IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.2024.Doi Number*

# A Reinforcement Learning based Intelligent Duty Cycle MAC Protocol for Internet of Things

**SHAH ABDUL LATIF[1], MICHEAL DRIEBERG[1], SOHAIL SARANG[2], AZRINA ABD AZIZ[1], RIZWAN AHMAD[3] AND GORAN M STOJANOVIĆ[2]**

[1]Department of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS, 32610 Seri Iskandar, Perak, Malaysia
[2]Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovića 6, 21000, Novi Sad, Serbia
[3]School of Electrical Engineering and Computer Science, National University of Sciences and Technology (*NUST*), Islamabad 44000, Pakistan

Corresponding author: Shah Abdul Latif (e-mail: abdul_22009852@utp.edu.my).

This research study was jointly supported by the graduate assistantship program at Universiti Teknologi PETRONAS and the GRETA project. GRETA has received funding from the European Union's Horizon Europe EIC 2023 Pathfinder Challenge Programme Grant 101161032..

**ABSTRACT** The Wireless Sensor Networks (WSNs) enabled Internet of Things (IoT) applications face energy efficiency challenge due to the limited battery capacity of the sensor nodes. Hence, the network's performance often involves a tradeoff with network lifetime. Traditional medium access control (MAC) protocols are less adaptable to the dynamic network conditions. While existing reinforcement learning (RL) based MACs are more adaptable, they still encounter challenges such as complexity and dimensionality. Therefore, this work aims to develop an RL based intelligent Duty cycle MAC (RiD-MAC) protocol that incorporates suitable network information to balance complexity and performance, effectively. The proposed RiD-MAC protocol is based on the Q-learning algorithm, meticulously designed with remaining energy as the state space and duty cycle as the action space. The reward is then formulated based on energy consumption and throughput. It is implemented on OMNeT++ platform-based Castalia simulator and the performance is compared with three state-of-the-art protocols, including AQSen-MAC, rlDC-MAC and QX-MAC under three simulation scenarios, stationary nodes with periodic traffic, hybrid traffic and node mobility. The simulation results demonstrate that RiD-MAC protocol significantly improves energy efficiency, with reduction in receiver energy consumption of up to 21%, and receiver energy consumption per bit of up to 26%, when compared to state-of-the-art protocols.

**INDEX TERMS** Machine learning, reinforcement learning, MAC protocol, intelligent duty cycle, internet of things.

## I. INTRODUCTION

The Internet of Things (IoT) forms a global infrastructure that connects living and non-living things such as humans, animals, electronic devices, vehicles, buildings and others [1], [2]. In recent years, tremendous growth has been observed in IoT applications areas including smart cities, agriculture, health, military, security, industrial automation, environmental monitoring and others [1], [3]. Furthermore, the number of connected IoT devices is predicted to reach approximately 29 billion globally by 2030 [1]. Wireless Sensor Networks (WSNs) are vital building blocks for IoT. A typical architecture of WSNs-enabled IoT is illustrated in Figure 1. WSNs can collect information from the surrounding environment, process and communicate it wirelessly [4]. It is an infrastructure-less wireless network that comprises of small sensor nodes. Each sensor node comprises of sensing element, processing element, communication element and battery. The challenge is that WSN nodes are solely powered from small non-rechargeable batteries with limited capacity [5]. The batteries can deplete within a brief period, which limits the lifetime of the sensor node. Additionally, sensor nodes are typically deployed in environments where battery replacements are costly and difficult [6]. Therefore, energy efficiency is crucial to prolong the functions of the sensor nodes.

In WSNs, the sensor nodes share the wireless medium to communicate and exchange data with each other. Only one sensor node can transmit data successfully through the medium at a time. Conversely, a collision occurs if two or more sensor nodes attempt to transmit data over the shared medium simultaneously. The medium access control (MAC)

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3606053



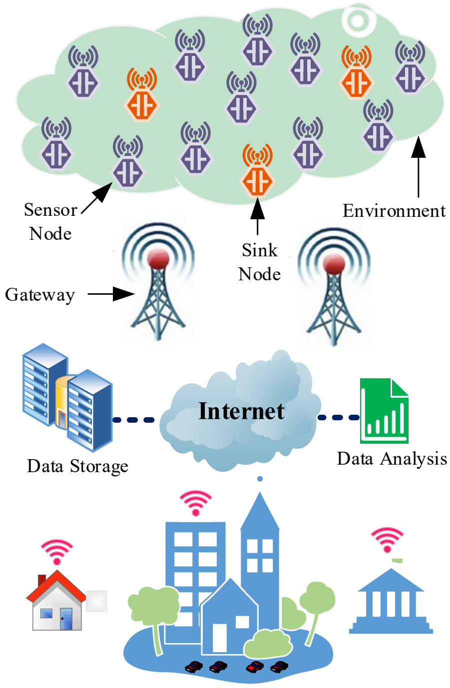


FIGURE 1. Architecture of WSNs-enabled IoT.

| **Nomenclature** | |
|---|---|
| ***Acronyms*** | |
| AQSen-MAC | Energy-efficient Asynchronous QoS MAC |
| CCA | Clear Channel Assessment |
| CI | Confidence Interval |
| CSMA | Carrier Sense Multiple Access |
| DC | Duty Cycle |
| IoT | Internet of Things |
| MAC | Medium Access Protocol |
| ML | Machine Learning |
| QX-MAC | Q-learning-based X-MAC |
| RiD-MAC | RL-based intelligent Duty cycle MAC |
| RL | Reinforcement Learning |
| rlDC-MAC | Reinforcement Learning based Duty Cycle MAC |
| SIFS | Short Interframe Space |
| WSN | Wireless Sensor Network |
| ***Symbols*** | |
| A | Action Space |
| D | End-to-end Packet Delay [s] |
| E | Receiver Energy Consumption per bit [J/bit] |
| $E_C$ | Energy Consumed [J] |
| R | Reward |
| S | State Space |
| T | Throughput [bps] |
| $T_{sleep}$ | Sleep Period [s] |
| α | Learning Rate |
| γ | Discount Factor |

protocol is employed to handle the access to the shared wireless medium during the transmitting and receiving of data packets [7]. It aims to minimize data collisions and retransmissions using time division, frequency division and carrier sense techniques. Additionally, many MAC protocols use collision avoidance mechanisms such as ready to send (RTS) or clear to send (CTS) and back off mechanisms [8].

Most of the energy consumption is contributed by transmission, reception and sensing operations [9]. Therefore, it is essential to focus on energy efficiency at MAC protocol to prolong network lifetime. Duty cycle (DC) techniques are employed to manage active period and sleep period to achieve energy efficiency [10]. Due to this, there is trade-off between maximizing network performance and extending network lifetime. The MAC protocol enables the sensor nodes to adjust their operations and use available energy efficiently to improve network performance while balancing network lifetime [11]. However, these sensor nodes operate in a dynamic network conditions that experiences regular variations in factors such as the number of nodes, topology, traffic load and other parameters [12]. Therefore, adaptive mechanisms are necessary for the MAC protocol to further enhance the network performance and extend its lifetime.

Machine learning (ML) offers adaptive techniques to manage dynamic network conditions [13], [14]. ML algorithms build models based on training experience and can adapt to the frequent changes in the dynamic network conditions. Supervised learning requires a large set of labelled data to predict the output accurately, while unsupervised learning can identify hidden patterns in unlabeled dataset [15]. On the other hand, reinforcement learning (RL) does not require prior data, but it is dependent on continuous interaction with the environment and feedback to make decisions [16]. An RL agent learns through trial and error based on its exploration and feedback from its environment [17]. Moreover, RL is simpler to implement and less complex than supervised and unsupervised methods [18]. For these reasons, RL is often preferred and employed in WSNs.

Q-learning is a well-known RL method that does not require prior knowledge of the environment. A Q-learning agent interacts with the environment, performs an action and receives positive or negative reward as feedback, depending on the action's impact on the environment [19]. Compared to other RL algorithms, Q-learning achieves fast convergence under given conditions. Additionally, it conserves computational resources due to its low power consumption. Furthermore, the Q-learning algorithm's structure is simple to implement [20]. Therefore, the Q-learning algorithm is chosen for this study.

### A. MOTIVATION

Prior to this, several types of MAC protocols have been developed, from traditional MAC protocols [21]-[26] to ML-based MAC protocols [27]-[35]. AQSen-MAC [25] employs a fixed duty cycle calculation formula which limits its performance under dynamic network conditions. Additionally, the nodes starts with an initial energy of 75% only instead of full capacity, which is not realistic. rlDC-MAC [34] uses RL but considers three parameters in the state space and two parameters in the action space, which lead to exponential growth in the Q-value table dimensions. This increases complexity and resource usage, and may also cause frequent changes in the state, leading to delayed convergence. Additionally, it employs a complex reward function that makes the tuning of weight factors difficult. QX-MAC [30] chooses only queue length as the state space. Although simpler to implement, this choice may cause faster energy depletion due to not accounting for the energy cost of its actions. Additionally, a binary reward function has been employed, which may reduce adaptability to dynamic network conditions and limit the ability to capture critical network information. Therefore, there is a need for a protocol that is adaptable to dynamic network conditions, has moderate complexity and employs appropriate network parameters for RL algorithm design and learning formulation.

Thus, the focus of this work is the development of a RL-based intelligent Duty cycle MAC (RiD-MAC) protocol for WSNs-enabled IoT. The proposed RiD-MAC protocol uses Q-learning to adjust the duty cycle of the receiver node based on its remaining energy as the state space and duty cycle as the action space. The protocol aims to balance complexity and performance effectively by employing appropriate and sufficient network information. The RiD-MAC protocol ensures robust performance under dynamic network conditions, including three simulation scenarios, stationary nodes with periodic traffic, hybrid traffic and node mobility, thereby providing realistic adaptation of duty cycle adjustment.

### B. MAIN CONTRIBUTIONS

The main contributions of this research work are as follows:

- The proposed Q-learning based intelligent duty cycle MAC protocol adjusts the sleep period of the receiver node with respect to remaining energy.
- The receiver node wakes up periodically to receive data packets from the intended senders. The receiver node remains active for longer periods of time when it has high energy and sleeps more when it has low energy.
- The Q-learning algorithm is designed with remaining energy as the state space and duty cycle as the action space. The reward is formulated based on energy consumption and throughput.
- The proposed RiD-MAC protocol employs epsilon-greedy (ε-greedy) policy to address the exploration-exploitation dilemma in RL.
- The proposed RiD-MAC protocol is implemented and evaluated at the packet level on OMNeT++ platform-based Castalia simulator.
- The performance of the proposed RiD-MAC protocol is compared with state-of-the-art protocols, including AQSen-MAC [25], QX-MAC [30] and rlDC-MAC [34] under dynamic network conditions.
- The RiD-MAC protocol substantially reduces the receiver energy consumption by up to 21%, and receiver energy consumption per bit by up to 26%, when compared to state-of-the-art protocols.

### C. ORGANIZATION

The rest of this paper is organized as follows: Section II presents the literature review. Section III describes the machine learning techniques for WSNs-enabled IoT. Section IV outlines the RiD-MAC protocol design. Section V presents the results and discussion, and Section VI concludes the paper and provides the future work.

## II. LITERATURE REVIEW

The proposed MAC protocol in [21] adjusts its duty cycle based on traffic load by creating clusters. The protocol aims to reduce energy consumption while improving latency and packet collisions. However, border nodes can follow more than one schedule which increases the number of virtual clusters. Hence, energy consumption increases as more nodes remain active due to the large number of virtual clusters. The MAC protocol in [22] overcomes this issue by employing predefined clusters. The duty cycle is adjusted with respect to the clusters to vary their sleep schedules. However, the protocol forces nodes to follow a fixed duty cycle for each cluster, which may result in less adaptation to dynamic network conditions.

The hybrid protocol in [23] adjusts duty cycle based on residual energy and traffic load. It combines synchronous and asynchronous mechanisms, using the asynchronous mechanism to adjust the duty cycle and the synchronous mechanism to share the duty cycle with neighboring nodes. Results show that the protocol improves network lifetime, packet delivery ratio and delay. However, the protocol increases network complexity due to its hybrid mode. Additionally, it increases synchronization overhead. Furthermore, the protocol does not account for adaptation to dynamic network conditions.

The MAC protocol proposed in [24] aims to decrease energy consumption and delay. The proposed protocol considers traffic load to adjust the duty cycle of the receiver node, and the contention window of the sender node. The protocol reduces sleep time during high traffic and reduces wake up time during low traffic. Early acknowledgment is used to broadcast the information of duty cycle ratio and contention window to decrease energy consumption and delay. However, the protocol has increased complexity due to combining the adjustment of duty cycle and contention window size. Additionally, it also did not consider changes in dynamic network conditions.

The authors in [25] proposed the energy-efficient asynchronous QoS (AQSen) MAC protocol. The AQSen-MAC achieves good energy efficiency and network performance. The duty cycle of the receiver node is adjusted based on its remaining energy to improve energy efficiency. The protocol reduces the receiver energy consumption significantly. However, the results are obtained by setting initial energy up to 75% only instead of full capacity. Additionally, duty cycle adjustment is done by using a fixed formula which does not consider the dynamic aspect of WSNs.

The authors in [26] present an energy efficient and QoS focused protocol called EEQ MAC. The EEQ MAC adjusts nodes' duty cycle based on queue length and data priority. In contrast to [25], the EEQ MAC adjusts the duty cycle of receiver and sender node. Hence, it increases complexity, communication overhead, and synchronization issues. Additionally, the protocol does not consider energy aspect for the duty cycle adjustment, and dynamic aspects of WSNs.

The authors in [27] propose a dynamic clustering technique based on adaptive sleep scheduling. The sensor nodes are arranged into clusters and a cluster head is assigned in each cluster. The nodes join the cluster dynamically. Packet arrival time is considered to adjust sleep scheduling. The algorithm operates in an iterative manner with each iteration consisting of foundation, formation, and forwarding. At the start of each slot, a node can select from a set of three actions: transmit, sleep and listen. If transmit is selected, then the time slot to send packet is obtained. The algorithm uses separate Q-learning mechanisms for action selection and time slot determination. Hence it increases complexity and computational overheads. Furthermore, the algorithm may experience convergence issues since each node has to update and store the Q-value table.

The authors in [28] proposed an RL based sleep scheduling for energy efficiency and lifetime of WSNs. A selective number of nodes remain active while all other nodes sleep. The Q-learning is employed to schedule the sleep period for the nodes based on variations in remaining energy. The state space is considered as the set of all the nodes whereas action space is designed as the set of all neighboring nodes for a given a node. This may result in an infinite number of state-action pairs which effectively increases the dimension of the Q-table and complexity for larger networks. Additionally, even distribution of energy increases overhead and may compromise performance by ignoring shorter communication routes.

The authors in [29] developed a multi-tier learning based on RL and a multi-armed bandit (MAB) model for slot selection and sleep scheduling in WSNs. The authors strive to balance the trade-off between throughput and energy efficiency. Simulations were performed to validate the analytical model developed for sleep scheduling. Adaptive decision making is achieved by using RL and the MAB model in tandem. However, this combination adds communication overhead and increases complexity. Additionally, convergence may be affected due to the use of the two models on each node, which causes frequent transients in decision making.

The active period of the sender nodes is adjusted using Q-learning in the QX-MAC protocol proposed in [30]. The sender node reserves the active period based on the number of packets queued for transmission in the current state. A new state is achieved once this active period is over. If the queue is empty in the new state, a positive reward is given, otherwise a negative reward is given. On the other hand, the receiver node is notified about the queue status of the sender node by turning the "more bit" to 1 when the sender has more packets lined up for the same receiver and 0 otherwise. However, the dynamic network conditions may not be fully captured using queue length as a state. A network may experience congestion by the time the queue grows. Additionally, a sensor node might deplete its energy prematurely due to a high queue length and low energy level. Furthermore, with a binary reward, the quality of an action is given equal weight for successful transmissions from both high energy and low energy sensor nodes.

The paper in [31] employs Q-learning and linear regression to adjust the duty cycle of the sensor nodes. However, the combination of the two schemes may add communication overhead and increase complexity, similar to [29]. The authors developed a QL model with the state space, action space and reward. The normalized traffic load is considered as the state, and the best action, defined as the duty cycle, is determined based on the load. However, the Q-table dimensions may vary randomly due to the load being used as a state, resulting in slow or non-convergence for larger loads. Additionally, the scheme has not been compared to any other ML-based protocol.

The proposed work in [32] adjusts the duty cycle and packet forwarding using RL and the Monte Carlo technique. The aim is to increase energy efficiency and reduce delay. The duty cycle is adjusted using an event driven approach. However, the combination of the two schemes may add communication overhead and increase complexity, similar to [29], [31]. Furthermore, no other ML-based protocol has been compared to the proposed technique

The authors in [33] propose an RL based sleep scheduling protocol that is adaptive to temperature. The authors analyze the impact of temperature variations on energy consumption. The sensor node takes action to transmit, listen or sleep based on temperature variations in and around the node. The state is considered based on the node's energy and its neighborhood. However, RL is implemented on each node which may affect the learning process as well as consuming more energy during the exploration phase. Additionally, it takes a long time to converge.

In [34], the authors introduced rlDC-MAC to balance network performance and energy efficiency using Q-learning. The network lifetime was extended by calculating the duty cycle for sensor nodes. Whereas, the network performance was improved by calculating favorable transmission contention window. When a sensor node has data packet to send, it starts by calculating the wake-up duration and favorable contention window before transmitting the data packet. The sink node receives the data packet and responds by sending an acknowledgement. However, the protocol suffers from a faster depletion rate of the battery. Additionally, the protocol considers a triplet of parameters as a state, which restricts its practical feasibility. Consequently, the dimensions of Q-value table increase due to the large number of state-action pairs. Furthermore, the complexity of the protocol itself is increased. Moreover, the learning is dependent on the centralized arbitration of specific gateway nodes.

The authors in [35] proposed a Q-learning MAC (QL-MAC) protocol to extend the network lifetime. The QL-MAC modifies the duty cycle of the node based on its traffic load and transmission state of its neighboring nodes. It divides the time into frames, and frames into slots like asynchronous CSMA-CA protocol. Q-values are computed and stored in every slot within each frame. However, the primary focus of the protocol is to increase network lifetime, which results in high probability of extended delay and low throughput throughout the whole simulation. Additionally, the protocol might be less adaptive to the dynamic network conditions due to its heavy dependence on Q-learning hyper parameters that are selected manually. Furthermore, it requires a direct control channel to exchange the learned information.

Most of the traditional MAC protocols [21]-[26] use a fixed formula to adjust the duty cycle adjustment. In some protocols, the sensor node employs a constant duty cycle throughout its lifetime. Consequently, either the network performance or the lifetime of the node is compromised. Additionally, traditional MAC protocols are less adaptable to dynamic aspect of network. Lastly, the efficient use of the available energy remains a concern.

ML-based MACs offer a potential solution to overcome the limitations faced in traditional MACs. It is observed in the literature that Q-learning is used more often to address the dynamic nature of WSNs. In particular, Q-learning based MAC protocols adjust the duty cycle adaptively to frequent changes in the network. However, some research works consider excessive network parameters for Q-learning, which increases its complexity and limits its effectiveness. Additionally, practical feasibility is restricted due to the lack of network information. A summary of ML-based MAC protocols [27]-[35] is presented in Table 1. These protocols mainly suffer from increased complexity and overheads, high dimensionality, and delayed convergence.

## III. MACHINE LEARNING ALGORITHMS FOR WSNS-ENABLED IOT

In WSNs-enabled IoT, ML is widely employed to overcome various issues such as energy efficiency, quality of service, data aggregation, data integrity, routing, localization, synchronization and energy forecasting [36], [37]. They have been implemented in various applications such as in predictive

**TABLE 1. Summary of ML-based MAC protocols.**

| Author(s) and Publication Year | Objective | ML Technique | Limitations |
|---|---|---|---|
| El-Shenhabi et al. (2025) [27] | To develop a dynamic clustering technique based on an adaptive sleep scheduling to increase energy efficiency and prolong WSN lifetime | Q-learning | • Increased complexity<br>• Delayed convergence<br>• Increased computational overhead |
| Wang et al. (2023) [28] | To adjust the sleep period for energy efficiency and extending network lifetime | Q-learning | • High dimensionality<br>• Increased complexity<br>• No packet level Implementation |
| Dutta et al. (2023) [29] | To develop a multi-tier learning for slot selection and sleep scheduling to balance network performance and energy efficiency | Q-learning, Multi-Armed Bandit | • Increased complexity<br>• Increased overhead<br>• Delayed convergence |
| Afroz et al. (2022) [30] | To adjust the duty cycle of receiver node and active period of sender node to improve network performance. | Q-learning | • Limited state information<br>• Insufficient reward<br>• Less adaptable to dynamic network conditions |
| Huang et al. (2021) [31] | To adjust duty cycle of the sensor node according to load. | Q-learning, Linear regression | • Increased complexity and overhead<br>• Delayed convergence<br>• Not compared to ML |
| Huang et al. (2021) [32] | To adjust duty cycle for energy efficiency and delay reduction | Q-learning, Monte Carlo | • Increased complexity<br>• Increased overhead<br>• Not compared to ML |
| Banerjee et al. (2020) [33] | To adjust sleep schedule based on temperature variations | Q-learning | • Increased complexity<br>• No packet level Implementation |
| Trinh et al. (2020) [34] | To adjust duty cycle and contention window size to balance energy efficiency and network performance. | Q-learning | • High dimensionality<br>• Increased complexity<br>• Delayed convergence |
| Savaglio et al. (2019) [35] | To adjust duty cycle of the sensor node for extending network lifetime. | Q-learning | • Heavily relies on hyper parameters<br>• Relies on explicit control channel |

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3606053



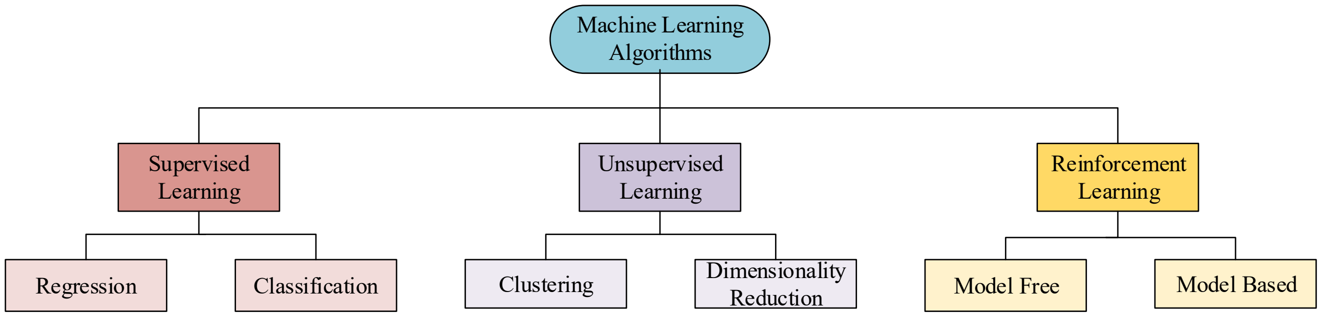


**FIGURE 2.** **Taxonomy of machine learning algorithms.**

maintenance and fault diagnosis [38], smart health care systems [39], agriculture [40], and others. The ML algorithm takes a dataset as an input for training the model. The model is then evaluated to assess its accuracy. The process of evaluation and optimization continues iteratively until the model achieves the required level of accuracy. Lastly, the trained model is validated on a new dataset to ensure a balance between overfitting and underfitting [41]. Figure 2 illustrates the main categories of the ML algorithms.

Supervised learning uses a labelled dataset to train the model [42]. It can be categorized into regression and classification. Regression is used to predict quantitative variables while classification is applied to predict categorical outcomes [42], [43]. Unsupervised learning algorithms use unlabeled dataset that contain only input data. Thus, the model tries to identify hidden patterns and groups data according to their similarities [37]. Unsupervised learning is categorized into clustering and dimensionality reduction. Clustering groups similar data components into clusters whereas dimensionality reduction reduces the number of input features to address the issue of high dimensionality [44].

RL does not require prior datasets to train a model. Instead, RL relies on data collected through continuous interaction with the environment [16]. RL utilizes a trial-and-error based strategy and is iterative in nature. An RL agent explores the environment and takes action based on either exploration or prior experience. The agent learns from the reward, which is considered as the impact of its action on the environment. In the simplest form, the reward can be positive or negative. However, a complex reward function can be formulated based on environmental conditions. The training process continues until the reward saturates or attains a predefined level [45]. RL can be further divided into model-free and model-based techniques. The RL agent learns through its continuous interaction with the environment in model-free technique. The agent does not model the environment. Model-free techniques include Q-learning and SARSA. In model-based technique, the agent builds a model that resembles the environment and takes actions based on this model. Monte Carlo method is an example of model-based RL technique [46].

Q-learning is a popular RL method which is widely used in WSNs [19]. A Q-learning agent does not depend on any prior knowledge of an environment. Instead, it interacts with the

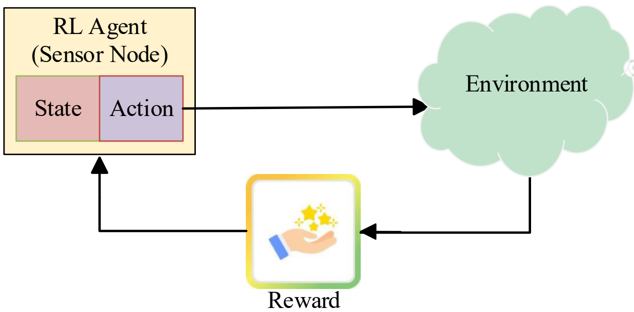


**FIGURE 3.** **Q-learning mechanism [47].**

environment and learns by experience. The agent observes the current state in the environment and takes an action. It receives a reward depending on the impact of the action and updates to the new state as shown in Figure 3. The actions of the agent depend on a Q-function which signifies the quality of a particular action at a particular state. The Q-function is updated iteratively as follows [48]:

$$Q(S_{t+1}, A_t) = (1-\alpha)Q(S_t, A_t) + \alpha\,[R + \gamma max_a Q(S_{t+1}, a)] \quad (1)$$

where $Q(S_t, A_t)$ is the Q-value for the current state $S_t$ and action $A_t$, $Q(S_{t+1}, A_t)$ is the updated Q-value for the next state, $t$ is the time step, and $R$ is the reward for state $S_t$ and action $A_t$. The variable $a$ denotes all possible actions available for the new state $S_{t+1}$. The term $\gamma$ is called the discount factor and its value is between 0 and 1. The value of discount factor is chosen either close to one for preferring the long-term reward or close to zero for preferring the short-term reward. $\max_a Q(S_{t+1}, a)$ is the maximum Q-value and $\gamma \max_a Q(S_{t+1}, a)$ is the discounted future value. $\alpha$ denotes the learning rate which controls the speed of learning and its value ranges between 0 and 1.

The agent can take actions based on either exploration or exploitation. Exploitation means the agent takes the action with the highest Q-value. On the other hand, in exploration, the agent takes a random action by ignoring the Q-value to discover new possibilities of maximizing the reward. If the agent explores too much, it might be wasting a lot of time on random actions. On the other hand, an agent might miss out on better actions when it relies more on exploitation. Therefore, it is crucial to strike the right balance between exploration and exploitation. The epsilon-greedy ($\varepsilon$-greedy) policy is employed to achieve this balance. In the ε-greedy policy, the agent explores more in the beginning with a probability of $\varepsilon$

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2025.3606053



and slowly transits to exploitation in the end with a probability of $1 - \varepsilon$ [34]. Additionally, the agent can adapt to the frequent changes in the dynamic network conditions using the $\varepsilon$-greedy policy.

## IV. RiD-MAC PROTOCOL DESIGN

This section provides a detailed description of the RiD-MAC protocol design. It is divided into three parts: The overview of the baseline protocol, the Q-learning framework design and the duty cycle adjustment mechanism.

### A. BASELINE PROTOCOL OVERVIEW

The proposed RiD-MAC is a contention based asynchronous protocol. The baseline design of RiD-MAC is inspired by AQSen-MAC Protocol [25]. The RiD-MAC follows a receiver-initiated approach. The receiver wakes up periodically to receive data packets from the intended senders. The operation cycle of the proposed protocol is divided into an active period and a sleep period as shown in Figure 4. Data communication takes place during the active period. Whereas the receiver node conserves energy during the sleep period. The active period duration is fixed to 17ms. On the other hand, the sleep period duration varies based on duty cycle of the receiver node. The sleep period is equal to zero when the DC value becomes one. This means that the node is active during the entire cycle, and it does not sleep. The sleep period is equal to the active period when the DC value is 0.5. The sleep period is 9 times the active period when the DC value is 0.1. The sleep period of the receiver node is calculated using the following equation:

$$T_{sleep} = \frac{T_{active} \times (1 - DC)}{DC} \quad (2)$$

where $T_{sleep}$ is the sleep period and $T_{active}$ is the active period.

The communication overview of the proposed protocol is shown in Figure 5. During the active period, the receiver node listens to the channel and carries out clear channel assessment (CCA) to check the channel status. If the channel is found idle, it broadcasts the wake-up beacon to all senders to announce its availability to receive data packets. The receiver waits for a specific time $T_w$ to get a response from the senders. On the other hand, sender nodes that have data packets, listen to the channel. When a sender node receives the wake-up beacon, it performs CCA to check channel status. If the channel is idle, the sender node transmits the Tx beacon. After receiving the Tx beacon, the receiver terminates the waiting time $T_w$ and transmits the Rx beacon. The Rx beacon shows the readiness of the receiver to accept data packets. The Rx beacon triggers the sender node to transmit data packet. Finally, the receiver node receives the data packet and sends ACK packet to confirm the data reception.

The wake-up beacon carries the source address and receiver energy information. The Tx beacon consists of source address, destination address and network allocation vector. The Rx beacon comprises of source address, selected sender address and network allocation vector. Additionally, the three beacons use frame control and frame check sequence from IEEE 802.15.4 standard.

### B. Q-LEARNING DESIGN

The RiD-MAC protocol employs Q-learning to adjust the sleep period of the receiver node. The protocol adopts different duty cycle values based on remaining energy intelligently. On the contrary, other MAC protocols such as AQSen-MAC use

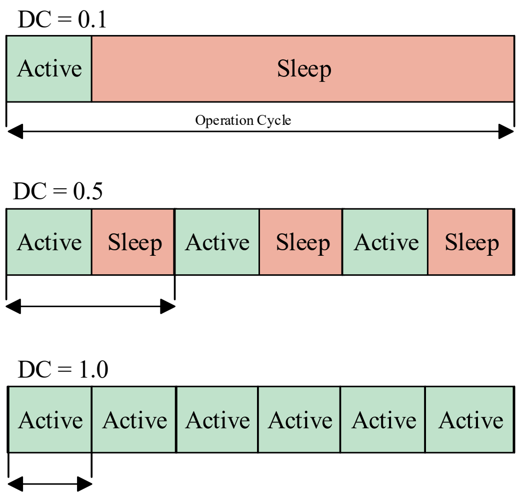


FIGURE 4. Operation cycle of RiD-MAC protocol.

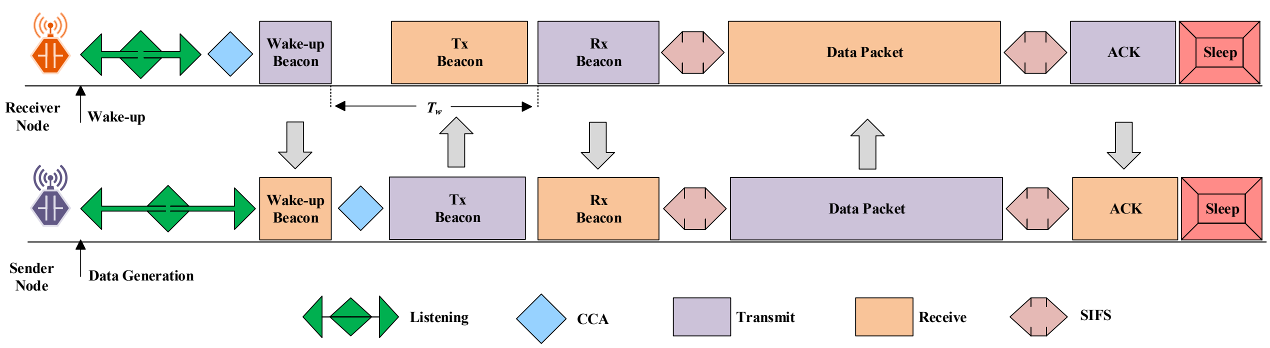


FIGURE 5. Communication overview of RiD-MAC protocol.

a fixed formula to adjust the duty cycle. The design of Q-learning is discussed below:

### 1) STATE SPACE

The state space, $S$ is designed with respect to remaining energy, $E_L$. It is discretized to four energy levels in percentage. Correspondingly, the node can attain four states. The sensor node starts with a current energy level of 100%.

$$S = (10, 40, 70, 100)$$

### 2) ACTION SPACE

The action space, $A$ represents the duty cycle of the receiver node. It is also discretized to four DC values. Correspondingly, the node can take action from the four values. The node starts with a duty cycle of 0.5 for the first cycle.

$$A = (0.1, 0.4, 0.7, 1.0)$$

### 3) Q-VALUE TABLE

The Q-value table stores Q-values for each state-action pair. It is updated iteratively based on (1) and comprises of rows and columns. The structure of the Q-value table is specified by states as its rows and actions as its columns. The order of the Q-value table is rows by columns. Since the RiD-MAC has four states and four actions hence the order is 4 × 4, and the table has a total of 16 Q-values. Initially, all the Q-values are initialized to zero as shown in Table 2.

The main diagonal highlighted in green color represents the best action for each state. For example, the node will receive maximum reward for state 70% only if it chooses the best action of 0.7. The Q-values positioned under the main diagonal represent insufficient actions for each state. For example, if the node chooses an action of 0.4 at state 70%, it will receive low reward. This is because the energy is still sufficient at this state, so the network performance will be degraded if a low duty cycle is chosen. Hence, this action is considered insufficient. Similarly, the action 0.7 or lower and the action 0.1 are insufficient for state 100% and state 40%, respectively. The Q-values placed above the main diagonal represent excessive actions for each state. For example, if the node chooses an action of 0.7 at state 40%, it is considered excessive. Because the remaining energy is low at this state, so the network lifetime will be affected if a high duty cycle is chosen. Hence, the node should be prevented from choosing such actions. Similarly, the action of 0.4 or greater and action 1.0 are excessive for state 10% and state 40%, respectively.

The order of Q-value table is critical in Q-learning design. It depends on the number of discrete levels for state space and action space. Complexity of Q-learning increases with more discrete levels for state and action space. As a result, Q-learning will experience frequent transients due to quick changes in state. Additionally, the change in state may occur before the best action is found for the given state. Another factor that contributes highly to the complexity of Q-value table is the number of network parameters considered for state and action spaces. This creates the issue of high dimensionality. For example, rlDC-MAC considers three network parameters in the state space and two network parameters in the action space. This results in a multi-dimensional Q-value table. If all the network parameters are assumed to be discretized to four levels, then the total number of combinations for the state is equal to 64. Whereas the total number of combinations for action equal to 16. This means, there are 16 actions available to be taken during each state, which results in 1024 Q-values. Consequently, its Q-learning may experience increased complexity, frequent transients, reduced steady state durations and inadequate actions for various states. The proposed RiD-MAC addresses these concerns by considering one network parameter for the state and the action, resulting in two-dimensional Q-value table. The QX-MAC also considers one parameter as a state, but it might not be able to capture the dynamic network conditions, as discussed in Section II.

**TABLE 2.** Initial Q-value table.

| States | Actions | | | |
|---|---|---|---|---|
| Remaining Energy (%) | Duty Cycles | | | |
| | 0.1 | 0.4 | 0.7 | 1.0 |
| 10 | 0 | 0 | 0 | 0 |
| 40 | 0 | 0 | 0 | 0 |
| 70 | 0 | 0 | 0 | 0 |
| 100 | 0 | 0 | 0 | 0 |

### 4) REWARD

The reward, $R$ is designed with throughput, $T$ and energy consumed, $E_C$. The reward will be maximized when either the throughput increases, or the energy consumption decreases. Therefore, the node strives to increase throughput and reduce energy consumption. It is formulated as follows:

$$R = w_T T - w_E E_C \tag{3}$$

where $w_T$, and $w_E$ are the corresponding weight coefficients. The weight coefficients' values are adaptive to the state of the node and the action taken during the state. The reward is maximized if a higher value of duty cycle is chosen as an action when energy level is high, or a lower duty cycle value is chosen when energy level is low. On the other hand, the reward is minimized if a lower value of duty cycle is chosen as an action when energy level is high or a higher value of duty cycle is chosen when the energy level is low.

Throughput, $T$ is calculated as:

$$T = \frac{N_{pktRx} \times L_{pkt}}{T_s} \tag{4}$$

where $N_{pktRx}$ is the total number of packets received, $L_{pkt}$ is the data packet size in bits and $T_s$ is the simulation time.

Energy consumed, $E_C$ is calculated as:

$$E_c = \sum_{i=0}^{n} P_i \times t_i \tag{5}$$

where $n$ is the total number of radio states, $P_i$ is the power consumption rate of radio state $i$ and $t_i$ is the time spent in radio state $i$.

The choice of reward parameters reflects the balance between network performance and energy efficiency. On the contrary, QX-MAC considers binary reward which may not be able to fully account for the dynamic network conditions as discussed in Section II.

#### 5) NEXT STATE

The next state is updated based on maximum battery energy and cumulative energy consumed from start to the current operation cycle. It is obtained by:

$$S_{t+1} = E_m - E_{cc} \tag{6}$$

where $S_{t+1}$ is the next state, $E_m$ is the maximum battery energy and $E_{cc}$ is the cumulative energy consumed.

#### 6) COMPUTATIONAL COMPLEXITY

RiD-MAC involves a total of around 10 individual Q-learning logic operations executed during each cycle. All individual operations are performed in constant time $O(1)$, which is independent of state-action space size. These include lookups, such as fetching consumed energy and number of received packets. They also include arithmetic operations for calculating reward and updating the Q-value using the Bellman equation. Lastly, comparisons are made for determining the current state or next state. On the other hand, exploitation and finding maximum Q-value for next state are considered action dependent operations whose time grows linearly with the number of actions, denoted as $O(A)$. During exploitation, the sensor node scans and compares all Q-values for the current state to determine the action with the highest Q-value. The maximum Q-value for the next state is obtained in a similar way. Additionally, Q-value table printing and initialization to zeros, requires $O(S \times A)$ time but these operations are performed only once during start-up, hence their impact is negligible. On the other hand, $O(A)$ and $O(1)$ are observed during each cycle but $O(A)$ is more dominant. Therefore, the overall time complexity of Q-learning logic processing is $O(A)$ per cycle [49]. Additionally, the node executes RL logic after packet receptions and just before sleeping, not continuously, which further reduces its computational load. Therefore, Q-learning logic only adds negligible processing overhead.

The memory complexity for Q-value table storage is given by $O(S \times A)$ [49]. RiD-MAC stores 16 Q-values in the Q-values table and each value requires 4 bytes (float), which results in a total memory usage of 64 bytes. Hence, this memory size is lightweight even for IoT devices operating as low as 2 kB of RAM [50]. Therefore, the proposed design is both memory efficient and scalable to more constrained IoT nodes.

### C. DUTY CYCLE ADJUSTMENT MECHANISM

The mechanism of the proposed duty cycle adjustment is depicted in Figure 6. The receiver node wakes-up and checks if it is its first wake-up, called primary wake-up, or not. If it is primary wake-up, then it sets the initial Q-values to zeros, which is done only once. Otherwise, the node observes the current state by checking its remaining energy. Next, it initiates the data communication process which includes CCA

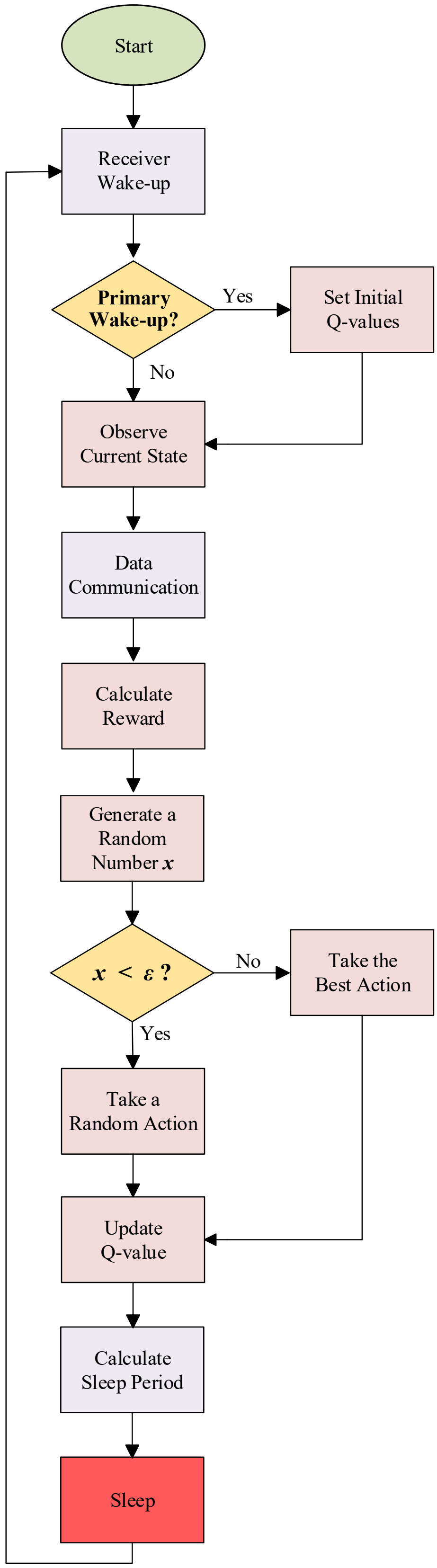


FIGURE 6. **Duty cycle adjustment mechanism of RiD-MAC protocol.**

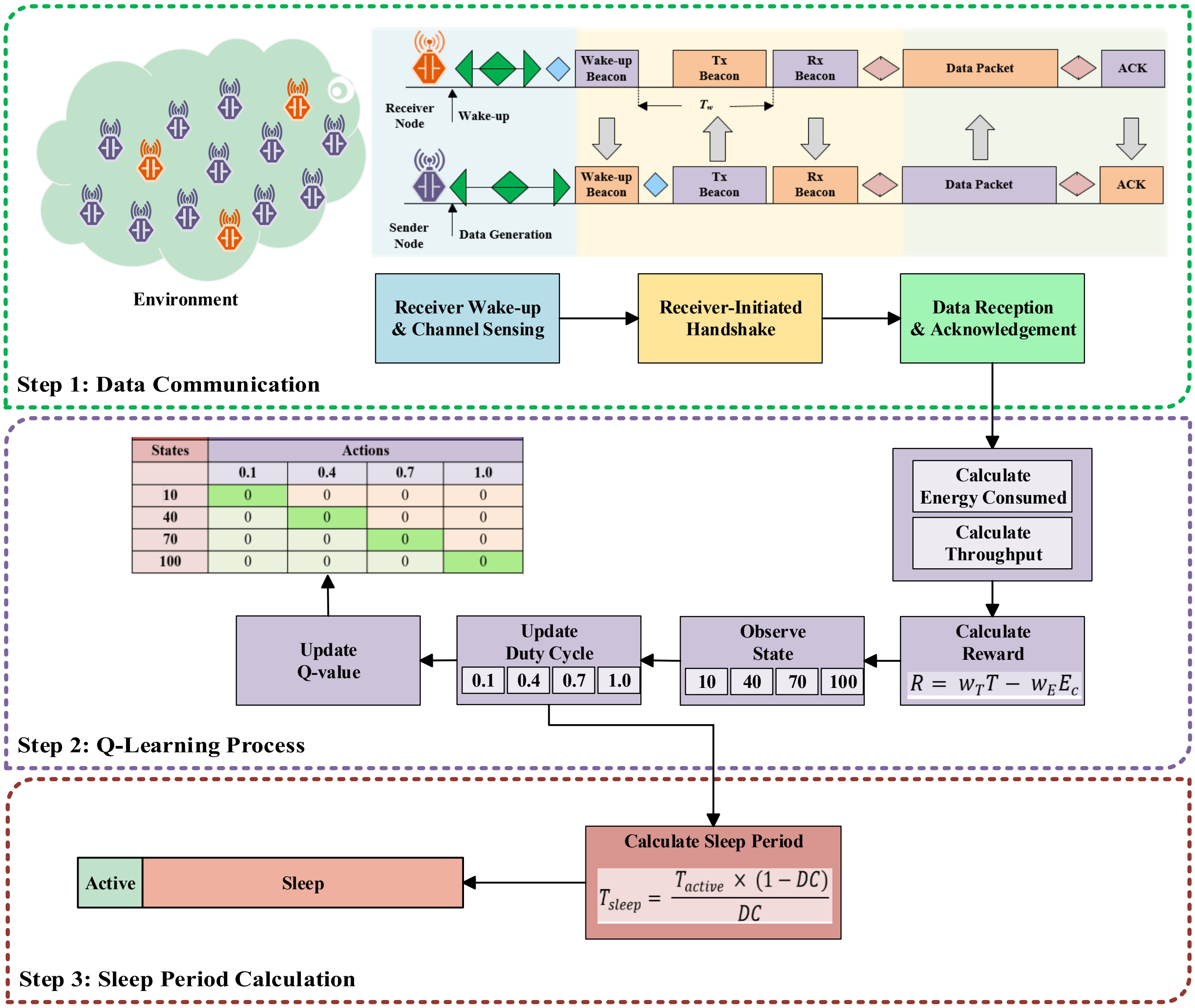


**FIGURE 7.** Block diagram of RiD-MAC protocol.

to check if the channel is busy or idle, broadcasts a wake-up beacon, exchanges Tx and Rx beacon, receives data packet and transmits the ACK packet, as explained in detail in Figure 5. At the end of the data communication, the receiver node calculates the accumulated throughput and energy consumed during the current cycle to determine the reward using (3). Additionally, the energy consumed is used to calculate the next state using (6). Next, the duty cycle is chosen based on the $\varepsilon$-greedy policy. The node generates a random number x, and if the random number is less than the exploration probability ($\varepsilon$) value, then exploration takes place, and a random action is chosen from the action space. Otherwise, exploitation takes place, and the best action is chosen based on the Q-value table. Then, the Q-value is updated using (1) and stored in the Q-value table. This ensures that the node continually learns from its experience. Subsequently, it calculates the sleep period using (2) and goes back to sleep. The process is then repeated in the subsequent cycles.

The overall architecture of RiD-MAC protocol can be summarized in three steps as shown in Figure 7. Data communication takes place in the first step at the MAC layer. The receiver node broadcasts a wake-up beacon after each wake-up and exchanges Tx and Rx beacons. Subsequently, it receives data packets and acknowledges. Once data communication is over for a given cycle, the Q-learning process is initiated. The node calculates the reward by observing the energy consumed and throughput accumulated in the cycle. It receives a high reward when energy consumption is minimized or throughput is maximized, and vice versa. Next, the node checks the remaining energy and determines the current state by comparing it with four predefined energy levels in the state space.  After that, it chooses an action from the action space, consisting of four actions, to update the duty cycle based on the reward and the current state. Then, the node calculates a new Q-value and updates the Q-value table. In the end, the node calculates its sleep period using the chosen duty cycle and goes to sleep for the specified duration.

## V. RESULTS AND DISCUSSIONS

The proposed RiD-MAC protocol is implemented and evaluated on Castalia simulator [51] based on OMNeT++ platform [52]. It is an open-source simulator for WSNs. Researchers broadly use it to test protocols in realistic radio models and wireless channels. In particular, Castalia provides realistic sensor node behavior related to radio access methods. It gives reliable validation of a protocol before implementation on a specific sensor platform. In RiD-MAC protocol, the sensor nodes use CC2420 radio. It is a commonly employed radio frequency transceiver in various wireless communication applications due to its low power consumptions. The radio operates in four different states such as transmission state, reception state, sleep state and idle listening state.

In this work, the sender nodes adopt the p-persistent CSMA method to access the wireless channel. The node senses the channel continuously until it is idle, and transmits with respect to probability $p$. The value of $p$ depends on the number of sender nodes and is given by $1/n_{SN}$, where the $n_{SN}$ is number of sender nodes. For example, if there are two sender nodes then each node has a probability of 0.5 to transmit to the receiver successfully. The p-persistent CSMA experiences moderate collision and delay in comparison to 1-persistent CSMA and non-persistent CSMA [36]. 1-persistent CSMA can cause frequent collisions due to its immediate transmission after sensing an idle channel. On the other hand, non-persistent CSMA reduces collisions by introducing random back-off time after busy channels, but it often increases the overall delay. Therefore, p-persistent CSMA achieves an effective trade-off between minimizing collisions and maintaining reasonable transmission delays.

The results are presented in two subsections. The first subsection provides baseline evaluation with varying number of sender nodes from 1 to 7. The second subsection includes evaluation under dynamic network conditions and scalability of Q-learning state-action space configuration. The simulations are run for 1200 s. Each sender node generates 1200 packets at a packet rate of 1 packet/s. The data packet length is 28 bytes. All the results and the performance comparisons include confidence intervals (CI) for both the subsections. Simulation was executed for 30 repetitions for each scenario with a different set of random seeds for each repetition. Castalia simulator can be instructed to calculate 95% CI of the results over all the repetitions. The 95% CI means that the true average of the parameter lies within the specified interval for 95% of the possible sample sets chosen. The simulation parameters are listed in Table 3.

**TABLE 3. Simulation parameters.**

| Parameter | Value |
|---|---|
| Simulation duration | 1200 s |
| Sender nodes | 1 to 7 |
| Number of repetition runs | 30 |
| Radio type | CC2420 |
| Active period ($T_{active}$) | 17 ms |
| $T_w$ | 5 ms |
| Slot time duration | 0.32 ms |
| SIFS | 0.192 ms |
| CCA delay | 0.128 ms |
| Maximum battery energy ($E_m$) | 100 J |
| Initial energy level | 100% |
| Wake-up beacon length | 6 bytes |
| ACK packet length | 11 bytes |
| Rx beacon length | 13 bytes |
| Tx beacon length | 14 bytes |
| Data packet length | 28 bytes |
| Data rate | 250 kbps |
| Packet rate | 1 packet/s |

**TABLE 4. Q-learning parameters.**

| Parameter | RiD-MAC | rlDC-MAC | QX-MAC |
|---|---|---|---|
| States | 4 | 27 | 3 |
| Actions | 4 | 3 | 3 |
| Learning rate | 0.9 | 0.5 | 0.5 |
| Discount factor | 0.1 | 0.5 | 0.618 |
| Max exploration probability | 1 | 0.7 | 1 |
| Min exploration probability | 0.1 | - | 0.05 |
| Decay Factor | 0.003 | - | 0.00001 |

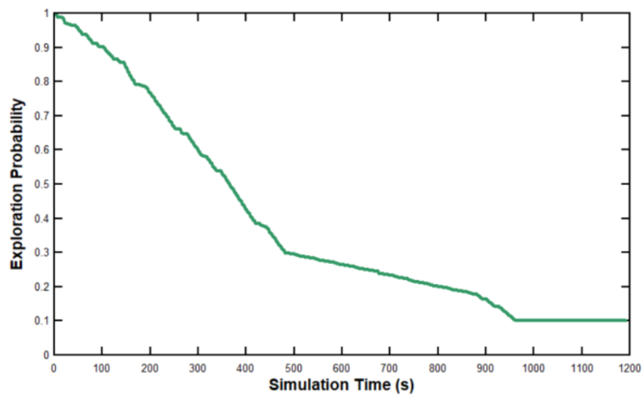


**FIGURE 8. Exploration probability decay.**

The Q-learning parameters are presented in Table 4 for the three protocols except the AQSen-MAC. The RiD-MAC has 4 states and actions. Whereas the rlDC-MAC has 27 states since its three state variables are discretized to 3 values each. However, only one action variable is considered for fair comparison. The state and action for the QX-MAC are discretized to 3 values each. The RiD-MAC employs a higher learning rate of 0.9 in comparison to the others to achieve faster convergence and ensure adaptability to the frequent changes in the dynamic network conditions. Similarly, a low discount factor prioritizes the immediate rewards and ensures the node learns quickly. The resultant exploration probability decay is illustrated in Figure 8. The exploration probability is decayed by a factor of 0.003 from the maximum value of 1 to the minimum value of 0.1. This allows the node to explore more in the beginning while relying more on exploitation towards the end of the simulation.

### A. BASELINE EVALUATION

The performance of the proposed protocol is evaluated using a single-hop scenario, representing a single cluster configuration. In this configuration, up to seven sender nodes communicate directly to one receiver node, ensuring small cluster size. This is in line with [53], [54], which emphasize that cluster size of around 5 to 8 nodes improves energy efficiency and reduces intra-cluster communication overhead under single hop. Additionally, [55] suggests that cluster size should be below 10 nodes to achieve a balanced trade-off between energy consumption and performance. Therefore, RiD-MAC protocol forms a solid base, which is scalable to multi-cluster and multi-hop communication scenarios.

The sensor nodes are connected within a network through star topology. In this topology, all the sender nodes communicate directly with a single central node, called the receiver node. It manages data collection and coordination. The star topology simplifies communication paths and reduces routing complexity in comparison to mesh or tree topologies. The receiver node is connected to seven sender nodes, which are located within an area of 900 $m^2$, as shown in Figure 7.

The final Q-value table is shown in Table 5. The main diagonal highlighted in green color contains the highest Q-values for each state action. The highest Q-value shows the best action for each state, which is desired, as explained in Section IV. The node has four actions to choose from at each state. It starts from state 100% and chooses an action randomly. Every state will have a transient period of a few cycles in the beginning to learn the best action. Once the best action is learnt, the node attains steady-state and settles down to the best action until the state changes. Let us assume the

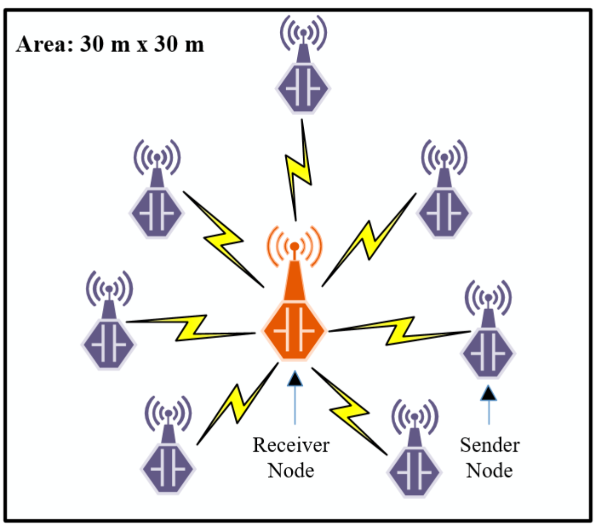


FIGURE 7. Simulation network topology.

TABLE 5. Final Q-value table.

| States | Actions | | | |
|---|---|---|---|---|
| Remaining Energy (%) | Duty Cycles | | | |
| | 0.1 | 0.4 | 0.7 | 1.0 |
| 10 | 12.69 | 4.44 | 4.53 | 0 |
| 40 | 20.3 | 110.52 | 9.87 | -245.83 |
| 70 | -247.93 | 21.03 | 110.65 | 21.02 |
| 100 | 77.29 | 193.38 | 290.81 | 399.58 |

action of 0.4 was chosen for the first cycle, a low reward will be given to the node for taking an insufficient action at this state. In the next cycle, the node may choose the same action or a different one randomly. Since the exploration probability is high at the beginning, so it's very likely that the node will take a different action. For example, it chooses the action of 0.7 in the second cycle. Again, a low reward will be given to the node but slightly higher than that of action 0.4. It will be similar if action 0.1 is chosen but the reward will be the lowest. When the node chooses the action of 1.0, it will get maximum reward because this is the best action for this state. After several cycles, the node will start to settle down to the best action. When the state changes from 100% to 70%, the node enters the transient period again and explores by taking a random actions. It may choose insufficient actions or excessive actions before settling down to the best action. However, it will only get the maximum reward at action of 0.7. Similarly, the process is repeated for state 40% and state 10%.

The receiver node duty cycle is shown in Figure 9. It can be seen that the duty cycle generally corresponds to the state. Some transient changes in duty cycle is due to the exploration nature of the actions taken. Initially, a high duty of 1 is mostly employed until approximately 200 cycles for the state 100%. Duty cycle is around 0.7 for the next 300 cycles for the state 70%. It is further reduced to 0.4 until 1000 cycles for the state 40%. Interestingly, it can be noted that the duty cycle settles to 0.1 at state 10% towards the end due to the exploitation nature. Most importantly, the duty cycle is following the states, which represent the remaining energy, aligning with the desired overall design.

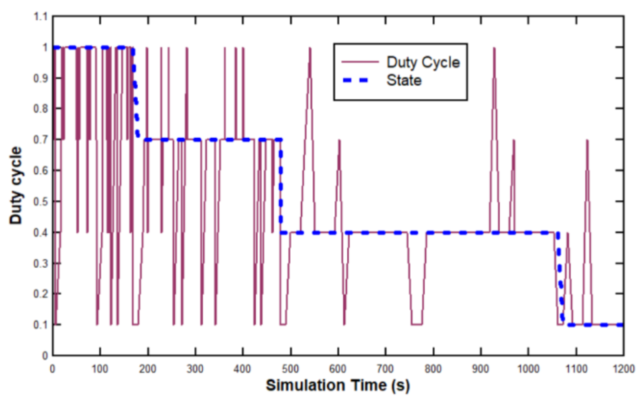


FIGURE 9. Receiver duty cycle.

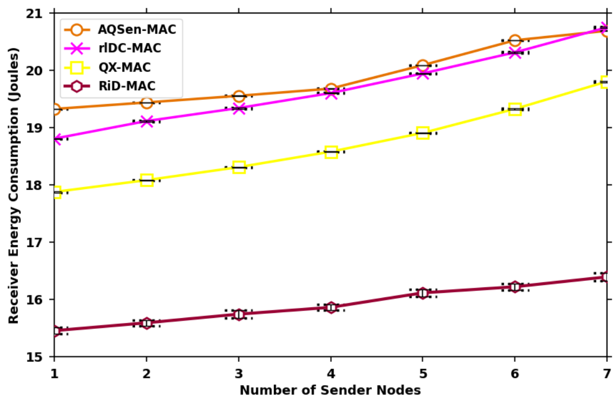


FIGURE 10. **Receiver energy consumption.**

Figure 10 shows the receiver energy consumption in Joules, which is calculated using (5). It can be observed that the energy consumption increases with increased number of sender nodes, as expected. In addition, the RiD-MAC protocol significantly outperforms rlDC-MAC, AQSen-MAC and QX-MAC by up to 21%, 21% and 17%, respectively. This is because the RiD-MAC adapts well to the remaining energy. The RiD-MAC chooses a low duty cycle to sleep more when it has low energy to prolong the network operation. Whereas QX-MAC performs slightly better than the rlDC-MAC and the AQSen-MAC. AQSen-MAC consumes higher energy because it does not consider network conditions and only uses a fixed formula to adjust the duty cycle. Interestingly, rlDC-MAC consumes slightly more energy than AQSen-MAC for senders 6 and 7, because it receives more packets compared to others.

The receiver energy consumption per bit is illustrated in Figure 11. It is defined as the ratio of the total energy consumed to the total packets received in bits. It is measured in J/bit and calculated using the following equation:

$$E = \frac{E_c}{N_{pktRx} \times L_{pkt}} \quad (7)$$

The RiD-MAC protocol improves receiver energy consumption per bit by up to 26%, 19% and 15% when compared to AQSen-MAC, rlDC-MAC and QX-MAC,

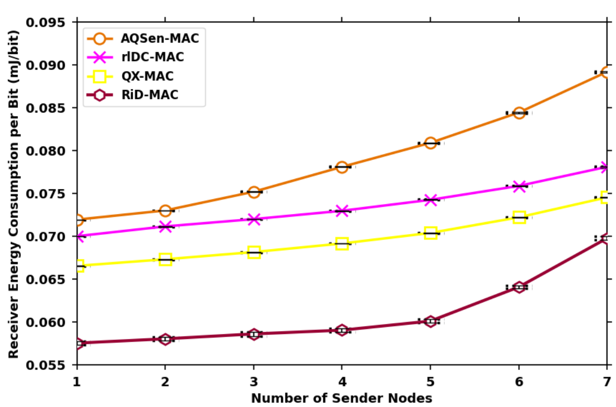


FIGURE 11. **Receiver energy consumption per bit.**

respectively. Among the compared protocols, rlDC-MAC and QX-MAC clearly outperform AQSen-MAC. The energy consumption per bit increases slightly when the sender nodes is above 5 for RiD-MAC. However, it is still significantly less than the other protocols.

Figure 12 illustrates the network throughput. The throughput is measured in bits per second (bps) and calculated using (4). It is observed that RiD-MAC improves network throughput by up to 8% when compared to AQSen-MAC. On the other hand, it maintains throughput on par with rlDC-MAC and QX-MAC for 1 to 5 senders, which is around 1120 bps for 5 senders. However, its throughput is marginally lower, around 1265 bps compared to 1338 bps for 6 senders, and around 1370 bps versus 1550 bps for 7 senders. Nevertheless, RiD-MAC outperforms rlDC-MAC and QX-MAC with better energy consumption per bit for 6 and 7 senders. This means that rlDC-MAC and QX-MAC receive more packets but at the expense of significantly higher energy consumption. On the contrary, RiD-MAC receives marginally less packets but consumes much less energy.

The average end-to-end packet delay is presented in Figure 13. It is defined as the time taken from generation of a data packet at a sender node to its successful reception at the receiver node. It is calculated as the sum of queuing ($d_{queu}$), transmission ($d_{tran}$), propagation ($d_{prop}$) and processing

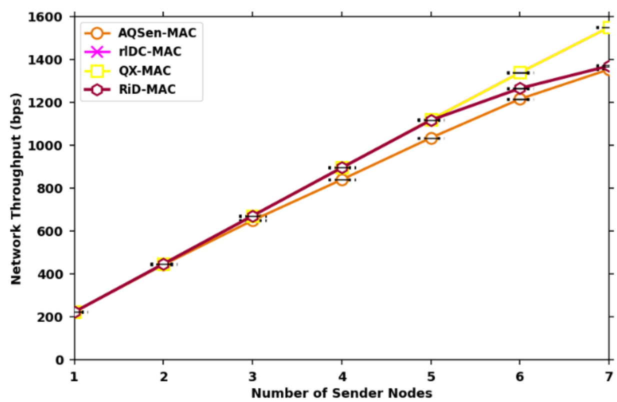


FIGURE 12. **Network throughput.**

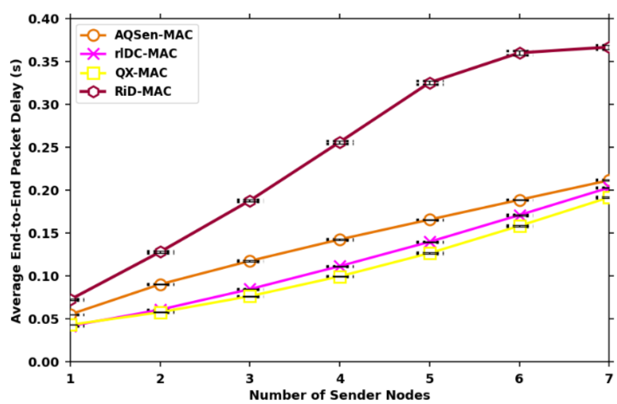


FIGURE 13. **Average end-to-end packet delay.**

($d_{proc}$) delays and is given as below:

$$D = d_{queu} + d_{tran} + d_{prop} + d_{proc} \quad (8)$$

It can be observed that the RiD-MAC experiences higher delay in comparison to others. This is because the receiver node chooses to sleep more when it has low energy in order to prolong the network lifetime. Consequently, the sender node waits slightly longer until the receiver wakes up. Additionally, the queue length may increase due to high packet rate or large number of sender nodes. Hence, it is a trade-off between energy efficiency and delay, in that there will be an inevitable increase in delay as RiD-MAC conserves energy by sleeping more, especially when the remaining energy is low. However, the proposed RiD-MAC achieved maximum average end-to-end packet delay of below 0.4 s, which is still within the tolerable range. Therefore, it can be adopted for various IoT applications, including elderly healthcare (in both indoor and outdoor scenarios) and industrial manufacturing process monitoring, where a packet delay of less than 1 s is acceptable [56], [57]. It is also suitable for other IoT domains such as smart agriculture and infrastructure management, where energy-efficient and intelligent data transmission is critical for extending network lifetime, and delays between 1 s to 5 s is still considered as real-time [56], [57], [58], [59]. Nevertheless, it is also important to note that the parameters of RiD-MAC can be adjusted if the application requires further delay reduction, at a slightly higher energy consumption.

### B. EVALUATION UNDER DYNAMIC SCENARIOS

In this subsection, the adaptability of RiD-MAC is firstly demonstrated across node mobility and hybrid traffic. Secondly, evaluation of different state-action configurations shows scalability of the RiD-MAC design. Both performance evaluations are carried out with four sender nodes. Additionally, stationary nodes with the periodic traffic scenario is considered as the baseline evaluation.

#### 1) HYBRID TRAFFIC AND NODE MOBILITY

The hybrid traffic comprises of a periodic phase followed by a bursty phase to replicate non-emergency and emergency scenarios, respectively [60]. The simulation time is divided into repeating blocks of 20 s each. During the initial 15 s, the node transmits 5 packets by employing a low packet rate of 0.333 packets/s. In the last 5 s, the burst mode is activated where the node transmits 15 packets at higher packet rate of 3 packets/s. The bursty phase is modeled as a Poisson process with a mean packet rate of 3 packets/s. This allows the node to generate packets randomly to capture the stochastic traffic nature of event driven IoT applications. Nevertheless, the Poisson process ensures that the average number of packets remain consistent to the mean.

The receiver node employs the Random Waypoint Mobility model to emulate a realistic dynamic scenario. This model is commonly used for movement of nodes in WSNs [61]. In accordance with the model, the receiver node selects a random location and chooses a random speed between 12 m/s to 15 m/s. Once it reaches the destination, it pauses for a duration of 4 s and again selects a new random location to move towards at a random speed. The process continues throughout the duration of the simulation.

Figure 14 shows the receiver energy consumption per bit for stationary nodes with periodic and hybrid traffic, and mobile nodes with periodic traffic. This metric reflects the energy cost of receiving data hence it is a key indicator of the protocol's efficiency in resource-constrained IoT applications. Under hybrid traffic, RiD-MAC demonstrates improvements of up to 21%, 15% and 12% against AQSen-MAC, rlDC-MAC and QX-MAC, respectively. rlDC-MAC and QX-MAC is able to maintain their performance quite well due to their choice of state space, which includes energy and throughput, and queue length, respectively. On the other hand, in the mobile nodes scenario, RiD-MAC is still able to provide improvements of up to 25%, 19% and 14% against AQSen-MAC, rlDC-MAC and QX-MAC, respectively.

The average end-to-end packet delay across stationary nodes with periodic and hybrid traffic, and mobile nodes with periodic traffic, is shown in Figure 15. It is observed that rlDC-MAC and QX-MAC experiences lower delays across the scenarios. This is because rlDC-MAC considers delay as one of the state space parameters, hence it adjusts its duty cycle to reduce the delay. In contrast, QX-MAC increases its duty cycle to tackle rising packet queue lengths. However, both protocols do so at the price of increased energy consumption, resulting in much higher energy consumption per bit, as seen previously. On the other hand, RiD-MAC design choice allows for a marginal increase in delay to obtain meaningful gain in energy efficiency. Nevertheless, it is important to note that its packet delay is still within the acceptable range, and is suitable for various applications as mentioned in the previous subsection. These results show that RiD-MAC protocol continues to exhibit superior energy efficiency and outperforms the state-of-the-art across these dynamic scenarios, while maintaining an acceptable level of delay.

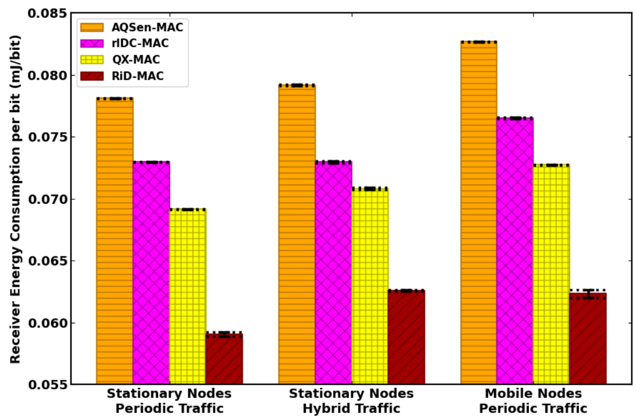


**FIGURE 14. Receiver energy consumption per bit across stationary nodes with periodic and hybrid traffic, and mobile nodes with periodic traffic.**

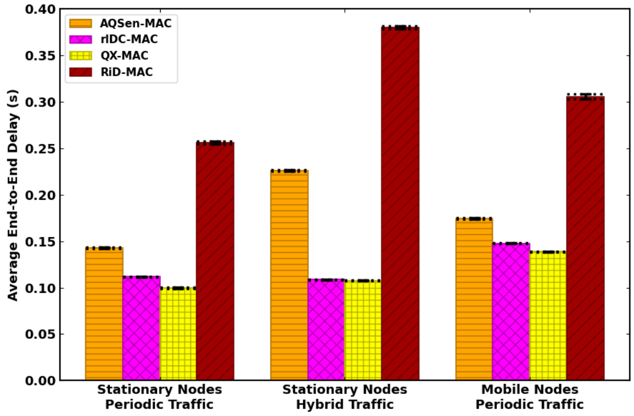


**FIGURE 15. Average end-to-end packet delay across stationary nodes with periodic and hybrid traffic, and mobile nodes with periodic traffic.**

### 2) STATE-ACTION SPACE CONFIGURATIONS

RiD-MAC protocol adopts a 4 × 4 state-action space to ensure a good balance between complexity and performance. Empirical evaluation was performed using state-action space configurations such as 3 × 3, 4 × 4, 5 × 5 and 6 × 6, with four senders per receiver for each case. All configurations were run for 30 repetitions with 95% CI, which confirms the statistical stability of the results. Figure 16 shows metric values that are normalized with respect to the 4 × 4 configuration. The 3 × 3 configuration shows a marginal increase in all the energy and delay performance metrics when compared to the 4 × 4 configuration, thereby favoring the latter. It is also evident that the relative receiver energy consumption increases by up to 10% as the state-action space grows from 4 × 4 to 6 × 6. These results show that the 4 × 4 configuration is the best for this evaluation scenario. Additionally, a large state-action space increases computational complexity and may converge to a poor policy [48]. It also require more memory to store Q-value table [49], which may be infeasible for resource-constrained IoT devices [62]. Moreover, authors in [49] employ a small state-action space with modest state and only two actions to achieve energy and memory efficiency. Therefore, our findings are consistent with prior research work. However, the proposed protocol is not limited to the 4 × 4 configuration. Instead, its design is scalable to larger state-action spaces if required for deployment in more complex and dynamic environments.

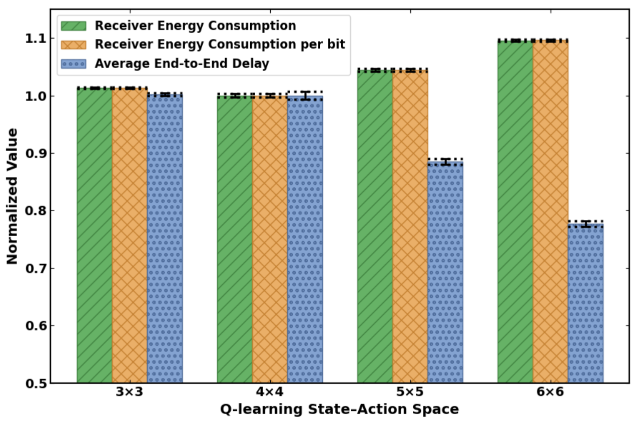


**FIGURE 16. Normalized performance metrics across different state–action space configurations, with the 4 × 4 configuration used as the baseline.**

## VI. CONCLUSION

This paper proposed a reinforcement learning-based RiD-MAC protocol that intelligently adapts to dynamic network conditions by adjusting the duty cycle of the receiver node based on its remaining energy. It was developed by incorporating sufficient network information to address challenges such as complexity and dimensionality, while balancing simplicity and performance effectively. The Q-learning algorithm was designed with remaining energy as the state space and duty cycle as the action space. The reward function was formulated based on energy consumption and throughput, to achieve a good trade-off between energy efficiency and performance. The best action was learnt over time through accumulated experience. It also employed the ε-greedy policy to effectively address the exploration-exploitation dilemma in RL. Simulation results show that RiD-MAC protocol provides significantly reduced receiver energy consumption of up to 21% and reduced receiver energy consumption per bit of up to 26%, when compared to other state-of-the-art protocols. Furthermore, the protocol maintains an average delay of under 0.4 s for up to seven senders, which is still within the acceptable range. Therefore, RiD-MAC protocol can be deployed in various IoT applications, including healthcare, environmental monitoring, smart agriculture and others.

This work demonstrates the adaptability of the proposed protocol to network changes while staying aware of the node's remaining energy. However, the node's limited battery capacity means that energy can only decrease over time, which means that network performance is always traded off for lifetime. Therefore, energy harvesting will be considered as future work to further enhance the network's performance.

## REFERENCES

[1] A. Saleem, S. Shah, H. Iftikhar, J. Żywiołek, and O. Albalawi, "A comprehensive systematic survey of IoT protocols: Implications for data quality and performance," *IEEE Access*, early access, 2024, doi: 10.1109/ACCESS.2024.3486927.

[2] M. Weqar, S. Mehfuz, D. Gupta, and S. Urooj, "Adaptive switching based data-communication model for Internet of Healthcare Things networks," *IEEE Access*, vol. 12, pp. 11530–11548, 2024, doi: 10.1109/ACCESS.2024.3354722.

[3] R. B. Preethi and M. S. Nair, "Augmenting energy sustainability of static nodes using hybrid KGNN-AHP driven approach for IoT-based heterogeneous WSN," *IEEE Access*, vol. 13, pp. 3320–3354, 2025, doi: 10.1109/ACCESS.2024.3523401.

[4] M. Z. Hasan and Z. M. Hanapi, "Efficient and secured mechanisms for data link in IoT WSNs: A literature review," *Electronics*, vol. 12, no. 2, pp. 1–23, Art. no. 292, 2023, doi: 10.3390/electronics12020458.

[5] P. Kaur and P. Singh, "Adaptive data transmission protocols for energy harvesting WSNs used in agriculture," *Journal of Telecommunications and Information Technology*, vol. 1, no. 1, pp. 97–103, Mar. 2024, doi: 10.26636/jtit.2024.1.1390.

[6] L. Nguyen and H. T. Nguyen, "Mobility based network lifetime in wireless sensor networks: A review," *Computer Networks*, vol. 174, pp. 1–24, 2020, doi: 10.1016/j.comnet.2020.107236.

[7] J. Li, T. Han, W. Guan, and X. Lian, "A preemptive-resume priority MAC protocol for efficient BSM transmission in UAV-assisted VANETs," *Applied Sciences*, vol. 14, no. 5, pp. 1–27, Art. no. 2320, 2024, doi: 10.3390/app14052151.

[8] R. Zhu, A. Boukerche, D. Li, and Q. Yang, "Delay-aware and reliable medium access control protocols for UWSNs: Features, protocols, and classification," *Computer Networks*, vol. 252, pp. 1–21, Art. no. 110089, 2024, doi: *10.1016/j.comnet.2024.1100631*.

[9] F. Ojeda, D. Mendez, A. Fajardo, and F. Ellinger, "On wireless sensor network models: A cross layer systematic review," *Journal of Sensor and Actuator Networks*, vol. 12, no. 4, pp. 1–35, Art. no. 50, Aug. 2023, doi: *10.3390/jsan12040050*.

[10] A. Roy and N. Sarma, "A synchronous duty-cycled reservation based MAC protocol for underwater wireless sensor networks," *Digital Communications and Networks*, vol. 7, no. 3, pp. 385–398, 2021, doi: *10.1016/j.dcan.2020.09.001*.

[11] C. Blondia, "Evaluation of the end-to-end response times in an energy harvesting wireless sensor network using a receiver-initiated MAC protocol," *Ad Hoc Networks*, vol. 136, pp. 1–12, 2022, doi: 10.1016/j.adhoc.2022.102995.

[12] P. D. Nguyen and L. W. Kim, "Sensor system: A survey of sensor type, ad hoc network topology and energy harvesting techniques," *Electronics*, vol. 10, no. 2, pp. 1–20, Art. no. 219, Jan. 2021, doi: 10.3390/electronics10020219.

[13] R. Priyadarshi, "Exploring machine learning solutions for overcoming challenges in IoT-based wireless sensor network routing: A comprehensive review," *Wireless Networks*, vol. 30, no. 4, pp. 2647–2673, 2024, doi: 10.1007/s11276-024-03697-2.

[14] E. Barbierato and A. Gatti, "The challenges of machine learning: A critical review," *Electronics*, vol. 13, no. 2, pp. 1–30, 2024, doi: 10.3390/electronics13020416.

[15] E. Dritsas and M. Trigka, "Machine learning in information and communications technology: A survey," *Information*, vol. 16, no. 1, pp. 1–26, 2024, doi: 10.3390/info16010008.

[16] M. Abbasi, A. Shahraki, M. J. Piran, and A. Taherkordi, "Deep reinforcement learning for QoS provisioning at the MAC layer: A survey," *Engineering Applications of Artificial Intelligence*, vol. 102, pp. 1–20, 2021, doi: 10.1016/j.engappai.2021.104234.

[17] A. R. Gaidhani and A. D. Potgantwar, "A review of machine learning-based routing protocols for wireless sensor network lifetime," *Engineering Proceedings*, vol. 59, no. 1, pp. 1–13, 2024, doi: 10.3390/engproc2023059231.

[18] D. Han, B. Mulyana, V. Stankovic, and S. Cheng, "A survey on deep reinforcement learning algorithms for robotic manipulation," *Sensors*, vol. 23, no. 7, Art. no. 3327, pp. 1–35, 2023, doi: 10.3390/s23073762.

[19] P. N. Karunanayake, A. Könsgen, T. Weerawardane, and A. Förster, "Q learning based adaptive protocol parameters for WSNs," *Journal of Communications and Networks*, vol. 25, no. 1, pp. 76–87, 2022, doi: 10.23919/JCN.2022.000035.

[20] Q. Gang, et al., "A Q-learning-based approach to design an energy-efficient MAC protocol for UWSNs through collision avoidance," *Electronics*, vol. 13, no. 22, pp. 1–33, Art. no. 3140, 2024, doi: 10.3390/electronics13224388.

[21] M. U. Rehman, I. Uddin, M. Adnan, A. Tariq, and S. Malik, "VTA-SMAC: Variable traffic-adaptive duty cycled sensor MAC protocol to enhance overall QoS of S-MAC protocol," *IEEE Access*, vol. 9, pp. 33030–33040, 2021, doi: 10.1109/ACCESS.2021.3061357.

[22] J. D. Abdulai, A. A. Amengu, F. A. Katsriku, and K. S. Adu-Manu, "CBU-SMAC: An energy-efficient CLUSTER-BASED UNIFIED SMAC algorithm for wireless sensor networks," *Journal of Ambient Intelligence & Humanized Computing*, vol. 15, no. 4, pp. 2073–2092, Apr. 2024, doi: 10.1007/s12652-023-04737-z.

[23] Z. Ahmed, M. M. Rehan, O. Chughtai, and M. W. Rehan, "AD-RDC: A novel adaptive dynamic radio duty cycle mechanism for low-power IoT devices," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13376–13389, 2022, doi: 10.1109/JIOT.2022.3145017.

[24] G. Kim, J. G. Kang, and M. Rim, "Dynamic duty-cycle MAC protocol for IoT environments and wireless sensor networks," *Energies*, vol. 12, no. 21, pp. 1–13, Nov. 2019, doi: 10.3390/en12214069.

[25] S. Sarang, G. M. Stojanović, S. Stankovski, Ž. Trpovski, and M. Drieberg, "Energy-efficient asynchronous QoS MAC protocol for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2020, no. 1, pp. 1–13, 2020, doi: 10.1155/2020/8860371.

[26] B. A. Muzakkari, M. A. Mohamed, M. F. A. Kadir, and M. Mamat, "Queue and priority-aware adaptive duty cycle scheme for energy efficient wireless sensor networks," *IEEE Access*, vol. 8, pp. 17231–17242, 2020, doi: 10.1109/ACCESS.2020.2968121.

[27] A. N. El-Shenhabi, E. H. Abdelhay, M. A. Mohamed, and I. F. Moawad, "A reinforcement learning-based dynamic clustering sleep scheduling algorithm (RLDCSSA-CDG) for compressive data gathering in wireless sensor networks," *Technologies*, vol. 13, no. 1, pp. 1–25, 2025, doi: 10.3390/technologies13010025.

[28] X. Wang, H. Chen, and S. Li, "A reinforcement learning-based sleep scheduling algorithm for compressive data gathering in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2023, no. 1, pp. 1–17, 2023, doi: 10.1186/s13638-023-02237-4.

[29] H. Dutta, A. K. Bhuyan, and S. Biswas, "Reinforcement learning based flow and energy management in resource-constrained wireless networks," *Computer Communications*, vol. 202, pp. 73–86, 2023, doi: 10.1016/j.comcom.2023.02.011.

[30] F. Afroz and R. Braun, "Empirical analysis of extended QX-MAC for IoT-based WSNS," *Electronics*, vol. 11, no. 16, pp. 1–28, Art. no. 2543, Aug. 2022, doi: 10.3390/electronics11162543.

[31] H. Y. Huang, K. T. Kim, and H. Y. Youn, "Determining node duty cycle using Q-learning and linear regression for WSN," *Frontiers of Computer Science*, vol. 15, no. 1, pp. 1–7, Art. no. 151101, 2021, doi: 10.1007/s11704-020-9153-6.

[32] H. Y. Huang, T. J. Lee, and H. Y. Youn, "Event-driven duty cycling with reinforcement learning and Monte Carlo technique for wireless network," *Mobile Information Systems*, vol. 2021, pp. 1–12, Art. no. 6644389, 2021, doi: 10.1155/2021/6644389.

[33] P. S. Banerjee, S. N. Mandal, D. De, and B. Maiti, "RL-sleep: Temperature adaptive sleep scheduling using reinforcement learning for sustainable connectivity in wireless sensor networks," *Sustainable Computing: Informatics and Systems*, vol. 26, pp. 1–18, 2020, doi: 10.1016/j.suscom.2020.100380.

[34] B. N. Trinh, L. Murphy, and G. M. Muntean, "A reinforcement learning-based duty cycle adjustment technique in wireless multimedia sensor networks," *IEEE Access*, vol. 8, pp. 58774–58787, 2020, doi: 10.1109/ACCESS.2020.2982590.

[35] C. Savaglio, P. Pace, G. Aloi, A. Liotta, and G. Fortino, "Lightweight reinforcement learning for energy efficient communications in wireless sensor networks," *IEEE Access*, vol. 7, pp. 29355–29364, 2019, doi: 10.1109/ACCESS.2019.2902371.

[36] S. Sarang, G. M. Stojanović, M. Drieberg, S. Stankovski, K. Bingi, and V. Jeoti, "Machine learning prediction based adaptive duty cycle MAC protocol for solar energy harvesting wireless sensor networks," *IEEE Access*, vol. 11, pp. 17536–17554, 2023, doi: 10.1109/ACCESS.2023.3246108.

[37] M. Pundir and J. K. Sandhu, "A systematic review of quality of service in wireless sensor networks using machine learning: Recent trend and future vision," *Journal of Network and Computer Applications*, vol. 188, pp. 1–33, 2021, doi: 10.1016/j.jnca.2021.103084.

[38] N. Es-Sakali, Z. Zoubir, S. I. Kaitouni, M. O. Mghazli, M. Cherkaoui, and J. Pfafferott, "Advanced predictive maintenance and fault diagnosis strategy for enhanced HVAC efficiency in buildings," *Applied Thermal Engineering*, vol. 254, pp. 1–17, 2024, doi: 10.1016/j.applthermaleng.2024.123910.

[39] S. Rattal, A. Badri, M. Moughit, E. M. Ar-Reyouchi, and K. Ghoumid, "AI-driven optimization of low-energy IoT protocols for scalable and efficient smart healthcare systems," *IEEE Access*, vol. 13, pp. 48401–48415, 2025, doi: 10.1109/ACCESS.2025.3551224.

[40] K. Medani, C. Gherbi, H. Mabed, and Z. Aliouat, "Energy-efficient Q-learning-based path planning for UAV-aided data collection in agricultural WSNs," *Internet of Things*, vol. 33, pp. 1–17, 2025, doi: 10.1016/j.iot.2025.101698.

[41] M. A. Ridwan, N. A. M. Radzi, F. Abdullah, and Y. E. Jalil, "Applications of machine learning in networking: A survey of current

issues and future challenges," *IEEE Access*, vol. 9, pp. 52523–52556, 2021, doi: 10.1109/ACCESS.2021.3069210.

[42] G. Obaido, et al., "Supervised machine learning in drug discovery and development: Algorithms, applications, challenges, and prospects," *Machine Learning with Applications*, vol. 17, pp. 1–20, 2024, doi: 10.1016/j.mlwa.2024.100576.

[43] H. Sharma, A. Haque, and F. Blaabjerg, "Machine learning in wireless sensor networks for smart cities: A survey," *Electronics*, vol. 10, no. 9, pp. 1–22, Art. no. 1052, 2021, doi: 10.3390/electronics10091012.

[44] S. Fotopoulou, "A review of unsupervised learning in astronomy," *Astronomy and Computing*, vol. 48, pp. 1–23, 2024, doi: 10.1016/j.ascom.2024.100851.

[45] K. Hu, et al., "A review of research on reinforcement learning algorithms for multi-agents," *Neurocomputing*, vol. 599, pp. 1–33, 2024, doi: 10.1016/j.neucom.2024.128068.

[46] M. N. Al-Hamadani, M. A. Fadhel, L. Alzubaidi, and B. Harangi, "Reinforcement learning algorithms and applications in healthcare and robotics: A comprehensive and systematic review," *Sensors*, vol. 24, no. 8, pp. 1–25, Art. no. 2357, 2024, doi: 10.3390/s24082461.

[47] C. M. Wu, Y. C. Kao, K. F. Chang, C. T. Tsai, and C. C. Hou, "A Q-learning-based adaptive MAC protocol for Internet of Things networks," *IEEE Access*, vol. 9, pp. 128905–128918, 2021, doi: 10.1109/ACCESS.2021.3103718.

[48] 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, doi: 10.1016/j.comcom.2021.07.014.

[49] E. Dvir, M. Shifrin, and O. Gurewitz, "Cooperative multi-agent reinforcement learning for data gathering in energy-harvesting wireless sensor networks," *Mathematics*, vol. 12, no. 13, pp. 1–34, Art. no. 2102, 2024, doi: 10.3390/math12132102.

[50] T. Perić, M. Zajc, and M. Beko, "TinyML: Machine learning on ultra-low-power microcontrollers," *Future Internet*, vol. 14, no. 12, pp. 1–18, Art. no. 363, 2022, doi: 10.3390/fi14120363.

[51] K. A. Ngo, T. T. Huynh, and D. T. Huynh, ''Simulation wireless sensor networks in Castalia,'' in *Proceedings of the 2018 International Conference on Intelligent Information Technology (ICIIT)*, Hanoi, Vietnam, Feb. 2018, pp. 39–44.

[52] A. Varga. OMNeT++: Discrete Event Simulator. [Online]. Available: https://omnetpp.org

[53] M. A. Alharbi, M. Kolberg, and M. Zeeshan, "Towards improved clustering and routing protocol for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, pp. 1–31, Feb. 2021, doi: 10.1186/s13638-021-01911-9.

[54] B. Zeng, S. Li, and X. Gao, "Threshold-driven K-means sector clustering algorithm for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2024, no. 68, pp. 1–19, Sep. 2024, doi: 10.1186/s13638-024-02403-2.

[55] A. Sohoub, S. Sati, and M. Eshtawie, "Optimal cluster size for wireless sensor networks," *International Journal of Wireless and Microwave Technologies*, vol. 13, no. 1, pp. 36–49, Jan. 2023, doi: 10.5815/ijwmt.2023.01.04.

[56] 3rd Generation Partnership Project (3GPP), *Study on Ambient Power-Enabled Internet of Things*, 3GPP TR 22.840, Tech. Rep., 2023. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/22_series/22.840/22840-120.zip

[57] M. S. Es-haghi, C. Anitescu, and T. Rabczuk, "Methods for enabling real-time analysis in digital twins: A literature review," *Computers & Structures*, vol. 297, Art. no. 107342, 2024, doi: 10.1016/j.compstruc.2024.107342.

[58] A. K. Bapatla, S. P. Mohanty, and E. Kougianos, "sFarm: A distributed ledger based remote crop monitoring system for smart farming," in *Internet of Things. Technology and Applications*, L. M. Camarinha-Matos, G. Heijenk, S. Katkoori, and L. Strous, Eds., IFIP Advances in Information and Communication Technology, vol. 641. Cham, Switzerland: Springer, 2022, pp. 17–27, doi: 10.1007/978-3-030-96466-5_2.

[59] M. Arif, J. A. Maya, N. Anandan, D. A. Perez, M. A. Tonellotto, H. Zangl, and B. Rinner, "Resource-efficient ubiquitous sensor networks for smart agriculture: A survey," *IEEE Access*, vol. 12, pp. 193332–193364, Dec. 2024, doi: 10.1109/ACCESS.2024.3516814.

[60] N. Senadhira, S. Durrani, S. A. Alvi, N. Yang, and X. Zhou, "UAV-assisted IoT monitoring network: Adaptive multiuser access for low-latency and high-reliability under bursty traffic," *IEEE Transactions on Communications*, early access, Jan. 2024, doi: 10.1109/TCOMM.2024.3516503.

[61] S. Singh, U. Singh, N. Mittal, and F. Gared, "A self adaptive attraction and repulsion based naked mole rat algorithm for energy efficient mobile wireless sensor networks," *Scientific Reports*, vol. 14, Art. no. 1040, Jan. 2024, doi: 10.1038/s41598-024-51218-0.

[62] K. Xu, et al., "Q-learning and efficient low quantity charge method for nodes to extend the lifetime of wireless sensor networks," *Electronics*, vol. 12, no. 22, pp. 1–16, Art. no. 4676, 2023, doi: 10.3390/electronics12224676.

**SHAH ABDUL LATIF** received the B.Eng. degree in telecommunication engineering from Hamdard University, Karachi, Pakistan, in 2014, the M.Eng. degree in electronic engineering from NED University of Engineering and Technology, Karachi, Pakistan, in 2017. He is currently pursuing a Ph.D. degree with the Department of Electrical and Electronics Engineering, Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia. Previously, he has over five years of teaching experience in the field of Electrical and Electronic Engineering at the undergraduate level. His research interests include machine learning, medium access control protocol, energy harvesting communications, wireless sensor networks, and Internet of Things.

**MICHEAL DRIEBERG** (Member, IEEE) received the B.Eng. degree from Universiti Sains Malaysia, Penang, Malaysia, in 2001, the M.Sc. degree from Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia, in 2005, and the Ph.D. degree from Victoria University, Melbourne, Australia, in 2011, all in electrical and electronics engineering. He is currently a Senior Lecturer with the Department of Electrical and Electronics Engineering, Universiti Teknologi PETRONAS. His research interests include radio resource management, medium access control protocols, energy harvesting communications, and performance analysis for wireless and sensor networks. Dr. Drieberg has published and served as a reviewer for several high impact journals and flagship conferences. He has also made several contributions to the wireless broadband standards group.

**SOHAIL SARANG** (Member, IEEE) received the B.Eng. degree in telecommunication engineering from Hamdard University, Karachi, Pakistan, in 2014, the M.Sc. degree in electrical and electronics engineering from Universiti Teknologi PETRONAS (UTP), Malaysia, in 2018, and the Ph.D. degree in electrical and computer engineering from the Faculty of Technical Sciences, University of Novi Sad (UNS), Serbia. Currently, he is a Postdoctoral Researcher at the Department of Electrical Engineering, Faculty of Technical Sciences, UNS. His research interests include energy harvesting communications, low-power sensor networks, battery-free IoT, MAC protocols, and machine learning-driven communication algorithms and protocols.

**Azrina Abd Aziz** (Senior Member, IEEE) received her B.Eng. (Hons.) degree in Electrical and Electronic Engineering from The University of Queensland, Australia, in 1997, the M.Sc. degree in System-Level Integration from the Institute for System Level Integration (ISLI), Scotland, in 2003, and the Ph.D. degree in Computer Systems Engineering from Monash University, Melbourne, Australia, in 2013. She is currently a Senior Lecturer in the Department of Electrical and Electronic Engineering at Universiti Teknologi PETRONAS (UTP), Malaysia. Before entering academia, Dr. Azrina gained industry experience in integrated circuit (IC) packaging, specifically in the final visual inspection process. Her research interests include energy-efficient topology control techniques for wireless sensor networks (WSNs), wireless body area networks (WBANs) for biomedical applications and medical imaging, and the application of machine learning in these domains. Dr. Azrina is a registered member of the Board of Engineers Malaysia and a registered Professional Technologist and is actively involved in professional communities as a Vice Chair of the IEEE Robotics and Automation Society, Malaysia and a member of IEEE Women in Engineering Malaysia.

**RIZWAN AHMAD** (Member, IEEE) received the M.Sc. degree in communication engineering and media technology from the University of Stuttgart, Stuttgart, Germany, in 2004, and the Ph.D. degree in electrical engineering from Victoria University, Melbourne, VIC, Australia, in 2010. From 2010 to 2012, he was a Postdoctoral Research Fellow with Qatar University, Doha, Qatar, on a QNRF Grant. He is currently working as a Professor at the School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad, Pakistan. He also serves as director of the Communication Systems and Networking (CSN) lab at SEECS – NUST. He has authored over 100 journal articles and conference papers. His research interests include public safety networks, medium access control protocols, spectrum and energy efficiency, energy harvesting, and the performance analysis for wireless communication and networks. He was a recipient of the prestigious International Postgraduate Research Scholarship from the Australian Government. He served as a reviewer for IEEE journals and conferences.

**GORAN M. STOJANOVIĆ** (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the Faculty of Technical Sciences (FTS), University of NoviSad (UNS), Serbia, in 1996, 2003, and 2005, respectively. He is currently a Full Professor at FTS, UNS. He has 27 years of experience in research and development. He has more than18 years of experience in writing, implementation, and coordination of EU-funded projects (Horizon Europe, H2020, EUREKA, ERASMUS, CEI), with a total budget exceeding 22.86 MEUR. He was a supervisor of 14 Ph.D. students, 40 M.Sc. students, and 60 diploma students at FTS-UNS. He is an author/co-author of 280 articles, including 180 peer-reviewed journals with impact factors, five books, three patents, and two chapters in monograph. His research interests include sensors, flexible electronics, textile electronics, edible electronics, and microfluidics. He was a keynote speaker at 14 international conferences.