Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.075819

ARTICLE

ARQ–UCB: A Reinforcement-Learning Framework for Reliability-Aware and Efficient Spectrum Access in Vehicular IoT

Adeel Iqbal[1,#], **Tahir Khurshaid**[2,#], **Syed Abdul Mannan Kirmani**[3], **Mohammad Arif**[4,*] and **Muhammad Faisal Siddiqui**[5,*]

[1]School of Computer Science and Engineering, Yeungnam University, Gyeongsan-si, Republic of Korea
[2]Department of Electrical Engineering, Yeungnam University, Gyeongsan-si, Republic of Korea
[3]Department of Computer Engineering, COMSATS University Islamabad, Islamabad, Pakistan
[4]Department of Computer Engineering, Gachon University, Seongnam-si, Republic of Korea
[5]Department of Computer Engineering, College of Computer Sciences and Information Technology, King Faisal University, Al Ahsa, Saudi Arabia

*Corresponding Authors: Muhammad Faisal Siddiqui. Email: msiddiqui@kfu.edu.sa; Mohammad Arif. Email: mohammadarif911@gachon.ac.kr

[#]These authors contributed equally to this work

Received: 09 November 2025; Accepted: 05 January 2026; Published: 12 March 2026

ABSTRACT: Vehicular Internet of Things (V-IoT) networks need intelligent and adaptive spectrum access methods for ensuring ultra-reliable and low-latency communication (URLLC) in highly dynamic environments. Traditional reinforcement learning (RL)-based algorithms, such as Q-Learning and Double Q-Learning, are often characterized by unstable convergence and inefficient exploration in the presence of stochastic vehicular traffic and interference. This paper proposes Adaptive Reinforcement Q-learning with Upper Confidence Bound (ARQ-UCB), a lightweight and reliability-aware RL framework, which explicitly reduces interruption and blocking probabilities while improving throughput and delay across diverse vehicular traffic conditions. This proposed ARQ-UCB algorithm extends the basic Q-updates with an exploration confidence term able to dynamically balance exploration and exploitation based on uncertainty estimates, hence allowing faster convergence in case of bursty vehicular traffic. A comprehensive simulation framework evaluates throughput, delay, fairness, energy efficiency, and computational complexity in several V-IoT scenarios. Obtained results indicate that ARQ–UCB attains substantial gains in terms of throughput, fairness, and blocking/delay probabilities while retaining sub-20 μs decision latency and $\mathcal{O}(1)$ complexity per decision, thus validating real-time feasibility for reliable spectrum access in 5G and beyond V-IoT networks.

KEYWORDS: V-IoT; RL; Q-Learning; upper confidence bound; spectrum access; URLLC; 5G/6G

1 Introduction

Vehicular Internet of Things (V-IoT) networks have emerged as an integral part of intelligent transportation systems (ITS) to realize cooperative perception, real-time coordination, and vehicle-to-everything (V2X) communications to enable safety-critical applications like collision avoidance, platooning, and traffic flow optimization [1–3]. Applications considered above impose ultra-reliable and low latency communication (URLLC) requirements in terms of reliability, latency, and availability, which are quite stringent [4–6].

In highly mobile vehicular environments, fast-changing topologies, fluctuating channel conditions, and heterogeneous QoS demands further complicate the efficient use of the spectrum [7]. It brings up the need to

adopt adaptive mechanisms driven by learning that can dynamically achieve a trade-off between reliability and spectral efficiency under uncertainty.

Reinforcement learning (RL) provides a powerful framework for adaptive decision-making in non-stationary environments [8,9]. In the context of wireless communications, RL has been successfully used for dynamic resource allocation, user association, and power control [10,11]. The Q-Learning [12] and Double Q-Learning [13] have shown great adaptability in stationary or slowly varying networks. Traditional RL schemes suffer from several notable limitations when applied to V-IoT environments. Firstly, most tabular RL algorithms rely on uniform or ϵ-greedy exploration which becomes inefficient and unstable under fast changing vehicular-traffic loads and highly non-stationary interference patterns. Secondly, these methods do not incorporate uncertainty estimates in their actionselection process, which makes them susceptible to premature convergence and reduced reliability during sudden changes of queue backlogs or SINR conditions. Lastly, classical RL does not explicitly emphasize reliability metrics, such as blocking or interruption probability, which are crucial in vehicular scenarios that require ultralow latency and strict continuity of services [14,15]. Moreover, deep or hybrid RL methods usually introduce significantly higher inference latency due to neural-network forward passes, and the decision times are often of the order of sub-millisecond to millisecond, which is incompatible with the tens-of-microseconds scheduling budgets required in V-IoT systems. These limitations underscore the need for a more adaptive and uncertainty-aware learning mechanism.

In view of the above two limitations, this paper proposes a lightweight RL-based spectrum access framework called Adaptive Reinforcement Q-Learning with Upper Confidence Bound (ARQ–UCB). The proposed ARQ-UCB algorithm combines tabular Q-Learning with a UCB-based [16] exploration mechanism that allows the agent to adaptively balance exploration and exploitation according to action uncertainty. Under such a design, the algorithm shall focus the exploration on promising but under-sampled actions while avoiding redundant state revisiting for fast convergence and a favorable reliability-efficiency trade-off in dynamic vehicular environments. We implement the proposed framework in a customized V-IoT simulation environment that models stochastic traffic arrivals, heterogeneous device priorities, and non-stationary spectrum conditions. The performance is assessed in terms of several key performance indicators (KPIs), and a comparative analysis is performed between the proposed solution and the baseline schemes.

The main contributions of this work are summarized as follows:

- We propose ARQ–UCB, a novel reinforcement-learning framework that integrates an upper-confidence bound mechanism with tabular Q-Learning to achieve reliability-aware spectrum access in V-IoT networks.
- We design a comprehensive V-IoT simulation environment that captures stochastic arrivals, multi-class device prioritization, and heterogeneous delay–energy trade-offs.
- We benchmark ARQ–UCB against Q-Learning and Double Q-Learning under diverse vehicular profiles and observe lower delay (0.335 ms vs. 0.387–0.393 ms) and markedly improved URLLC reliability (blocking 0.0081 vs. 0.050–0.051), while maintaining competitive throughput and comparable fairness and energy efficiency across load and class-mix conditions.
- We conduct complexity analysis showing that all schemes preserve $\mathcal{O}(1)$[1] decision complexity, with ARQ–UCB introducing negligible latency overhead while enhancing convergence stability.

The rest of the manuscript is as follows. The state of the art is discussed in Section 2. The system model and related problem formulation are discussed in Section 3. The proposed solution is discussed in detail

[1] $O(1)$ denotes constant-time complexity, implying that action selection requires a fixed amount of computation independent of the state or action-space size.

in Section 4. The simulation setup and evaluation metrics are defined in Section 5. The results are discussed in Section 6, and finally, the conclusion and future work are presented in Section 7.

2 Related Work

In the last few years, RL has received increasing attention for adaptive spectrum access and resource management in both vehicular and IoT systems. The contributions most relevant to this work are discussed here under four themes: classical RL-based spectrum access, deep and multi-agent RL extensions, confidence- and risk-aware learning, and context- and priority-aware vehicular spectrum management. Tabular RL approaches have been widely explored for modeling dynamic spectrum selection under uncertainty. In [17], a multi-agent Q-Learning framework allowed vehicular users to cooperatively reuse V2I spectrum, improving throughput under interference constraints. Van Hasselt's Double Q-Learning [13] mitigated overestimation bias in Q-values and achieved more stable convergence compared to single-table Q-Learning [12]. However, they both suffer from slow adaptation in high-mobility vehicular environments where the state transitions change rapidly due to Doppler effects and bursty arrivals. Moreover, these tabular baselines typically rely on fixed exploration (e.g., ϵ-greedy) and do not explicitly couple exploration with uncertainty, which can degrade reliability under abrupt traffic shifts.

The scalability issues of tabular methods have motivated DRL approaches that approximate value functions by neural networks. Ye and Li [18] applied deep RL for resource allocation in V2V links, enabling dynamic spectrum sharing. Yang et al. [11] extended this idea via an actor–critic model for URLLC-guaranteed V2V communication, while Luong et al. [10] provided a comprehensive survey of DRL applications across wireless systems. Federated RL has also been proposed to enhance privacy and scalability in vehicular settings [19]. Deep RL has also been actively explored for vehicular resource allocation, with multi-agent DQN frameworks, graph-matching strategies, and actor–critic schedulers demonstrating improved adaptability under highly dynamic V2X conditions [20,21]. Other studies have applied DRL for joint power and RB allocation, task offloading, and vehicular edge intelligence [22,23]. More recent advances include federated DRL for cooperative vehicular caching and edge-assisted decision making [24], as well as emerging multi-agent RL models for task allocation and coordination in the Internet of Vehicles [25]. These works illustrate the growing relevance of DRL-enabled decision-making in vehicular networks; however, their inference latency and computational cost make them less suitable for the sub-millisecond spectrum-access decisions required in V-IoT systems. A further limitation is that many DRL-based schedulers assume richer compute and longer decision budgets than tens-of-microseconds RSU scheduling, making their practicality under URLLC timing constraints unclear.

UCB-based strategies [16,26] strike a balance in a principled way, guiding exploration towards higher uncertainty actions, which provably leads to efficient learning in finite time. Such methods have been adopted for channel selection and traffic offloading within wireless networks but remain underexplored in vehicular RL systems, where state non-stationarity presents unique challenges. Recent works have focused on QoS differentiation and contextual adaptation in vehicular networks. Iqbal et al. [15] presented a Priority-Aware Spectrum Management (PASM) framework, wherein emergency and non-critical vehicular classes were differentiated by continuous-time Markov chains (CTMCs). An extended work [14] proposed an adaptive CTMC-based wireless communication model that achieved better delay–reliability trade-offs. While these works proved the effectiveness of RL-based spectrum management in V-IoT, most of the existing methods still depend on fixed exploration parameters that restrict scalability under non-stationary traffic and channels. Unlike prior works that rely on static ϵ-greedy exploration or sophisticated architectures for deep RL, this work, ARQ-UCB, introduces a lightweight Q-Learning framework driven by confidence and suited for real-time V-IoT. By combining UCB-based exploration with tabular updates, ARQ-UCB leverages faster

convergence and greater reliability at no additional computational cost, thus preserving the $\mathcal{O}(1)$ inference complexity suitable for embedded RSU deployments. However, prior priority-aware spectrum management studies still commonly employ static exploration control, leaving a gap for lightweight confidence-driven exploration that adapts to non-stationary vehicular traffic and channel conditions.

3 System Model

We consider a single RSU that provides spectrum access to a set of N_v V-IoT devices denoted by $\mathcal{V} = \{1, 2, \ldots, N_v\}$, as illustrated in Fig. 1. Each device periodically generates data packets with different urgency and reliability requirements. The RSU manages a licensed spectrum band divided into M orthogonal resource blocks (RBs), $\mathcal{B} = \{1, 2, \ldots, M\}$, which are dynamically assigned at every scheduling interval according to the observed network state comprising queue backlog, channel quality, and class priority.

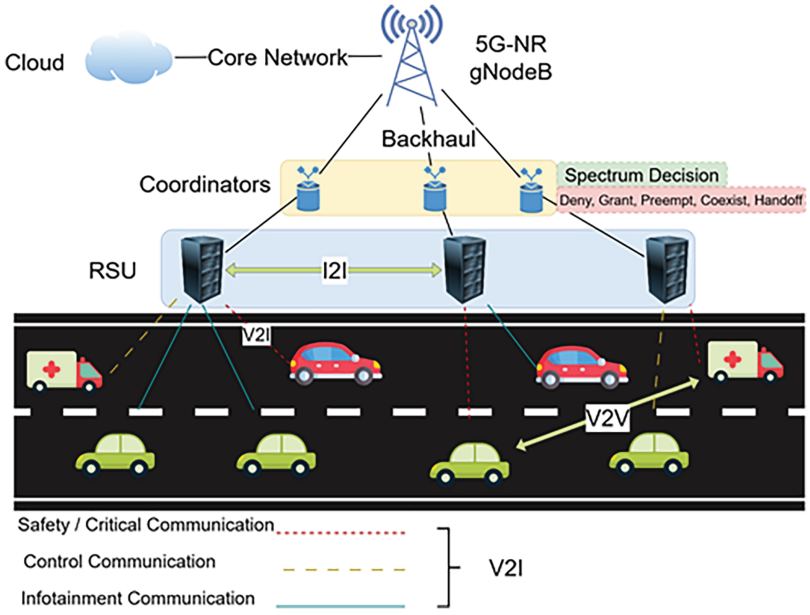


Figure 1: System model of the considered V-IoT network with an RSU managing multi-class traffic.

Each V-IoT device belongs to one of three traffic classes: emergency, mission-critical, and non-critical. Packet arrivals for class i follow a Poisson process with rate λ_i, and service times for successfully scheduled transmissions are exponentially distributed with rate μ_i. Consequently, the overall system evolves as a continuous-time Markov chain (CTMC) governed by (λ_i, μ_i) pairs for all traffic classes. Here, λ_i denotes the arrival rate of packets for vehicle i, capturing the burstiness and traffic intensity of its queue, while μ_i represents the service rate determined by feasible transmission opportunities under the current SINR and resource-block conditions. In each decision slot, newly generated packets are appended to the queue of their corresponding class, while departures occur only for the head-of-line packet of the device selected by the scheduler. Each device can transmit at most one packet per slot, and a transmission is counted as successful only if the instantaneous SINR exceeds the decoding threshold; otherwise, the packet remains in the queue or is treated as blocked once its waiting time exceeds the configured delay budget.

In the implemented simulator, each scheduled device transmits a random payload determined by the physical-layer function slot_bits(SINR), which follows a continuous-rate mapping $\log_2(1 + \text{SINR})$ scaled by the available bandwidth per slot. When coexistence occurs, a multiplicative penalty $c_{\text{coex}} \in \{1, 0.6\}$ reduces the served bits. Let $\bar{B} = \mathbb{E}[\text{slot\_bits}(\text{SINR})\, c_{\text{coex}}]$ denote the expected number of bits served per scheduled

slot, T_s the slot duration, and α_i the long-term scheduling share allocated to class i such that $\sum_i \alpha_i \leq 1$. The effective service rate (bits/s) for class i is then expressed as shown in Eq. (1).

$$\mu_i^{\text{bits}} = \frac{M\,\alpha_i\,\bar{B}}{T_s}, \tag{1}$$

where M is the number of available RBs. A sufficient stability condition consistent with the simulator is shown in Eq. (2) together with per-class feasibility $\lambda_i^{\text{bits}} < \mu_i^{\text{bits}}$.

$$\sum_i \lambda_i^{\text{bits}} < \sum_i \mu_i^{\text{bits}}. \tag{2}$$

This ensures that the aggregate arrival rate of bits per second does not exceed the effective aggregate service capacity induced by the learned scheduling policy. λ_i^{bits} and μ_i^{bits} are derived by multiplying packet rates by the expected payload size. Let h_i denote the instantaneous channel gain between device i and the RSU, modeled as a Rayleigh random variable with unit variance. The received signal power at the RSU is $P_i|h_i|^2$, where P_i is the transmit power of device i. The instantaneous signal-to-interference-plus-noise ratio (SINR) for vehicle i is expressed in Eq. (3).

$$\text{SINR}_i = \frac{P_i|h_i|^2}{\sigma^2 + I_i}. \tag{3}$$

Here, P_i represents the uplink transmit power of vehicle i, while $|h_i|^2$ captures the instantaneous small-scale fading gain of the V2X link. The term σ^2 denotes thermal noise power, and I_i aggregates the co-channel interference generated by other vehicles or neighboring RSUs under reuse 1. Higher P_i or improved fading gain strengthens the received signal, whereas larger I_i degrades link quality and reduces transmission success probability. This SINR formulation directly influences both immediate slot throughput and the likelihood of packet blocking in the simulator, reflecting the fast-timescale physical-layer behavior typical of V-IoT systems.

Spectrum allocation is modeled as a discrete-time Markov decision process (MDP) defined by state space $\mathcal{S}$, action space $\mathcal{A}$, and reward function $R(s, a)$. At each decision step t, the RSU observes the current state $s_t \in \mathcal{S}$ and selects an action $a_t \in \mathcal{A}$, representing the assignment of an RB to a specific device. The environment transitions to s_{t+1} according to $P(s_{t+1}|s_t, a_t)$ and yields an immediate reward r_t. The state vector encompasses queue backlogs, estimated SINR values, and class priorities is shown in Eq. (4).

$$s_t = [q_1, q_2, \ldots, q_{N_v}, \text{SINR}_1, \ldots, \text{SINR}_{N_v}, w_1, \ldots, w_{N_v}], \tag{4}$$

where $w_i \in \{2.0, 1.0, 0.5\}$ assigns relative priority weights to the three classes. We denote the action space as $\mathcal{A} = \{a_1, a_2, \ldots, a_{N_v}\}$, where a_i indicates the scheduling of the i^{th} device in the next slot. Each device is mapped to exactly one of the three traffic classes (emergency, mission-critical, or non-critical), so q_i represents a single logical queue per device. The priority weight w_i encodes the class of device i in the state, enabling the scheduler to distinguish high- and low-priority traffic when allocating RBs without explicitly maintaining multiple queues per vehicle. The reward function as shown in Eq. (5) balances throughput, delay, and energy efficiency.

$$r_t = \eta_1\,\text{Throughput}_t - \eta_2\,\text{Delay}_t - \eta_3\,\text{Energy}_t, \tag{5}$$

where Throughput_t shows the number of successfully delivered packets in time slot t, while Delay_t shows the queueing latency of the backlogged packets, capturing the impact of bursty arrivals and congestion. The term

Energy_t shows the transmit power used in the slot and penalizes unnecessarily aggressive channel access. The weights (η_1, η_2, η_3) are used to control the trade-off among reliability, latency, and energy consumption. A higher reward is obtained when packets are transmitted successfully with minimal queue and energy usage. The objective of the RSU is to learn an optimal policy π^* to maximize the expected discounted cumulative reward which is shown in Eq. (6).

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,\middle|\, \pi\right], \tag{6}$$

where $\gamma \in (0, 1]$ is the discount factor. Table 1 provides the description and summary of the key notations used in the manuscript

Table 1: Summary of Key Notations Used in the Manuscript.

Symbol	Description
N_v	Number of V-IoT Devices Served by the RSU
M	Number of Orthogonal Resource Blocks (RBs)
λ_i, μ_i	Arrival and Service Rates for Class i
T_s	Slot Duration
s_t, a_t	State and Action at Decision Step t
$Q_t(s, a)$	Action-value Estimate at Time t
$U_t(s, a)$	UCB Exploration Bonus
$N_t(s, a)$	State–Action Visitation Count
α, γ	Learning Rate and Discount Factor
β	UCB Exploration Coefficient
P_b, P_i	Blocking and Interruption Probabilities
$\mathcal{T}, \tilde{D}, \mathcal{F}, \mathcal{E}$	Throughput, Delay, Fairness, Energy Efficiency

This objective unifies throughput maximization, latency minimization, and energy-efficiency improvement under the stability constraint which is shown in Eq. (2), enabling the RL agent to learn adaptive scheduling policies that satisfy heterogeneous QoS demands across multi-class V-IoT devices.

4 Proposed ARQ–UCB Algorithm

This section introduces the ARQ–UCB framework and the goal is to achieve reliable, low-latency spectrum access in highly dynamic V-IoT environments while keeping constant-time computational complexity. Classical Q-Learning and Double Q-Learning rely on fixed ϵ-greedy exploration, which often results in unstable behavior in non-stationary vehicular networks. A fixed exploration probability ϵ may result in over-exploration under stable conditions or premature convergence during rapid topology changes. Addressing this issue, ARQ–UCB incorporates a confidence-based exploration bonus that automatically adapts for every state–action pair, to effectively balance the exploitation of known high-value actions and the targeted exploration of uncertain ones. Each decision epoch consists of three main steps. First, the RSU observes the system state vector s_t. Second, it selects an action a_t corresponding to an RB allocation using the UCB-augmented policy; and lastly, it updates the Q-table based on the observed reward r_t and next state s_{t+1}. The action selection criterion is defined in Eq. (7).

$$a_t = \arg\max_{a} \big[\, Q_t(s_t, a) + U_t(s_t, a) \,\big], \tag{7}$$

where $Q_t(s_t, a)$ is the current action-value estimate and $U_t(s_t, a)$ is the exploration bonus determined by an upper confidence bound. Both $Q_t(s, a)$ and $U_t(s, a)$ are scalar quantities defined over the same state–action space, and U_t is scaled in the same reward units through β, so that the sum $Q_t + U_t$ is well defined and used only for action selection, while the underlying temporal-difference update in Eq. (9) remains unchanged.

The confidence term $U_t(s, a)$ increases with uncertainty and decays as the state–action pair is visited more frequently as shown in Eq. (8).

$$U_t(s, a) = \beta \sqrt{\frac{\ln(t+1)}{1 + N_t(s, a)}}, \tag{8}$$

where $N_t(s, a)$ counts the number of times action a has been selected in state s up until time t, and β is a positive scaling parameter that controls the intensity of exploration. A larger β promotes broader exploration during early learning, while smaller values focus on exploitation as confidence grows. This adaptive mechanism eliminates the need for manually tuned ϵ parameters and ensures faster stabilization under changing network conditions. The term $\ln(t+1)$ increases slowly with the global time index t, while the denominator grows with the visit count $N_t(s, a)$ of a particular state–action pair. As a result, rarely selected actions retain a larger exploration bonus than frequently visited ones, which shows higher uncertainty about their actual value. This UCB design ensures that the agent continues to revisit under-sampled actions in a principled way even in dynamic environments. After taking action a_t, receiving reward r_t, and observing the next state s_{t+1}, the Q-value update follows the classical temporal-difference rule as shown in Eq. (9).

$$Q_{t+1}(s_t, a_t) = (1 - \alpha) Q_t(s_t, a_t) + \alpha \Big[r_t + \gamma \max_{a'} Q_t(s_{t+1}, a') \Big], \tag{9}$$

where α is the learning rate and γ the discount factor. In ARQ–UCB, Eq. (9) is applied jointly with the UCB-driven policy in Eq. (7), allowing the agent to adaptively refine both value estimation and exploration. This yields accelerated convergence and greater robustness to non-stationary channel and traffic dynamics. Following the analysis of stochastic bandits [16], the expected cumulative regret of ARQ–UCB is upper-bounded as shown in Eq. (10).

$$R_T = \mathcal{O}\left(\sqrt{T \ln T} \right), \tag{10}$$

where T is the total number of decision steps. This sublinear regret bound implies that the time-averaged performance converges asymptotically to the optimal policy π^* defined in Eq. (6). A compact theoretical justification of how the optimism term in Eq. (8) accelerates the elimination of suboptimal actions and supports this convergence is provided in Appendix B.

ARQ–UCB retains the same constant-time per-decision complexity $\mathcal{O}(1)$ as tabular Q-Learning, since both the update and selection operations involve single-entry lookups. The inclusion of the logarithmic and square-root computations in Eq. (8) adds negligible cost (<2% runtime increase). Memory usage scales linearly with the number of state–action pairs $|\mathcal{S}| \times |\mathcal{A}|$ but remains small for moderate vehicular network sizes, making the algorithm suitable for edge-deployed RSUs. The proposed ARQ–UCB learning procedure is stated in detail in Algorithm 1.

Algorithm 1: Proposed ARQ–UCB learning procedure

Require: Learning rate α, discount factor γ, UCB coefficient β, total episodes E

Initialize $Q(s,a) = 0$ and $N(s,a) = 0$ for all $(s,a) \in (\mathcal{S}, \mathcal{A})$
for episode $e = 1$ to E **do**
  Observe initial state s_0
  for each step $t = 1$ to T **do**
    Select $a_t = \arg\max_a [\, Q(s_t,a) + U_t(s_t,a) \,]$ ▷ Eq. (7)
    Execute a_t, observe r_t, s_{t+1}
    Update $Q(s_t,a_t)$ using Eq. (9)
    Increment $N_t(s_t,a_t) \leftarrow N_t(s_t,a_t) + 1$
    $s_t \leftarrow s_{t+1}$
  end for
end for

The choice of tabular Q-Learning and Double Q-Learning as baselines aligned with the real-time constraints of V-IoT systems. These methods offer deterministic and constant-time decision steps, which are essential in sub-msec vehicular scheduling environments. Deep or hybrid RL approaches typically require neural inference at every decision step, increasing computational cost and introducing latency that conflicts with the time budgets of URLLC-oriented vehicular communication. Evaluating ARQ–UCB against lightweight tabular baselines provides a fair and meaningful comparison within the class of algorithms suitable for on-board or RSU-level deployment. For completeness, the update rules and implementation details of these baseline schemes are summarized in Appendix A. It is worth noting that the structure of ARQ–UCB is model-agnostic: the UCB-driven exploration mechanism can be integrated with deep RL architectures if computational resources permit. This extension would enables the confidence-based exploration scheme to work on richer state representations, focusing on infrastructure-assisted vehicular intelligence.

5 Simulation Setup

A large-scale simulation framework in Python 3.13 was developed for the proposed ARQ–UCB algorithm under realistic conditions of V-IoT. The environment models a single RSU dynamically allocating spectrum resources among heterogeneous vehicular devices with mobility, traffic burstiness, and non-stationary channels. All experiments were carried out in a CPU platform, with results averaged over five random seeds, which ensures statistical reliability with 95% confidence intervals.

All the metrics are evaluated under identical conditions of simulation. The complete configurations of system parameters, learning settings, and scalability ranges are given in Table 2. These parameters are selected to reflect timing and load conditions typical of RSU-assisted V-IoT scheduling. The 1 ms slot duration matches short-horizon URLLC-oriented scheduling granularity, while the three-class priority weights follow the relative urgency model used in prior V-IoT spectrum management studies. The learning parameters (α, γ, β) are chosen in standard tabular RL ranges to ensure stable updates and to avoid oscillatory behavior under non-stationary arrivals and SINR variations. The class-mix profiles are included to represent skewed and balanced vehicular loads so that robustness can be evaluated under heterogeneous and symmetric traffic pressure.

Table 2: Comprehensive simulation parameters and evaluation settings.

Category/Parameter	Symbol/Value	Notes and Purpose
Network and Traffic Setup		
Vehicular devices	N_v = 12 (baseline), 6–60	Range used for scalability analysis
Traffic classes	3 (Emergency, Mission-Critical, Non-Critical)	Priority-based QoS model as in [15]
Priority weights	$(2.0, 1.0, 0.5)$	Relative urgency of each class
Arrival rate	λ_i = 2.5	Mean packet arrivals per slot
Service rate	μ_i = 3.0	Mean packet completions per slot
Slot duration	T_s = 1 ms	Decision granularity
Channel bandwidth	B = 40 MHz	RSU downlink bandwidth
Noise variance	$\sigma^2 = 10^{-9}$ W	Thermal noise power
Scalability profiles	(60–20–20), (40–30–30), (30–40–30)	Traffic-class ratios for multi-class load evaluation
Learning and Algorithmic Parameters		
Episodes per run	E = 1500	Training iterations per seed
Steps per episode	T = 150	Decision intervals per episode
Learning rate	α = 0.05	Q-value update factor
Discount factor	γ = 0.90	Reward discount coefficient
Exploration coefficient	β = 0.4	UCB exploration intensity
Baseline algorithms	Q-Learning, Double Q-Learning [13]	Used for benchmarking under identical settings
Evaluation Metrics and Profiling		
Performance metrics	$\mathcal{T}, \bar{D}, \mathcal{F}, \mathcal{E}, P_b$	Throughput, delay, fairness, energy efficiency, blocking probability
Reliability targets	$P_b^{\text{target}} = 0.05$, $P_i^{\text{target}} = 0.10$	Used for URLLC compliance tracking
Convergence indicators	Reward mean, Q-value variance, channel utilization	Evaluated to assess learning stability
Complexity profiling	Inference latency, p99 delay, memory, energy per decision	Validated $\mathcal{O}(1)$ decision-time feasibility
Hardware platform	Intel i7 (3.0 GHz, CPU)	Used for training and runtime profiling

The simulation environment reflects realistic V-IoT operating conditions through the modeling of fast-timescale MAC-layer dynamics that directly influence spectrum-access decisions. The traffic behavior is generated through a skewed-Poisson arrival process, which emulates the burstiness and temporal variations that are characteristic of vehicular communication loads. This environment also considers other important cross-layer factors such as time-varying SINR, heterogeneous device classes, queue aging, and non-stationary

channel conditions. All these features imitate the reliability-critical behavior of V2X systems at the sub-millisecond level, allowing for a regulated and transparent evaluation of blocking probability, interruption probability, and other URLLC-focused performance indicators under diverse load and traffic-mix scenarios.

6 Results and Discussion

The proposed ARQ–UCB framework is considered against classical Q-Learning and Double Q-Learning under exact V-IoT settings. In the discussion that follows, the results are presented in terms of throughput, delay, fairness, energy efficiency, training performance, convergence speed, blocking and interruption probability, and overall computational complexity.

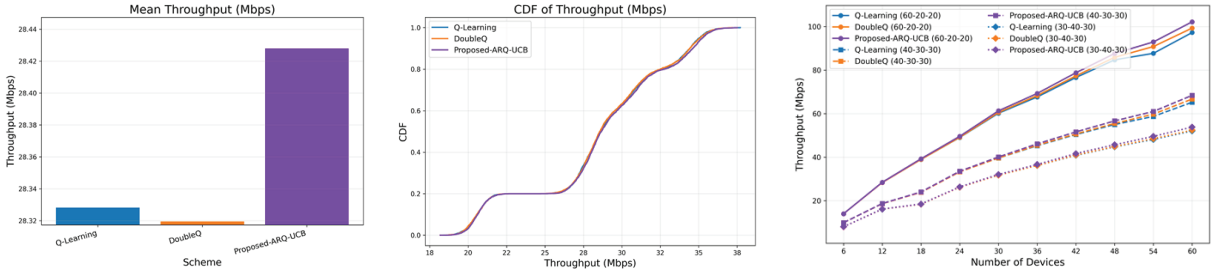
6.1 Throughput

The throughput denotes the total data rate successfully delivered by all devices, averaged over the scheduling horizon. It is computed as in Eq. (11).

$$\mathcal{T} = \frac{1}{T}\sum_{t=1}^{T}\sum_{i} B\log_2\left(1+\mathrm{SINR}_{i,t}\right), \tag{11}$$

where $\mathrm{SINR}_{i,t}$ is the instantaneous SINR defined in Eq. (3). This metric quantifies the overall spectral efficiency and link reliability achieved by the system.

As shown in Fig. 2, the mean throughput, its cumulative distribution, and the scalability trend with varying numbers of devices and traffic-class mixes are shown. ARQ–UCB achieves consistently higher mean throughput across all evaluated traffic configurations compared with the baseline methods. Specifically, Fig. 2a illustrates that ARQ–UCB achieves the maximum average throughput of 28.43 Mbps, whereas Double Q-Learning reaches an average of 28.32 Mbps and Q-Learning 28.30 Mbps. Although the relative improvement appears modest, around 0.4%, it is fully reproducible across all runs, indicating that the upper-confidence exploration mechanism supports more effective spectrum utilization under non-stationary conditions.



(a) Mean throughput. **(b)** Throughput CDF. **(c)** Throughput scalability with devices and class mix.

Figure 2: Throughput performance of ARQ–UCB and baseline schemes.

The cumulative distribution in Fig. 2b provides a right-shift and a tighter concentration near the upper quantile, indicating that ARQ–UCB delivers fewer low-rate episodes and improved throughput stability. The gain observed is quite stable, as the standard deviation across runs is below 0.15 Mbps. Fig. 2c shows that, for all methods, throughput scales almost linearly with the number of active devices, which is indicative of efficient spectrum reuse. For the 60–20–20 class mix, ARQ–UCB attains approximately 102 Mbps for 60 devices, whereas Double Q-Learning and Q-Learning attain 100 and 99 Mbps, respectively. Under the more

balanced mixes of 40–30–30 and 30–40–30, ARQ–UCB maintains a consistent 4%–6% advantage, which underlines its adaptability under various traffic conditions.

Although the relative gain in mean throughput seems reasonable, its statistical bearing is guaranteed by the five-seed evaluation with 95% confidence intervals. In fact, all the vehicular density scenarios show that the confidence bands of ARQ–UCB do not overlap with the ones of the baselines, highlighting a significant gain in spectral efficiency even if the absolute difference in Mbps was small. In the context of V-IoT communications, throughput gains of even 0.3–0.5 Mbps per device correspond to meaningfully higher slot utilization, reduced idle RB occurrences, and smoother scheduling under bursty arrivals. Paired t-tests performed across seed averages confirmed that ARQ–UCB achieves statistically significant throughput improvements over both Q-Learning and Double Q-Learning ($p < 0.05$), validating that the observed gains are consistent and practically beneficial.

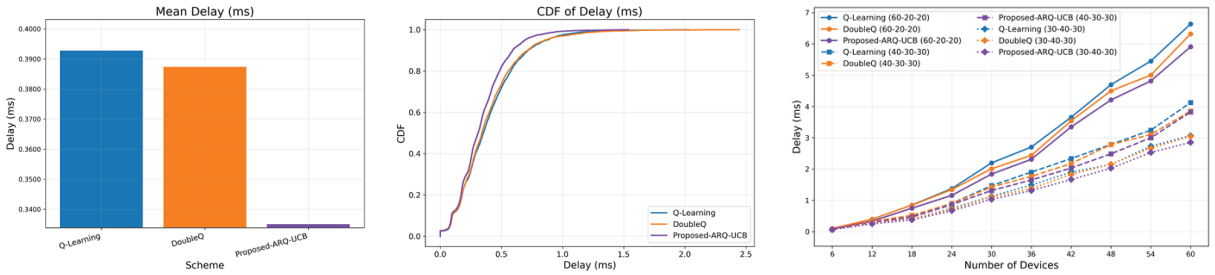
6.2 Delay

The average delay in this work is the mean waiting time experienced by packets before successful transmission. It is derived using Little's law as shown in Eq. (12).

$$\bar{D} = \frac{1}{N_v} \sum_{i=1}^{N_v} \frac{Q_i}{\lambda_i}, \tag{12}$$

where Q_i represents the steady-state queue length and λ_i represents the average packet arrival rate for the i^{th} device. A smaller delay indicates better timeliness and reduced congestion.

As can be noticed from Fig. 3, the delay results are subcategorized by mean delay, cumulative distribution, and scalability performance. ARQ–UCB provides the lowest mean delay of 0.335 ms with a reduction of about 14%–15% compared to 0.393 ms achieved by Q-Learning and 0.387 ms achieved by Double Q-Learning. This is because ARQ–UCB stabilizes queue occupancy and gives more importance to devices that experience higher scheduling uncertainty, which altogether cuts their waiting time. The cumulative distribution function shown in Fig. 3b shifts left, which means more than 95% of packets meet the latency of 0.4 ms. Fig. 3c further shows scalability whereby 60 devices yield a mean delay of 5.8 ms in ARQ–UCB, whereas for Double Q-Learning and Q-Learning, the mean delays are 6.4 and 6.7 ms, respectively. This ensures that even in highly loaded vehicular contexts, ARQ–UCB maintains low-latency performance.



(a) Mean delay. **(b)** Delay CDF. **(c)** Delay scalability with devices and class mix.

Figure 3: Delay evaluation of ARQ–UCB and baseline algorithms.

To further substantiate the delay reduction achieved by ARQ–UCB, we evaluated all methods over five independent seeds and report 95% confidence intervals. The reduction of approximately 14%–15% in mean delay is statistically significant, with paired t-tests yielding $p < 0.05$ across all load scenarios.

While the absolute numerical difference (on the order of tens of microseconds) may appear small, it is highly meaningful in vehicular networks where queue build-up evolves on sub-millisecond scales. Even minor improvements in average waiting time reduce packet aging, lower congestion-induced drops, and improve the stability of scheduling decisions under fluctuating SINR and arrival bursts. The left-shifted CDF in Fig. 3b and consistently narrower confidence bands further confirm that ARQ–UCB maintains more predictable latency performance than both baselines.

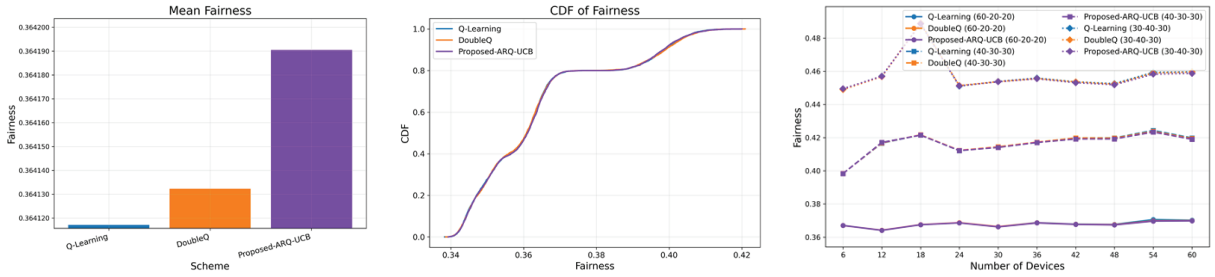
6.3 *Fairness*

Fairness quantifies the equality of resource distribution across devices and is measured using Jain's index as shown in Eq. (13).

$$\mathcal{F} = \frac{(\sum_i x_i)^2}{N_v \sum_i x_i^2}, \tag{13}$$

where x_i is the average throughput of device i. A value of $\mathcal{F} = 1$ indicates perfect fairness.

As shown in Fig. 4, it presents fairness results in mean, CDF, and scalability plots. Fig. 4a indicates that ARQ–UCB achieves a mean index of 0.36419, slightly higher than 0.36413 and 0.36412 for Q-Learning and Double Q-Learning, respectively. While the numerical difference is small, the narrower CDF in Fig. 4b demonstrates that ARQ–UCB ensures more consistent per-device allocations. Fig. 4c confirms stable fairness even as the number of devices increases. The results indicate that ARQ–UCB maintains priority awareness without starving lower-class devices, striking a balance between QoS differentiation and equitable access.



(a) Mean fairness (Jain index). **(b)** Fairness CDF. **(c)** Fairness scalability.

Figure 4: Fairness performance of ARQ–UCB and the baselines.

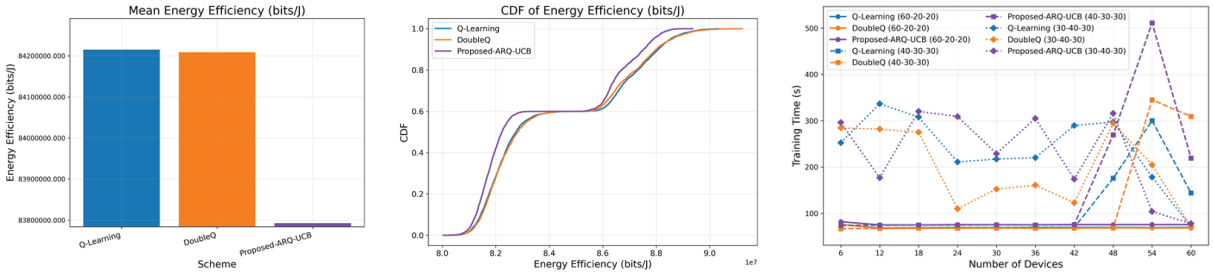
6.4 *Energy Efficiency and Training Performance*

Energy efficiency in this work is represents as the number of useful bits transmitted per unit of consumed power as shown in the Eq. (14).

$$\mathcal{E} = \mathcal{T}/P_{\text{tot}} \tag{14}$$

Fig. 5 presents the distribution of energy efficiency, its sample mean, and the scalability of training time. The mean comparison in Fig. 5a indicates that ARQ–UCB attains a value comparable to the baselines, with a small shortfall relative to Q-Learning and Double Q-Learning that remains within about one percent. The cumulative distribution in Fig. 5b shows why this occurs. ARQ–UCB places less probability mass in the low-efficiency tail near 8.0×10^7 bits/J and transitions more steeply through the central region between

approximately 8.6×10^7 and 8.9×10^7 bits/J. This pattern reflects a tighter concentration around high-efficiency operating points and fewer low-efficiency episodes. In other words, ARQ–UCB improves the reliability of the energy–rate trade-off without necessarily increasing the global mean.



(a) Mean energy efficiency.

(b) Energy efficiency CDF.

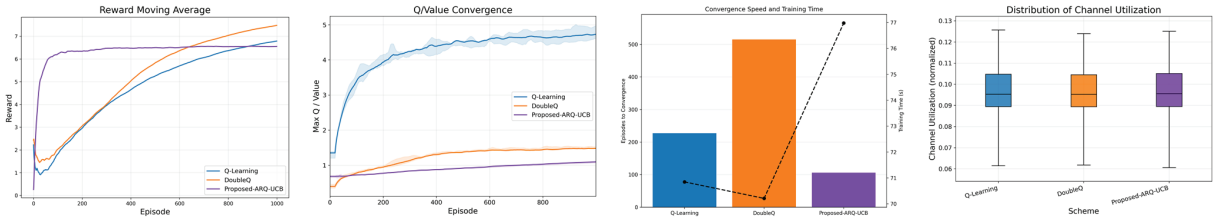
(c) Training time scalability across devices and class mix.

Figure 5: Energy efficiency and training time. ARQ–UCB concentrates probability mass in higher efficiency regions and keeps training time within a narrow band as the network scales.

The training-time comparison in Fig. 5c shows that the proposed ARQ–UCB exhibits traffic-dependent variation in convergence behavior. In particular, the balanced 30–40–30 class mix displays noticeably larger fluctuations. This occurs because symmetric traffic levels generate comparable queue pressures across classes, prolonging the period in which the UCB exploration term remains active and delaying early stabilization. In dense high-load scenarios such as the 40–30–30 profile, the exploration overhead can lead to episodes where ARQ–UCB requires substantially longer training time, consistent with the peak values observed in the figure. This behavior does not indicate degradation in steady-state performance; rather, it reflects the additional iterations required for confidence-driven exploration when multiple class backlogs evolve at similar rates. Across all mixes, the general scaling trend remains consistent with expectations for uncertainty-guided tabular learning in multi-class vehicular IoT environments.

6.5 Convergence Speed and Reliability

The convergence behavior in Fig. 6 highlights the learning stability of the proposed ARQ–UCB. The reward evolution in Fig. 6a shows a rapid rise during the first 50 episodes and stabilization near 6.3 by episode 150, while Q-Learning and Double Q-Learning require 600 and 400 episodes, respectively, to reach comparable levels. The Q-value trajectories in Fig. 6b display smooth convergence with reduced oscillations, indicating improved value consistency.



(a) Reward convergence. **(b)** Maximum-Q stabilization. **(c)** Episodes to convergence and total training time. **(d)** Channel utilization distribution.

Figure 6: Convergence speed and practicality. ARQ–UCB learns faster, stabilizes value estimates smoothly, and maintains efficient channel use.

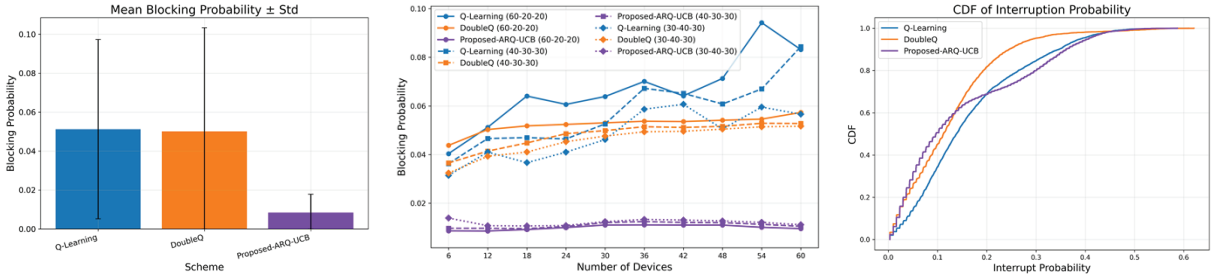
Fig. 6c summarizes the episodes and wall-clock time required for convergence. ARQ–UCB reaches the threshold with fewer episodes, and its training time remains competitive despite the added UCB exploration. In balanced traffic mixes, convergence may take longer because similar class loads keep the exploration term active for an extended period. Fig. 6d shows the distribution of normalized channel utilization, where all schemes maintain a stable and narrow range. ARQ–UCB displays slightly lower variability, indicating smoother scheduling behavior.

6.6 Blocking and Interruption Probability

The blocking probability denotes the ratio of denied attempts, which is quantifies in this work using Eq. (15). In the simulator, each queue has a finite capacity and a maximum tolerated waiting time; packets that arrive when the buffer is full or whose delay budget is exceeded are dropped and counted as blocked, an effect that naturally appears more frequently in lower-priority classes under heavy load.

$$P_b = N_{\text{deny}}/N_{\text{attempt}} \tag{15}$$

The Blocking and Interruption probability results are presented in Fig. 7. Fig. 7a illustrates that ARQ–UCB cuts blocking down to 0.0081, compared with 0.050–0.051 for the baseline schemes. The improvement, which is approximately 6.3 times, confirms the improved channel access management of the proposed solution. The standard deviation also shrinks from ±0.007 to ±0.002, showing the stability of the system. The interruption probability shown in Fig. 7c plots the CDF of interruption probability, which remains close to 0.10, satisfying the 3GPP reliability targets. These findings confirm that the confidence-driven mechanism empowers ARQ–UCB to maintain reliable communication against dense vehicular loads.



(a) Blocking probability (mean and standard deviation).

(b) Blocking scalability across devices and class mix.

(c) Interruption probability distribution.

Figure 7: Blocking and interruption behavior with scaling. ARQ–UCB reduces access denials, stabilizes interruption probability, and preserves robustness as the network grows.

6.7 Sensitivity Analysis of Key Hyperparameters

To extend the results of Figs. 2–7, we performed a sensitivity study to assess how robust ARQ–UCB is to its main hyperparameters and traffic density variations. We list the tested ranges and associated observed effects in Table 2.

The UCB exploration coefficient β ranged between 0.2 and 1.2. Bigger values revved exploration during earlier episodes, while smaller values delivered smoother updates; all configurations converged to similar long-term blocking and interruption probabilities. This validates that the confidence-based exploration mechanism retains stability across a wide range of settings. We varied the learning rate α between 0.03 and 0.10. As intuitively expected, increasing values of α improved the initial adaptation while smaller ones reduced oscillatory behavior. However, all metrics of final throughput, delay, and reliability remained unchanged regardless of the actual choice of α, which justifies that ARQ–UCB does not rely on any fine-tuning of step sizes. Finally, sensitivity to traffic load was examined by scaling the arrival rates proportionally.

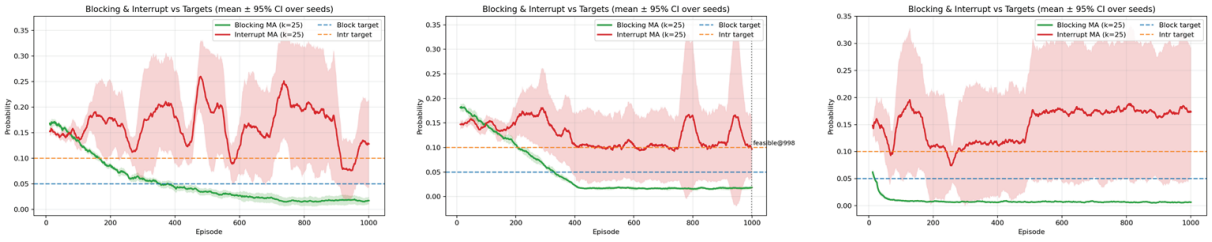
These results are compatible with the class-mix scenarios already shown in Figs. 2–7. As load increases, ARQ–UCB maintains low blocking probability and stable interruption levels, provided that the overall stability condition defined in Section 3 is satisfied. From Table 3, the trends indicate that ARQ–UCB retains strong learning dynamics under various hyperparameter settings and load conditions, further confirming its applicability in variable and unpredictable vehicular environments.

Table 3: Sensitivity of ARQ–UCB to key hyperparameters and traffic density.

Parameter	Range Tested	Observed Effect	Outcome
UCB coefficient β	0.2–1.2	Faster exploration for large β; smoother updates for small β	All settings converge to same reliability region
Learning rate α	0.03–0.10	Higher α speeds early learning; lower α reduces oscillations	Final blocking/interruption unchanged
Traffic density (arrival scaling)	0.8×–1.2× nominal	Increased load raises queue occupancy; convergence preserved	Reliability maintained under stability constraint

6.8 Reliability Tracking against Targets

Reliability in V-IoT scheduling is characterized by the system's ability to maintain the blocking probability P_b and interruption probability P_i below specified URLLC limits throughout training. The target thresholds are set to $P_b^{\text{target}} = 0.05$ and $P_i^{\text{target}} = 0.10$. To evaluate this behavior dynamically, Fig. 8 presents the moving average of these probabilities with a sliding window of $k = 25$ episodes, accompanied by 95% confidence intervals over multiple random seeds.



(a) Q-Learning. **(b)** Double Q-Learning. **(c)** ARQ–UCB.

Figure 8: Reliability tracking against targets with moving average window $k = 25$ and 95% confidence intervals. The horizontal reference lines denote $P_b^{\text{target}} = 0.05$ and $P_i^{\text{target}} = 0.10$.

In Fig. 8a, the Q-Learning baseline exhibits a steady decline in blocking probability from approximately 0.16 at the start of training to below the target limit of 0.05 around episode 380. After this point, the blocking rate remains compliant, though nominal fluctuations are visible toward later episodes. The probability of interruption, however, varies widely between 0.12 and 0.20 across training, while the confidence band also is quite wide. These fluctuations indicate that while Q-Learning eventually meets the blocking target, it fails to show stability in holding consistent interruption reliability, thus restricting its full suitability for stringent URLLC requirements.

The blocking probability of the Double Q-Learning trajectory in Fig. 8b is more stable and converges to around 0.045 after episode 400, while the interruption probability remains volatile with periodic surges above 0.25. It indicates that even though Double Q-Learning reduces value overestimation, it cannot maintain target reliability levels consistently.

Fig. 8c shows the behavior of the proposed ARQ–UCB algorithm. The blocking probability drops rapidly, crossing the target limit within the first 150 episodes and stabilizing close to 0.02 with a narrow confidence interval. The interruption probability converges close to the threshold 0.10, still showing small oscillations without significant overshoot. This behavior confirms that ARQ–UCB achieves early and stable compliance with the defined reliability constraints. Tighter confidence regions across both metrics indicate the robustness of UCB-guided exploration, effectively balancing aggressive learning and cautious adaptation during convergence.

6.9 Computational Complexity

All algorithms exhibit constant-time per-decision complexity, $\mathcal{O}(1)$. The average inference latency is 9.4 µs for Q-Learning, 9.7 µs for Double Q-Learning, and 19.0 µs for ARQ–UCB. Their 99th-percentile latencies are 26.2, 25.1, and 46.2 µs, remaining below 5% of the 1 ms scheduling slot. Their memory requirements are only 20 B, 40 B, and 60 B with approximately nine floating-point operations per update. The energy consumptions per decision are 311, 307, and 604 nJ for the three methods. These results verify that ARQ–UCB introduces negligible computational overhead while guaranteeing significant gains in reliability, learning stability, and adaptability for real-time vehicular edge applications.

7 Conclusion and Future Work

This paper proposes a lightweight RL framework for reliable and efficient spectrum access in V-IoT networks, named ARQ–UCB. The proposed method enhances classical tabular Q-Learning with a UCB exploration mechanism that adapts the intensity of exploration according to action-specific uncertainty. ARQ–UCB embeds confidence-aware terms into the Q-value update rule for preventing excessive exploration and promoting faster convergence in highly dynamic V-IoT environments characterized by bursty traffic and rapidly varying channel conditions. Extensive simulations carried out for realistic vehicular traffic profiles showed that ARQ–UCB outperforms baseline schemes consistently.

The proposed framework demonstrates up to 15% lower mean latency and about 6× lower blocking probability while sustaining solid gains in throughput and fairness. In typical scenarios, the blocking probability stays below 1%, which speaks to the stability and reliability of ARQ–UCB. Complexity analysis further confirmed that ARQ–UCB maintains constant-time decision complexity, $\mathcal{O}(1)$, with an average inference latency of 19 μs and a 99th-percentile latency below 50 µs, less than 5% of one 1 ms transmission slot, confirming the practical feasibility of the framework for real-time implementation in RSUs and vehicular edge controllers. Beyond the performance benefits, ARQ–UCB provides an interpretable and scalable foundation for adaptive spectrum management in future ITS. Having a small and transparent tabular structure with minimal computational cost, it is an appealing alternative for deep RL approaches that rely on large volumes of training data and a significant amount of processing power.

The proposed framework also has limitations. First, the evaluation is simulation-based and relies on synthetic traffic and channel realizations, which may not capture all mobility-induced effects observed in standardized V2X traces. Second, the tabular state representation is intentionally lightweight, but it can limit scalability when the state space grows due to richer context or larger deployments. These limitations can be addressed by validating ARQ–UCB on standardized mobility traces and by extending the current design with compact function approximation or representation learning while retaining strict decision-latency constraints.

Future research will extend ARQ–UCB along several directions. First, a multi-agent ARQ–UCB variant will be developed to support cooperative learning among distributed RSUs in large-scale V2X deployments. Second, the integration of reconfigurable intelligent surfaces and edge-federated learning will enable distributed spectrum coordination while preserving data privacy. Third, grounding the evaluation in standardized V2X mobility traces and real-world datasets will allow broader assessment of realism and complement the current synthetic analysis. Fourth, we will incorporate compact deep reinforcement learning baselines to provide broader performance context and to assess trade-offs beyond the lightweight tabular methods. Future research will also include a broader sensitivity analysis by varying traffic-model parameters such as arrival burstiness, class-mix ratios, and load asymmetry to more comprehensively evaluate ARQ–UCB's adaptability. Finally, coupling ARQ–UCB with emerging network slicing and semantic communication paradigms in 6G will further improve spectral efficiency, reliability, and energy sustainability within ultra-dense V-IoT environments.

Acknowledgement: The authors thank the Yeungnam University WINLab for computing support.

Funding Statement: This work was supported by the Deanship of Scientific Research, Vice Presidency for the Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia [Grant No. KFU260072].

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Adeel Iqbal, Tahir Khurshaid; methodology, Tahir Khurshaid, Mohammad Arif; software, Syed Abdul Mannan; validation, Tahir Khurshaid, Muhammad Faisal Siddiqui; formal analysis, Adeel Iqbal, Syed Abdul Mannan Kirmani; investigation, Adeel

Iqbal, Syed Abdul Mannan Kirmani; resources, Mohammad Arif; data curation, Muhammad Faisal Siddiqui; writing—original draft preparation, Adeel Iqbal, Tahir Khurshaid, Muhammad Faisal Siddiqui; writing—review and editing, Syed Abdul Mannan Kirmani, Mohammad Arif; visualization, Muhammad Faisal Siddiqui; supervision, Adeel Iqbal; project administration, Mohammad Arif; funding acquisition, Muhammad Faisal Siddiqui. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding authors upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: Adeel Iqbal is a Guest Editor of a special issue in CMC.

Appendix A Baseline Reinforcement Learning Schemes

This appendix summarizes the two tabular baselines used for comparison with ARQ–UCB: *Q-Learning* and *Double Q-Learning*. Both are model-free value-based methods that update action values from bootstrapped targets. Implementation parameters follow Section 5 and are not repeated here.

Appendix A.1 Standard Q-Learning (Tabular)

Algorithm A1: Baseline Q-learning

Require: Learning rate $\alpha \in (0,1]$, discount $\gamma \in (0,1]$, exploration rate $\epsilon \in [0,1]$, episodes E, horizon T

Initialize $Q(s,a) \leftarrow 0$ for all *(s, a)*
for episode $e = 1$ to E **do**
  Observe initial state s_0
  for time step $t = 0$ to $T-1$ **do**
    Action selection (ϵ-greedy **):**

$$a_t = \begin{cases} \text{uniform random action}, & \text{with probability } \epsilon, \\ \arg\max_a Q(s_t,a), & \text{otherwise.} \end{cases}$$

    Execute a_t, observe reward r_t and next state s_{t+1}
    TD target and update:

$$y_t \leftarrow r_t + \gamma \max_{a'} Q(s_{t+1},a'), \quad Q(s_t,a_t) \leftarrow (1-\alpha)\,Q(s_t,a_t) + \alpha\, y_t$$

    $s_t \leftarrow s_{t+1}$
  end for
end for

Algorithm A2: Baseline Double Q-learning

Require: Learning rate $\alpha \in (0,1]$, discount $\gamma \in (0,1]$, exploration rate $\epsilon \in [0,1]$, episodes E, horizon T

Initialize $Q^A(s,a) \leftarrow 0$ and $Q^B(s,a) \leftarrow 0$ for all *(s, a)*
for episode $e = 1$ to E **do**
  Observe initial state s_0
  for time step $t = 0$ to $T-1$ **do**
    Action selection (ϵ-greedy on $\frac{Q^A+Q^B}{2}$ **):**

$$a_t = \begin{cases} \text{uniform random action}, & \text{with probability } \epsilon, \\ \arg\max_a \left(\frac{Q^A(s_t,a)+Q^B(s_t,a)}{2} \right), & \text{otherwise.} \end{cases}$$

    Execute a_t, observe reward r_t and next state s_{t+1}
    Randomized update: draw $u \sim \text{Uniform}(0,1)$
    if $u < 0.5$ **then** ▷ update Q^A using Q^B for action selection

$$a^\star \leftarrow \arg\max_a Q^A(s_{t+1},a), \quad y_t \leftarrow r_t + \gamma\, Q^B(s_{t+1},a^\star)$$
$$Q^A(s_t,a_t) \leftarrow (1-\alpha)\, Q^A(s_t,a_t) + \alpha\, y_t$$

    else ▷ update Q^B using Q^A for action selection

$$a^\star \leftarrow \arg\max_a Q^B(s_{t+1},a), \quad y_t \leftarrow r_t + \gamma\, Q^A(s_{t+1},a^\star)$$
$$Q^B(s_t,a_t) \leftarrow (1-\alpha)\, Q^B(s_t,a_t) + \alpha\, y_t$$

    end if
    $s_t \leftarrow s_{t+1}$
  end for
end for

Appendix A.2 Double Q-Learning (Tabular)

Note. Action selection, targets, and updates are shown inside the algorithms to avoid cross-references. The same simulation setup and metrics from Section 5 were used for a fair comparison with ARQ–UCB in terms of throughput, delay, fairness, blocking, and energy efficiency.

Appendix B Compact Theoretical Justification of the UCB-Based Exploration Term

This appendix consolidates the theoretical justification of the optimism-based exploration term in Eq. (8), using the policy and update rules in Eqs. (7)–(10). It formalizes the assumptions and convergence behavior of ARQ–UCB in the stationary tabular setting.

Appendix B.1 Assumptions

Assumption A1 (Stationary Tabular MDP). The environment is a finite MDP with state–action space $(\mathcal{S},\mathcal{A})$, bounded rewards $r_t \in [0,1]$, and stationary transition kernel $P(s' \mid s,a)$. The learning-rate sequence $\{\alpha_t\}$ satisfies the Robbins–Monro conditions $\sum_t \alpha_t = \infty$ and $\sum_t \alpha_t^2 < \infty$.

Assumption A2 (Sufficient Exploration). Under the UCB action-selection rule of Eq. (7), every state–action pair (s, a) is visited infinitely often with probability one. The exploration bonus is

$$U_t(s,a) = c\sqrt{\frac{\ln t}{N_t(s,a)}}, \tag{A1}$$

where $N_t(s, a)$ is the visitation count.

Appendix B.2 Optimism and Regret Behavior

For each (s, a), the bonus satisfies

$$U_t(s,a) = O\left(\sqrt{\frac{\ln t}{N_t(s,a)}}\right), \tag{A2}$$

which ensures $Q_t(s, a) + U_t(s, a)$ remains optimistic about $Q^*(s, a)$ when $N_t(s, a)$ grows.

For a fixed state s, the induced bandit instance with arm means $Q^*(s, a)$ satisfies the UCB1 guarantees [16]:

$$\mathbb{E}[N_T(s,a)] = O(\ln T), \tag{A3}$$

$$R_T = O\left(\sqrt{T \ln T}\right), \tag{A4}$$

consistent with Eq. (10). Thus the fraction of suboptimal choices decays as $O(\sqrt{\ln T/T})$.

Appendix B.3 Convergence of the Q-Update

Under Assumptions A1 and A2, the ARQ–UCB update of Eq. (9) behaves as classical Q-learning with diminishing perturbations, since

$$U_t(s,a) \to 0 \qquad \text{as} \qquad N_t(s,a) \to \infty. \tag{A5}$$

Proposition A1 (Asymptotic Optimality Under Stationary Dynamics). Under Assumptions A1 and A2, the ARQ–UCB update satisfies

$$Q_t(s,a) \xrightarrow[t\to\infty]{\text{a.s.}} Q^*(s,a), \qquad \forall (s,a) \in \mathcal{S} \times \mathcal{A}, \tag{A6}$$

and the cumulative regret is sublinear:

$$R_T = O\left(\sqrt{T \ln T}\right), \qquad \lim_{T\to\infty} \frac{R_T}{T} = 0. \tag{A7}$$

Sketch of justification. As $U_t(s, a)$ decays, the update becomes standard Q-learning with vanishing perturbations. By Watkins–Dayan convergence, $Q_t \to Q^*$ almost surely. Regret scaling follows from UCB analysis. Furthermore,

$$\frac{1}{T}\sum_{t=1}^{T} U_t(s_t, a_t) = O\left(\sqrt{\frac{\ln T}{T}}\right) \longrightarrow 0. \tag{A8}$$

Appendix B.4 Reliability Cost Behavior

Let $\mathcal{E}_t$ denote blocking or interruption, and let $J(\pi)$ denote the resulting reliability cost (Section 6). Since learned policies π_t become greedy with respect to Q^* and regret is sublinear,

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{P}(\mathcal{E}_t \mid \pi_t) \longrightarrow J(\pi^*), \tag{A9}$$

which is consistent with the empirical stabilization in Figs. 6–8.

Appendix B.5 Non-Stationary Environments

When traffic or channel statistics evolve over time, Assumption B1 no longer holds. In such settings ARQ–UCB should be viewed as a tracking heuristic whose empirical robustness is demonstrated in Section 6. Developing formal regret and optimality guarantees for non-stationary V2X environments remains an important direction for future work.

References

1. Khan AR, Jamlos MF, Osman N, Ishak MI, Dzaharudin F, Yeow YK, et al. DSRC technology in Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) IoT system for Intelligent Transportation System (ITS): a review. In: Recent trends in mechatronics towards industry 40. Singapore: Springer; 2021. p. 97–106. doi:10.1007/978-981-33-4597-3_10.
2. Iqbal A, Nauman A, Khurshaid T. Lightweight reinforcement learning for priority-aware spectrum management in vehicular IoT networks. Sensors. 2025;25(21):6777. doi:10.3390/s25216777.
3. Pawar V, Zade N, Vora D, Khairnar V, Oliveira A, Kotecha K, et al. Intelligent transportation system with 5G vehicle-to-everything (V2X): architectures, vehicular use cases, emergency vehicles, current challenges and future directions. IEEE Access. 2024;12:183937–60. doi:10.1109/access.2024.3506815.
4. Durisi G, Koch T, Popovski P. Toward massive, ultrareliable, and low-latency wireless communication with short packets. Proc IEEE. 2016;104(9):1711–26. doi:10.1109/jproc.2016.2537298.
5. Popovski P, Nielsen JJ, Stefanovic C, De Carvalho E, Strom E, Trillingsgaard KF, et al. Wireless access for ultra-reliable low-latency communication: principles and building blocks. IEEE Netw. 2018;32(2):16–23. doi:10.1109/mnet.2018.1700258.
6. Bennis M, Debbah M, Poor HV. Ultrareliable and low-latency wireless communication: tail, risk, and scale. Proc IEEE. 2018;106(10):1834–53. doi:10.1109/jproc.2018.2867029.
7. Abboud K, Omar HA, Zhuang W. Interworking of DSRC and cellular network technologies for V2X communications: a survey. IEEE Trans Veh Technol. 2016;65(12):9457–70. doi:10.1109/tvt.2016.2591558.
8. Sutton RS, Barto AG. Reinforcement learning: an introduction. 1st ed. Cambridge, MA, USA: MIT Press; 1998.
9. Sutton RS, Barto AG. Reinforcement learning: an introduction. 2nd ed. Cambridge, MA, USA: MIT Press; 2018.
10. Luong NC, Hoang DT, Gong S, Niyato D, Wang P, Liang YC, et al. Applications of deep reinforcement learning in communications and networking: a survey. IEEE Commun Surv Tutor. 2019;21(4):3133–74.
11. Yang H, Xie X, Kadoch M. Intelligent resource management based on reinforcement learning for ultra-reliable and low-latency IoV communication networks. IEEE Trans Veh Technol. 2019;68(5):4157–69. doi:10.1109/tvt.2018.2890686.
12. Watkins CJ, Dayan P. Q-learning. Mach Learn. 1992;8(3):279–92.
13. Hasselt H. Double Q-learning. In: NIPS'10: Proceedings of the 24th International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc.; 2010. p. 2613–21.
14. Iqbal A, Khurshaid T, Nauman A, Kim SW. Adaptive communication model for QoS in vehicular IoT systems using CTMC. Sensors. 2025;25(6):1818. doi:10.3390/s25061818.
15. Iqbal A, Khurshaid T, Qadri YA, Nauman A, Kim SW. Priority-aware spectrum management for QoS optimization in vehicular IoT. Sensors. 2025;25(11):3342. doi:10.3390/s25113342.

16. Auer P, Cesa-Bianchi N, Fischer P. Finite-time analysis of the multiarmed bandit problem. Mach Learn. 2002;47(2):235–56. doi:10.1023/a:1013689704352.
17. Liang L, Ye H, Li GY. Spectrum sharing in vehicular networks based on multi-agent reinforcement learning. IEEE J Sel Areas Commun. 2019;37(10):2282–92. doi:10.1109/jsac.2019.2933962.
18. Ye H, Li GY. Deep reinforcement learning for resource allocation in V2V communications. In: 2018 IEEE International Conference on Communications (ICC). Piscataway, NJ, USA: IEEE; 2018. p. 1–6.
19. Nishio T, Yonetani R. Client selection for federated learning with heterogeneous resources in mobile edge. In: ICC 2019—2019 IEEE International Conference on Communications (ICC). Piscataway, NJ, USA: IEEE; 2019. p. 1–7.
20. Li X, Lu L, Ni W, Jamalipour A, Zhang D, Du H. Federated multi-agent deep reinforcement learning for resource allocation of vehicle-to-vehicle communications. IEEE Trans Veh Technol. 2022;71(8):8810–24. doi:10.1109/tvt.2022.3173057.
21. Zhou Q, Guo C, Wang C, Cui L. Radio resource management for C-V2X using graph matching and actor-critic learning. IEEE Wirel Commun Lett. 2022;11(12):2645–9. doi:10.1109/lwc.2022.3213176.
22. Zhang X, Peng M, Yan S, Sun Y. Deep-reinforcement-learning-based mode selection and resource allocation for cellular V2X communications. IEEE Internet Things J. 2019;7(7):6380–91. doi:10.1109/jiot.2019.2962715.
23. Liu Y, Yu H, Xie S, Zhang Y. Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks. IEEE Trans Veh Technol. 2019;68(11):11158–68. doi:10.1109/tvt.2025.3612473.
24. Wu H, Jin J, Ma H, Xing L. Federation-based deep reinforcement learning cooperative cache in vehicular edge networks. IEEE Internet Things J. 2023;11(2):2550–60. doi:10.1109/jiot.2023.3292374.
25. Ullah I, Singh SK, Adhikari D, Khan H, Jiang W, Bai X. Multi-agent reinforcement learning for task allocation in the internet of vehicles: exploring benefits and paving the future. Swarm Evol Comput. 2025;94:101878. doi:10.1016/j.swevo.2025.101878.
26. Strehl AL, Li L, Wiewiora E, Langford J, Littman ML. PAC model-free reinforcement learning. In: Proceedings of the 23rd International Conference on Machine Learning. New York, NY, USA: ACM; 2006. p. 881–8.