performance and reliability of LLN devices. For example, at the network layer, queue-utilization-based RPL (QU-RPL) improves end-to-end delivery performance by balancing the routing tree through the utilization of queue information and the hop count as a routing metric [9]. Similarly, Ancillotti et al. [10] proposed a link quality estimation (LQE) strategy for the RPL. The LQE-based RPL uses a received-signal strength indicator (RSSI) and ETX metric to improve the RPL network link repair procedure. This method improves the RPL link quality but increases the control overhead. Tang et al. [11] proposed the (congestion avoidance) CA-RPL, which is a composite metric. The CA-RPL is based on the weight of each path along with ETX-based link quality estimation. This method also increases the control overhead. Another, the congestion-aware objective function (CoA-OF), uses ETX, QU, and the remaining energy metric for path selection [12]. This protocol enhances the packet delivery ratio (PDR), energy consumption, and throughput but introduces frequent parent changes that result in more control overhead. Taghizadeh et al. [13] also proposed utilizing the QU factor. Their proposed method reduces the energy consumption and packet loss ratio but with an increase in the control overhead. Ghaleb et al. [14] proposed a mechanism to improve load balancing by introducing a fast propagation timer to update the list of child nodes. In this way, it improves the packet reception ratio but also increases the network convergence time. The (stability-aware load-balancing) SL-RPL mechanism also improves load-balancing to enhance RPL network performance [15]. The SL-RPL routing metric is based on the packet transmission rate and ETX mechanism. The SL-RPL objective is to provide load balancing and to reduce frequent parent changing.

Similarly, a genetic-based algorithm uses a weighted queue length, hop counts, ETX, delay, and a residual energy metric [16]. This method enhances the end-to-end delay, the average success ratio of a packet, and the remaining energy. However, the control overhead is not improved in this mechanism. Additionally, Aziz et al. [17] presented a multi-armed bandit (MAB)-based clustering technique for improving the ETX-probing method. This technique is based on a clustering mechanism, and communicating with the cluster heads incurs additional overhead. An ETX and energy-based ranking method that uses fuzzy logic has also been proposed [18]. Though the fuzzy-based method improves network throughput, it also increases overall energy consumption. Similarly, an aggregation RPL scheme was proposed to modify the standardized RPL [19]. In this mechanism, the nodes communicate to the aggregator, which is connected to the fog node. The proposed scheme shows a high PDR and fewer hop counts and delays. However, the control overhead, which is one of the most important aspects of IoT communication, was not evaluated.

All these solutions cannot improve the RPL-based network control overhead, which affects the network performance and energy consumption. In next-generation networks, such as 5G and beyond, there must be a shift in focus from rule-based methods to learning-based protocols.

3 Preliminaries

3.1 Machine Learning for IoT-Based Systems

ML algorithms can be grouped into supervised learning, unsupervised learning, and RL. Supervised learning allows an agent to learn the input values in order to predict output values

for future use. In this way, the agent learns to perform actions for a new unseen dataset. An agent utilizing a supervised learning mechanism predicts unseen events by using classification and regression techniques. The agent uses classification to specify groupings (i.e., classes) of data elements. Similarly, a regression mechanism is used to find outputs that are real variables [20]. Various types of regression algorithms include linear regression [21], logistic regression [22], and polynomial regression [23]. Similarly, classification problems can be solved with techniques such as *k*-nearest neighbor (KNN) [24], linear classifiers [25], support vector machines (SVMs) [26], random forest [27], and Bayesian learning (BL). In unsupervised learning, no label data are given to the agent; the agent finds the hidden patterns solely from the unlabeled input data set. The objective is to create symmetries in the dataset to form groups. Unsupervised learning techniques include principal component analysis (PCA) [28], *k*-means clustering [29], and independent component analysis (ICA) [30]. The main objective of unsupervised ML algorithms is to decrease the features in the dataset. The ML mechanism models and applications for IoT networks are depicted in Fig. 1.

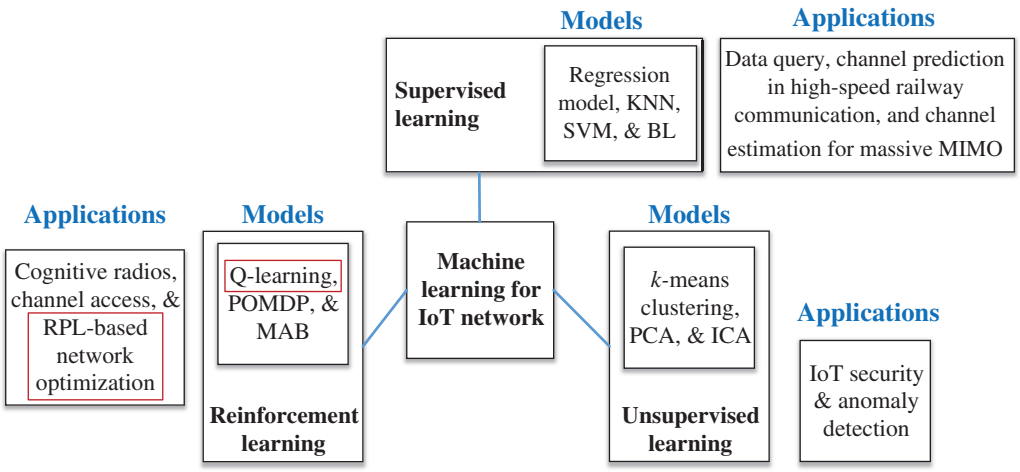

Figure 1: Machine learning for IoT networks

3.2 Reinforcement Learning

In RL, an agent learns the actions to maximize the expected sum of rewards. The agent learns the optimal actions to map the situations in an unknown environment. The states, actions, rewards, and state-transition probabilities describe the unknown environment. The unknown environment learning model could be a multi-armed bandit (MAB), Markov decision process (MDP), or partially observable Markov decision process (POMDP). If the environment has static states or has only one state, it is referred to as a stateless RL. These problems are solved with bandit techniques. In learning mechanisms, there should be an optimal exploration and exploitation. During exploration, it is possible to find an action that returns a poor reward, and it is also possible to become stuck in an optimal local action. The tasks are stochastic, and every action must be performed several times to obtain a reasonable estimation of future rewards [25].

The RL algorithm typically has four key sub-elements: the policy, reward, value function, and environment model (optional). The policy defines a method for a learning agent to behave in an environment. Similarly, with each action, there is an associated reward. The reward is a

numerical value, and the goal of an agent is to receive a maximum positive reward. The value function describes the long-term future reward for a given action. The reward might be small for the current action, but it could be a high-value function in the future. A model is useful in planning by evaluating possible future situations before they are experienced. RL problems could be either model-based or model-free (i.e., trial-and-error learners). Q-learning is a model-free RL mechanism to solve decision problems. The "Q" in Q-learning represents quality; it shows the quality or usefulness of the current action in obtaining a high reward. A Q-table is based on state and action values. The table is updated for each iteration or episode, and the device takes the next action based on the predicted state and action. To update a Q-value, the Q-learning algorithm utilizes the learning rate, discount factor, reward, and change in Q-value, represented as ΔQ [31]. The Q-learning technique has been used effectively to optimize cognitive radios [32] and the wireless channel access technique [33].

3.3 RPL Mechanism

The RPL routing protocol is based on a ranking mechanism. In a standardized version of RPL, the ranks are based on either OF0 (hop counts) or MRHOF (ETX). These two OFs for rank calculation perform poor routing decisions [5]. The RPL protocol uses control messages to construct a DODAG. These control messages are DIOs, destination advertisement objects (DAOs), DAO acknowledgments, and destination information solicitations (DISs). The objectives of these control messages and their transmission directions are depicted in Fig. 2a. The transmission sequence diagram of these control messages for the child and parent connection is shown in Fig. 2b. The exchange of these control messages consumes energy and valuable computational resources. Therefore, the optimization of control message transmission is vital for sensor network communication. The ranks are updated using DIO control messages. The frequency of DIO messages is adjusted using a trickle timer [34,35]. The trickle timer has a redundancy counter, redundancy coefficient, and interval length I. It picks a random transmission period from $[I/2; I]$ and increments the consistency counter according to the network consistencies. The DIO overhead is transmitted if the counter is less than the redundancy. The trickle timer algorithm exponentially increases the rate of DIO transmissions if the network is inconsistent and decreases the transmission rate to the initial value if the network is consistent. Inconsistent means there is a change in the rank value. Frequent DIO messages would cause delays, and the nodes would be consuming valuable computational resources.

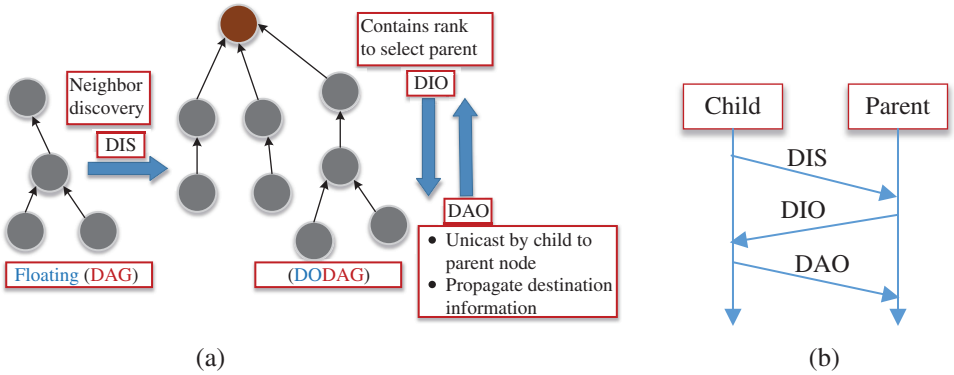

Figure 2: (a) RPL network control messages and (b) control message sequence diagram

4 Proposed Collision Observation-Based Optimization of Low-power and Lossy IoT Networks

4.1 Problem Formulation and System Model

IoT-based networks have wide-ranging application areas, such as smart, sustainable cities. Each node generates data based on its application and requirements. Therefore, the traffic pattern that is produced is unpredictable, which causes imbalanced load problems. The network layer defines the path and rules for the LLN node's efficient delivery of packets. In standardized protocols, the routing metric is based on OFs that perform poor routing decisions with high packet loss ratios (PLRs), overhead, and energy consumption.

In the proposed protocol, each node iteratively observes the environment to learn the collision information. The collision probability depends on the link (i.e., both the transmitter and receiver). The node uses the collision information at the network layer for rank formulation. Using collision information, the node generates ranks of potential parent nodes and, thus, automatically updates the routing table entries. We utilize the Q-learning mechanism to learn the state-action values to evaluate how desirable specific actions are in the current environment. The proposed framework is based on a DODAG graph. The graph is based on a parent–child network topology. Fig. 3 describes the Q-learning model and its elements for the proposed method. There are $N = PUC$ nodes in the network, where $P = (p_1, p_2, p_3, \ldots, p_i)$ are parent nodes, and $C = (c_1, c_2, c_3, \ldots, c_j)$ represents child nodes. In the proposed scheme, the agent is the LLN IoT device, and the environment is a wireless medium. The ranks of the node represent its states. Each node has a set of m states, $S = \{0, 1, 2, \ldots, m\}$. Each action represents a selection of a parent node for forwarding-path decisions. Each action generates a reinforcement signal or reward that describes whether the action is favorable or not. The reward is either positive or negative. The reward function is based on MAC protocol collision information at the parent node. Each node uses the CSMA/CA binary exponential backoff (BEB) method to access the channel for contention. The transmission frequency of each node is different; some nodes generate data packets with a high transmission rate, while others produce low traffic. Therefore, the overall traffic rate produced is heterogeneous with varying transmission rates. The link layer allows a maximum of eight re-transmission attempts, and the MAC layer maximum back off exponent is five.

4.2 Proposed Collision Observation-Based Optimization Mechanism

We explain our proposed collision observation-based learning mechanism for more efficient LLNs in this subsection. The proposed protocol is a Q-learning-based mechanism that iteratively finds the optimal state-action pair based on a reward system. We first replaced the ETX metric as a path cost with a less computationally expensive metric based on collision information in our proposed mechanism. Second, in the standardized protocol, the nodes create control overhead for each interval in announcing the rank status. Aide by the RL technique, we designed an algorithm to alleviate the need for a computationally expensive routing metric and a dependence on creating control overhead. The proposed Q-learning-based mechanism learns the node's collision status and utilizes it at the network layer to generate the routing table.

Consequently, it significantly reduces the control overhead. The transmission of data packets requires a large number of control overhead transmissions, and similarly, the transmitted data is often lost due to collisions in a dense networking environment. These problems can be addressed by integrating the learning capabilities in LLN nodes. Thus, Q-learning, which does not require large memory space and computational cost, is utilized efficiently. The Q-learning elements of the proposed mechanism are summarized in Tab. 1.

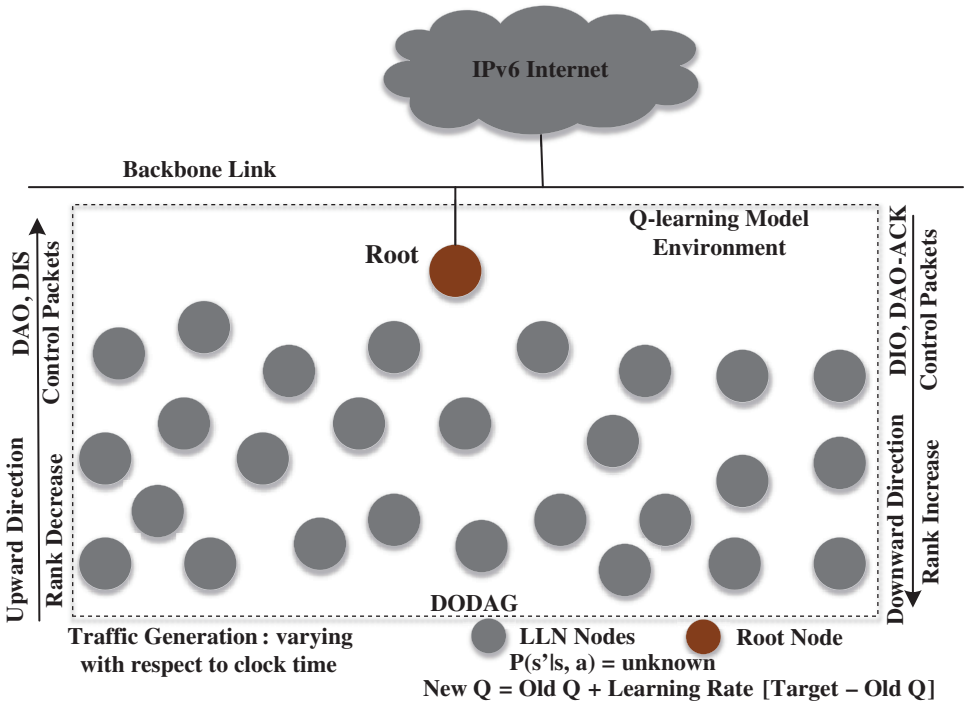

Figure 3: Q-learning model environment for LLN nodes

Table 1: Optimization of LLN nodes using the Q-learning model

Q-learning model	Optimization of LLN nodes
Agent	Sensor nodes
States	Node ranks
Action	Selection of parent node
Reward	Probability of collision
Objective	Minimize the collision probability

The CSMA/CA protocol uses a distributed coordination function (DCF) technique with BEB to avoid channel collision. In a wireless environment, collision happens when multiple transmissions occur simultaneously. The number of collisions is directly proportional to the network density. The collision information is derived from the information of the average contention window (CW) size and the number of nodes in its vicinity [36,37]. The information of nodes present in the vicinity is obtained using the IPv6 neighbor discovery function (RFC 4861) [38]. The backoff exponent (BE) increases with each collision. The value of the BE is within 0–5. The collision probability function uses BE stages and CW information during a given time slot. The CW information is obtained as follows:

$$CW = 2^{BE} - 1 \tag{1}$$

Using (1), the CW range is 0–31. The collision probability is calculated using the CW information as follows:

$$P_{coll} = 1 - (1 - 1/CW_{avg})^{K-1}, \tag{2}$$

where K represents the number of neighboring nodes, P_{coll} is the probability of collision, and CW_{avg} is the average CW of the transmission interval. The nodes utilize the collision information in the DIO message for the rank measurement as follows:

$$R(c_j) = h + R(p_i) + P_{coll}, \tag{3}$$

where $R(p_i)$ is the parent node rank, P_{coll} is the probability of collision at the receiver, and h represents a one-hop distance. It is an *RPL_min_hoprankinc* parameter defined in Section 6.1 of RFC 6719 [4]. If the value of *RPL_min_hoprankinc* is enormous, the advertised rank will be a simple hop count. RFC 6719 has carefully selected this parameter to balance the rank equation. Each node updates its rank using (3). Each node rank depends on its parent node's chosen rank value, path cost, and hop distance from the root node, as depicted in Fig. 4a. Fig. 4b provides the example scenarios in which a child node C has three potential parent nodes (i.e., p_0, p_1, p_2). As a result, the child node has three potential ranks (i.e., R_o, R_1, R_2). With the selection of any of the parent nodes for path forwarding, the child rank is updated accordingly. With each iteration, the child's rank either remains the same or changes according to the parent selection. Thus, we utilize rank value as a state in the Q-function. The child node in this example can transit between three states (i.e., s_0, s_1, s_2). With each action, the child node receives a reward in collision probability at a parent node. Using the state action and reward value, the node updates the Q-table. This state transition can also be expressed using the Markov decision process (MDP) mathematical framework (Fig. 5). The nodes interact with the environment in a sequence of time steps ($t = 0, 1, 2, 3, \ldots, n$). With each action, the agent transmits its state, referred to as a state-transition probability, $P(s'|s, a)$.

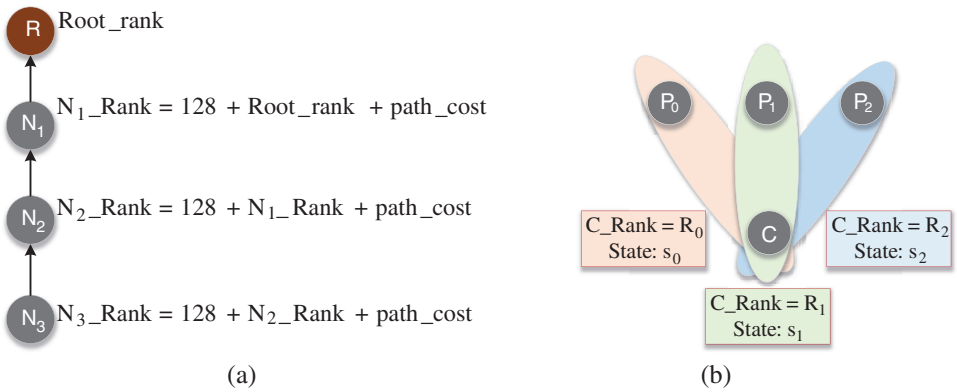

Figure 4: (a) RPL network rank update mechanism in DODAG and (b) example scenario depicting potential ranks and states of child node C

For Q-value generation, nodes use a Bellman equation that evaluates the optimal policy and value function, as shown in the following:

$$Q^{\pi *}(s_t, a_t) = \mathbb{E}\{r_t + \beta \times max_{a'} Q^{\pi *}(s', a') \,|s_t = s, a_t = a\}, \tag{4}$$

where β is a discount factor ($0 \leq \beta \leq 1$) that affects the weights given to future rewards, and r_t is the reward of the action a_t. The Q-learning evaluates the reward as the aggregated reward as follows:

$$Q(s_{t+1}, a_{t+1}) = (1-\alpha) \times Q(s_t, a_t) + \alpha \times \Delta Q(s_t, a_t) \tag{5}$$

where α is the learning rate ($0 \leq \alpha \leq 1$), representing the weights given to new values compared to the previous one. The $\Delta Q(s, a)$ value represents a learning estimate and is calculated as follows:

$$\Delta Q(s_t, a_t) = \left\{ r_t(s_t, a_t) + \beta \times min_a Q\left(s', a'\right) \right\}, \tag{6}$$

where $min_a Q\left(s', a'\right)$ is the best-estimated value of the state-action pair, and $r_t(s_t, a_t)$ indicates the obtained reward of state s and action a. This reward is described as follows:

$$reward \in \begin{cases} r^+, & if\ collision\ decreases \\ r^-, & otherwise \end{cases} \tag{7}$$

The reward indicates the action desirability. The reward r^+ is obtained if the collisions decrease at the parent node and vice versa. The collisions are measured using the information of average CW of a transmission interval (CW_{avg}) and the number of neighboring nodes. During the state transition of the proposed mechanism (Fig. 5), the node transits from one state to another with r^+ and r^- as rewards. The optimal Q-value can be found in a greedy manner. The policy for an epsilon-greedy (ε-greedy) action selection is to pick one of the actions with the best measured Q-value (i.e., exploitation) or select an action in a non-greedy manner (i.e., exploration) [39]. In the proposed mechanism, the exploration is performed using (3), and the exploitation phase uses the next state-action Q-values obtained from (5). We train the network during the exploration phase and then use the trained information to exploit the environment. In the proposed scenario, it is not efficient to choose a forwarding path randomly for training. It can have a routing-loop problem, which leads to a high packet loss ratio. During exploration, it is more efficient to select a forwarding path based on an RPL-based ranking mechanism that contains a ranking-based loop avoidance procedure. Thus, we utilize two equations for the exploration and exploitation phases, respectively.

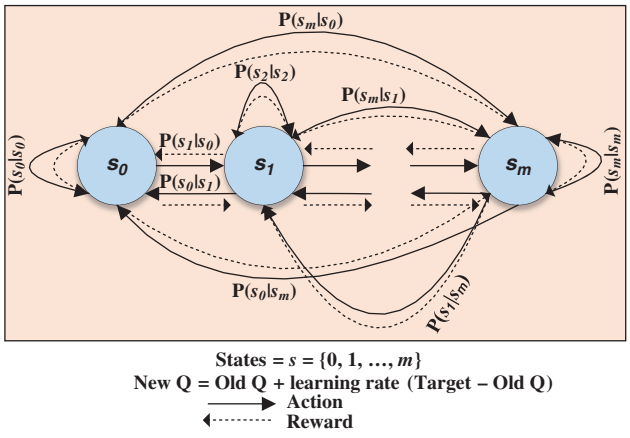

Figure 5: State-transition diagram of the proposed method

Limiting control overhead is one of the crucial requirements of IoT communication. The transmission of DIO control overhead is adjusted using a trickle timer mechanism. The LLN nodes are very limited in terms of their energy and computational resources. Therefore, it is highly preferable to limit the transmission of overhead packets without decreasing network performance. The DIO control packets carry the rank information of each node. Since the proposed learning mechanism does not require the rank information to perform the actions during exploitation, instead, it utilizes learned Q-values. Thus, the DIO transmissions are suppressed during the exploitation phase. The DIO transmissions continue again during the exploration phase. This suppression technique removes costly DIO packets during the exploitation phase and consequently significantly decreases the total percentage of network control overhead.

5 Performance Evaluation

The Contiki 3.0 Cooja simulator was used to evaluate the proposed protocol [40]. There was one root node and several client nodes. The network size was varied over a range of 20–100 client nodes. The simulation parameters utilized to evaluate the proposed protocol are listed in Tab. 2. The nodes were placed randomly, and the network was considered a varying traffic environment. These nodes were based on the Zolertia Z1 mote platform with a ROM size of 96 KB that supported 140 bytes of payload [41]. The nodes utilized MSP430 low-power microcontrollers. The simulation results of the proposed mechanism were compared with a standardized RPL (i.e., MRHOF and OF0), a queue utilization-based RPL (QU-RPL) [9], and an SL-RPL [15].

Table 2: Simulation parameters

Parameters	Value
Emulator	Contiki 3.0 Cooja
Packet size	127 bytes
Buffer occupancy	4 packets
Simulation time	3600 s
DAG size	20–100
PHY & MAC protocol	802.15.4 with CSMA
Packet size	127 bytes
uIP payload buffer size	140 bytes
Max back-off stages	5
Mote	Z1 Zolertia
CW_{min}	0
CW_{max}	31
RPL_min_hoprankinc (***h***)	128
Maximum retry limits	8
Packet transmission	Heterogeneous transmission
Script text analysis	Python 3.7

Fig. 6 describes network PLR for different-sized DAG networks in an LLN environment. The PLR is highly affected by congestion, collision, and other environmental factors. As shown in Fig. 6, the proposed protocol performed better compared to the other protocols. The performance of SL-RPL, QU-RPL, and MRHOF was also reliable when compared to OF0. The

OF0 used only hop counts to select a forwarding path. SL-RPL and QU-RPL used ETX with packet transmission rates and a queue-utilization factor, respectively. The proposed mechanism provided a low PLR, as shown in Fig. 6. The performance enhancement came from the learned collision information using the Q-learning technique. Network throughput in different-sized DAG networks is presented in Fig. 7. Throughput is related to PLR, and in the proposed protocol, one of the reasons for high throughput was a low PLR. The proposed mechanism measured the forwarding path according to the environment condition and thus led to better estimation than other protocols. The results of the PLR and throughput indicate that the collision metric had a significant impact on the network performance. OF0, which relied only on the hop-count metric, showed low throughput in the network environments, regardless of size.

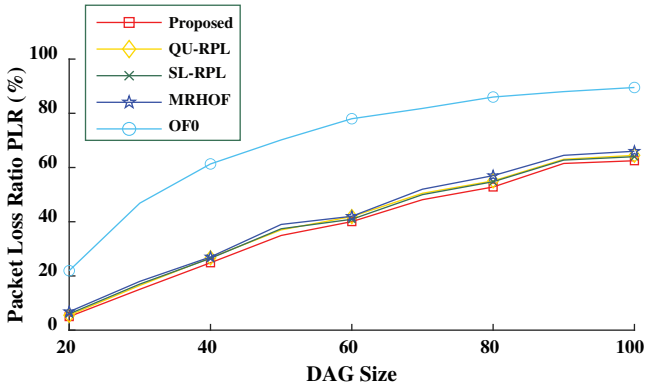

Figure 6: Network packet loss ratio (PLR; %) in different DAG sizes

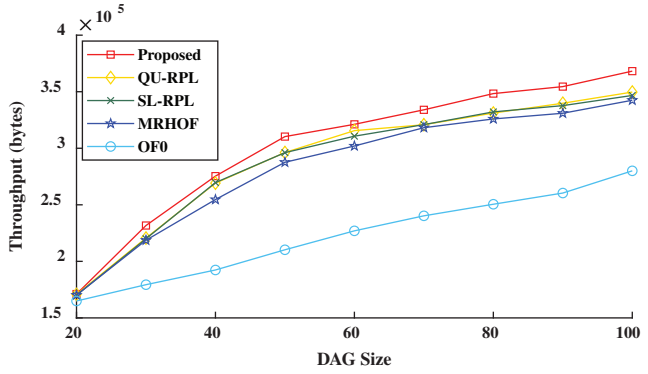

Figure 7: Network throughput (bytes) in different DAG sizes

Fig. 8 shows the total percentage of control overhead compared to the number of data packets. The proposed mechanism generated significantly less complete control overhead compared to other protocols. The RPL-based network incurred three types of control overhead (i.e., DIO, DAO, and DIS). The transmission of DIO packets increases if the network is inconsistent, and fewer DIOs are transmitted if the network is stable and consistent. In the Q-learning method, the nodes estimated the routing table entries using the Q-value and eliminated DIO control packet transmissions. In this way, the nodes incurred the lowest percentage of total control overhead

(7%–9%) while maintaining a low PLR. OF0 had the highest PLR and thus developed an unstable network that required more transmissions of control packets.

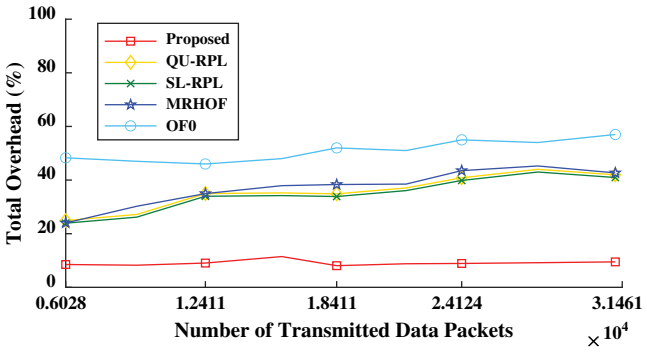

Figure 8: Total control overhead percentage with respect to different data packet transmissions

The energy consumption was directly related to control packet transmissions, which is evident in Fig. 9. Nodes consume energy during four modes of communication (i.e., low-power, CPU idle, transmission, and reception). The proposed mechanism's total energy consumption was the lowest due to the lower number of transmissions of control packets. Similarly, OF0 incurred the highest energy consumption due to its high PLR and high control overhead. SL-RPL, QU-RPL, and MRHOF consumed higher energy than the proposed method due to the CPU computation required to calculate the ETX measurements. The size of the data packets was more significant than the control overhead; thus, it required more energy to transmit the data packets. OF0 had very high PLR and control overhead, and the nodes consumed most of the energy in control overhead transmissions. The PLR and control overhead were lower in the proposed mechanism; the nodes spent most of the energy in data packet transmissions. The average end-to-end (E2E) packet deliver delays (in milliseconds) of different-sized DAG networks are shown in Fig. 10. The E2E delay was slightly higher in the proposed mechanism in comparison to QU-RPL, SL-RPL, and MRHOF due to the time spent in the learning process. However, the delay did not exceed that of OF0.

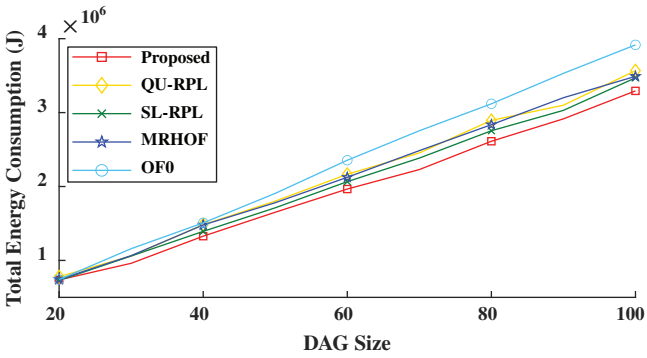

Figure 9: Total energy consumption in different-sized DAG networks

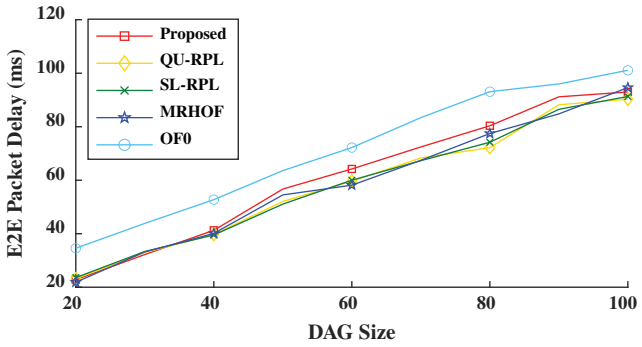

Figure 10: End-to-end (E2E) packet delivery delay in different-sized DAG networks

The simulation graphs show enhanced performance using the proposed mechanism. The improvement in the results shows the effectiveness of the RL mechanism. Using the RL-based approach, the control overhead can be reduced significantly. The improvement in PLR, throughput, control overhead, and energy consumption indicates that the proposed mechanism has potential in numerous IoT-based applications, such as smart city architecture.

6 Conclusion and Future Work

IoT networks for future generation protocols are expected to be densely deployed in a dynamic and lossy environment. The sensor's networks are highly constrained in their computational capabilities, energy, and memory consumption. Currently, IEEE 802.15.4 uses DCF with the BEB algorithm for channel access. Similarly, the network layer uses the RPL mechanism for routing decisions. Studies on optimizing sensor network performance, particularly in a lossy environment that is densely deployed with heterogeneous traffic applications, are still minimal. In such a densely deployed and uncertain wireless environment, high packet collision probability, congestion, and packet losses are still the key challenges. IoT devices can be optimized for next-generation networks by enhancing their communication capabilities at lower layers. Recently, RL-based protocols have shown promising applications and approaches to improve cognitive radios. Motivated by such applications, we proposed the utilization of an RL-based algorithm for LLNs. To handle the probability of collision caused by the increasing density of sensor networks, an RL technique (i.e., Q-learning method) is utilized. The proposed learning algorithm uses an intelligent-learning approach to optimize the LLN node performance using the RPL network ranking and MAC layer collision information. The proposed mechanism performs actions to select the forwarding path, and with each action, it updates the Q-table of state-action pairs to generate routing table entries. Contiki 3.0 Cooja simulation results indicated the optimized performance of the proposed protocol in an LLN environment. In the future, we plan to further optimize the network by improving the trickle timer mechanism using an RL-based approach.

Funding Statement: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government. (No. 2018R1A2B6002399).

Conflicts of Interest: The authors declare that they have no interest in reporting regarding the present study.

References

[1] A. Musaddiq, Y. B. Zikria, O. Hahm, H. Yu, A. K. Bashir *et al.,* "A survey on resource management in IoT operating systems," *IEEE Access*, vol. 6, pp. 8459–8482, 2018.

[2] N. Hajlaoui, I. Jabri and M. Ben Jemaa, "An accurate two dimensional Markov chain model for IEEE 802.11n DCF," *Wireless Networks*, vol. 24, no. 4, pp. 1019–1031, 2018.

[3] H. Altwassi, M. Qasem, M. Bani Yassein and A. Al-Dubai, "Performance evaluation of RPL objective functions," in *Proc. IEEE Int. Conf. on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable Autonomic and Secure Computing; Pervasive Intelligence and Computing*, Liverpool, UK, pp. 1606–1613, 2015.

[4] N. Pradeska, Widyawan, W. Najib and S. S. Kusumawardani, "Performance analysis of objective function MRHOF and OF0 in routing protocol RPL IPV6 over low power wireless personal area networks (6LoWPAN)," in *Proc. 8th Int. Conf. on Information Technology and Electrical Engineering*, Yogyakarta, Indonesia, pp. 1–6, 2016.

[5] A. Musaddiq, Y. B. Zikria, Zulqarnain and S. W. Kim, "Routing protocol for low-power and lossy networks for heterogeneous traffic network," *Journal on Wireless Communications and Networking*, vol. 2020, no. 1, pp. 1–23, 2020.

[6] D. P. Kumar, T. Amgoth and C. S. R. Annavarapu, "Machine learning algorithms for wireless sensor networks: A survey," *Information Fusion*, vol. 49, pp. 1–25, 2019.

[7] R. Boutaba, M. A. Salahuddin, N. Liman, S. Ayoubi, N. Shahriar *et al.,* "A comprehensive survey on machine learning for networking: Evolution applications and research opportunities," *Journal of Internet Services and Applications*, vol. 9, pp. 16, 2018.

[8] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi *et al.,* "Machine learning for Internet of Things data analysis: A survey," *Digital Communication and Networks*, vol. 4, no. 3, pp. 161–175, 2018.

[9] H. S. Kim, H. Kim, J. Paek and S. Bahk, "Load balancing under heavy traffic in RPL routing protocol for low power and lossy networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 4, pp. 964–979, 2017.

[10] E. Ancillotti, C. Vallati, R. Bruno and E. Mingozzi, "A reinforcement learning-based link quality estimation strategy for RPL and its impact on topology management," *Computer Communications*, vol. 112, no. 1, pp. 1–13, 2017.

[11] W. Tang, X. Ma, J. Huang and J. Wei, "Toward improved RPL: A congestion avoidance multipath routing protocol with time factor for wireless sensor networks," *Journal of Sensors*, vol. 2016, pp. 1–11, 2016.

[12] K. Bhandari, A. Hosen and G. Cho, "CoAR: Congestion-aware routing protocol for low power and lossy networks for IoT applications," *Sensors*, vol. 18, no. 11, pp. 3838, 2018.

[13] S. Taghizadeh, H. Bobarshad and H. Elbiaze, "CLRPL: Context-aware and load balancing RPL for IoT networks under heavy and highly dynamic load," *IEEE Access*, vol. 6, pp. 23277–23291, 2018.

[14] B. Ghaleb, A. Al-Dubai, E. Ekonomou, W. Gharib, L. Mackenzi *et al.,* "A new load-balancing aware objective function for RPL's IoT networks," in *Proc. IEEE 20th Int. Conf. on High Performance Computing and Communications; IEEE 16th Int. Conf. on Smart City; IEEE 4th Int. Conf. on Data Science and Systems (HPCC/SmartCity/DSS*, Exeter, United Kingdom, pp. 909–914, 2019.

[15] F. Wang, E. Babulak and Y. Tang, "SL-RPL: Stability-aware load balancing for RPL," *Transactions on Machine Learning and Data Mining*, vol. 13, no. 1, pp. 27–39, 2020.

[16] Y. Cao and M. Wu, "A novel RPL algorithm based on chaotic genetic algorithm," *Sensors*, vol. 18, no. 11, pp. 3647, 2018.

[17] M. Aziz, "On multi-armed bandits theory and applications," Ph.D. dissertation, Boston, MA, USA: Northeastern University, 2019.

[18] P. Fabian, A. Rachedi, C. Gueguen and S. Lohier, "Fuzzy-based objective function for routing protocol in the Internet of Things," in *Proc. of the IEEE Global Communications Conf.*, Abu Dhabi, UAE, pp. 1–6, 2018.

[19] R. J. Tom, S. Sankaranarayanan, V. H. C. de Albuquerque and J. J. P. C. Rodrigues, "Aggregator based RPL for an IoT-fog based power distribution system with 6LoWPAN," *China Communications*, vol. 17, no. 1, pp. 104–117, 2020.

[20] D. Bzdok, M. Krzywinski and N. Altman, "Points of significance: Machine learning: Supervised methods," *Nature Methods*, vol. 15, no. 1, pp. 5–6, 2018.

[21] B. Ramsundar and R. B. Zadeh, *TensorFlow for Deep Learning: From Linear Regression to Reinforcement Learning*. Sebastopol, CA, USA: O'Reilly Media, 2018.

[22] L. Liu, G. Luo, K. Qin and X. Zhang, "An algorithm based on logistic regression with data fusion in wireless sensor networks," *Journal on Wireless Communications and Networking*, vol. 2017, no. 1, pp. 10, 2017.

[23] K. Ohba, Y. Yoneda, K. Kurihara, T. Suganuma, H. Ito *et al.,* "Environmental data recovery using polynomial regression for large-scale wireless sensor networks," in *Proc. of the 5th Int. Conf. on Sensor Networks (SENSORNETS'16)*, Rome, Italy, pp. 161–168, 2016.

[24] Y. Han, K. Park, J. Hong, N. Ulamin and Y. K. Lee, "Distance-constrained k-nearest neighbor searching in mobile sensor networks," *Sensors*, vol. 15, no. 8, pp. 18209–18228, 2015.

[25] I. Jawhar, N. Mohamed, J. Al-Jaroodi, D. P. Agrawal and S. Zhang, "Communication and networking of UAV-based systems: Classification and associated architectures," *Journal of Network and Computer Applications*, vol. 84, pp. 93–108, 2017.

[26] Z. Dong, Y. Zhao and Z. Chen, "Support vector machine for channel prediction in high-speed railway communication systems," in *Proc. 2018 IEEE MTT-S Int. Wireless Sym.*, Chengdu, China, pp. 1–3, 2018.

[27] Z. Noshad, N. Javaid, T. Saba, Z. Wadud, M. Saleem *et al.,* "Fault detection in wireless sensor networks through the random forest classifier," *Sensors*, vol. 19, no. 7, pp. 1568, 2019.

[28] T. Yu, X. Wang and A. Shami, "Recursive principal component analysis-based data outlier detection and sensor data aggregation in IoT systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2207–2216, 2017.

[29] H. Harb, A. Makhoul and R. Couturier, "An enhanced K-means and ANOVA-based clustering approach for similarity aggregation in underwater wireless sensor networks," *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5483–5493, 2015.

[30] Z. Uddin, A. Ahmad, M. Iqbal and M. Naeem, "Applications of independent component analysis in wireless communication systems," *Wireless Personal Communications*, vol. 83, no. 4, pp. 2711–2737, 2015.

[31] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare and J. Pineau, "An introduction to deep reinforcement learning," *Foundation and Trends in Machine Learning*, vol. 11, no. 3–4, pp. 219–354, 2018.

[32] X. Li, J. Fang, W. Cheng, H. Duan, Z. Chen *et al.,* "Intelligent power control for spectrum sharing in cognitive radios: A deep reinforcement learning approach," *IEEE Access*, vol. 6, pp. 25463–25473, 2018.

[33] R. Ali, Y. A. Qadri, Y. B. Zikria, T. Umer, B. Kim *et al.,* "Q-learning-enabled channel access in next-generation dense wireless networks for IoT-based eHealth system," *Journal on Wireless Communications and Networking*, vol. 178, pp. 1–12, 2019.

[34] S. Goyal and T. Chand, "Improved trickle algorithm for routing protocol for low power and lossy networks," *IEEE Sensors Journal*, vol. 18, no. 5, pp. 2178–2183, 2018.

[35] A. Musaddiq, Y. B. Zikria and S. W. Kim, "Energy-aware adaptive trickle timer algorithm for RPL-based routing in the Internet of Things," in *28th Int. Telecommunication Networks and Applications Conf.*, NSW, Sydney, pp. 1–6, 2018.

[36] R. Ali, N. Shahin, A. Musaddiq, B. Kim and S. W. Kim, "Fair and efficient channel observation-based listen-before talk (CoLBT) for LAA-WiFi coexistence in unlicensed LTE," in *2018 Tenth Int. Conf. on Ubiquitous and Future Networks*, Prague, pp. 154–158, 2018.

[37] N. Shahin, R. Ali and Y. T. Kim, "Hybrid slotted-CSMA/CA-TDMA for efficient massive registration of IoT devices," *IEEE Access*, vol. 6, pp. 18366–18382, 2018.

[38] A. S. A. Mohamed Sid Ahmed, R. Hassan and N. E. Othman, "IPv6 Neighbor discovery protocol specifications, threats and countermeasures: A survey," *IEEE Access*, vol. 5, pp. 18187–18210, 2017.

[39] A. S. Mignona and R. L. A. Rochaa, "An adaptive implementation of of ϵ-greedy in reinforcement learning," *Procedia Computer Science*, vol. 109, pp. 1146–1151, 2017.
[40] Contiki, "Contiki: The open source operating system for the Internet of Things," 2015. [Online]. Available: http://www.contiki-os.org/.
[41] Zoleartia, "Z1 datasheet," 2010. Available: http://www.zolertia.com/.