

Article

MoReSP: A Multiobjective Mobility- and Reliability-Aware Scheduling Model for RSU-Assisted Vehicular IoT Networks

Muhammad Faisal Siddiqui ^{1,*}  and Adeel Iqbal ^{2,*} ¹ Department of Computer Engineering, College of Computer Science and Information Technology, King Faisal University, Al Ahsa 31982, Saudi Arabia² School of Computer Science & Engineering, Yeungnam University, Gyeongsan-si 38541, Republic of Korea

* Correspondence: msiddiqui@kfu.edu.sa (M.F.S.); adeeliqbal@yu.ac.kr (A.I.)

Abstract

Vehicular Internet of Things (V-IoT) networks require reliable scheduling for safety-critical communication, cooperative awareness, and cooperative perception under dynamic mobility and limited roadside infrastructure. This paper proposes MoReSP, a Mobility- and Reliability-aware Scheduling Policy for roadside unit (RSU)-assisted V-IoT networks. MoReSP uses mobility-regime inference, structured action scoring, safety projection, and episodic parameter adaptation to select among deny, grant, preempt, coexist, and handoff actions. Its multiobjective formulation jointly minimizes average delay, communication energy consumption, admission-adjusted reliability loss, a penalty for cooperative perception message (CPM) delivery/freshness, and RSU-load imbalance. The framework is evaluated under vehicle-load variation, Nagel–Schreckenberg (NaSch) density variation, and RSU-capacity scaling using admission-adjusted metrics that penalize excessive blocking and interruption. MoReSP is compared with five literature-grounded benchmark families: Age of Correlated Information (AoCI)-Heuristic, RSU-Coop, Handoff-Aware, vehicle-to-everything (V2X)-Priority, and Adaptive Learning-based Task Offloading multi-armed bandit (ALTO-MAB). Simulation results show that MoReSP achieves the lowest admission-adjusted system cost across all evaluated scenarios. At nominal RSU capacity, MoReSP reduces the system cost by 43.6% compared with the best baseline. Under high vehicle load, it reduces the cost by 54.4% at arrival scale 2.0 and maintains effective packet and CPM delivery ratios of 0.849 and 0.828, respectively. These results demonstrate that MoReSP provides a reliable and balanced scheduling solution for dynamic V-IoT environments.

Keywords: Vehicular Internet of Things; RSU-assisted scheduling; mobility-aware resource allocation; cooperative perception messages; vehicular edge computing; V2X communication; reliability-aware scheduling

**MSC:** 90B18; 90B20; 90C29; 90C40; 68T05

Academic Editor: Nadir Farhi

Received: 29 May 2026

Revised: 27 June 2026

Accepted: 29 June 2026

Published: 3 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Vehicular Internet of Things (V-IoT) networks are becoming a core communication layer for intelligent transportation systems, cooperative perception, traffic-aware mobility services, and safety-critical vehicle-to-everything (V2X) applications. In these networks, vehicles, roadside units (RSUs), sensors, and edge nodes exchange status updates, perception data, and control information under rapidly changing topologies and wireless channel conditions. Compared with static Internet of Things (IoT) deployments, V-IoT systems

experience stronger mobility effects, burstier packet arrivals, intermittent connectivity, and tighter latency-reliability requirements. These characteristics make RSU-side packet admission and scheduling particularly difficult when multiple vehicles compete for limited communication and edge resources.

Reliable V-IoT scheduling must support different traffic classes with different service requirements. Safety or event-driven messages require low latency and high reliability, cooperative awareness messages (CAMs) provide periodic vehicle-state information, and cooperative perception messages (CPMs) carry sensor- or perception-derived information for shared environmental awareness. In dense or unstable traffic conditions, queue buildup, handoff pressure, wireless-link degradation, and RSU load imbalance can reduce packet delivery and degrade the freshness of CPM information. A throughput-driven scheduler may admit too many packets and increase failures, while an overly conservative scheduler may reduce failed transmissions by blocking useful information. Therefore, practical V-IoT scheduling requires a multiobjective admission policy that minimizes delay, communication energy consumption, admission-adjusted reliability loss, CPM delivery/freshness penalty, and RSU-load imbalance, while conditioning decisions on mobility, channel, and queue dynamics.

Several research directions have addressed parts of this problem. Age-aware and freshness-aware scheduling methods aim to reduce stale vehicular information [1]. Vehicular edge computing and RSU-assisted resource allocation improve infrastructure-side coordination and offloading efficiency [2–4]. Handoff-aware mechanisms reduce mobility-induced service disruption [5]. Priority-aware and ultra-reliable low-latency communication (URLLC)-oriented V2X methods protect reliability-sensitive vehicular traffic [6–8]. Cooperative perception studies further show that communication decisions should consider the usefulness and freshness of exchanged perception information [9–12]. Adaptive online learning methods, including Adaptive Learning-based Task Offloading (ALTO) and multi-armed bandit (MAB)-style policies, provide lightweight decision-making under uncertain edge conditions [13]. These studies provide important foundations, but they usually emphasize one dominant objective, such as freshness, cooperation, handoff continuity, priority, or online adaptation.

The remaining challenge is to design an RSU-resident scheduler that treats these factors as coupled rather than independent. Mobility affects link reliability and handoff risk; admission decisions affect delay, blocking, and interruption; CPM usefulness depends on both perception value and successful, timely delivery; and RSU cooperation affects load balance and migration overhead. Optimizing one dimension in isolation can therefore lead to undesirable behavior under changing traffic density or vehicle load. This motivates a scheduling policy that combines mobility awareness, reliability-sensitive admission, CPM value awareness, and safety-feasible action selection within one decision process.

This paper proposes MoReSP, a Mobility- and Reliability-aware Scheduling Policy for RSU-assisted V-IoT networks. MoReSP observes queue, channel, traffic class, mobility, CPM value, and RSU load indicators and selects among deny, grant, preempt, coexist, and handoff actions. The policy combines mobility-regime inference, structured action scoring, safety projection, and episodic parameter adaptation. To avoid misleading conclusions from raw delivery metrics, the evaluation uses admission-adjusted indicators that penalize excessive blocking and interruption.

The main contributions of this paper are summarized as follows:

- A mobility- and reliability-aware RSU-side scheduling framework is developed for heterogeneous V-IoT traffic, including safety messages, CAMs, and CPMs, under dynamic mobility and multi-RSU edge conditions.
- A structured MoReSP decision policy is proposed by combining mobility-regime inference, interpretable action scoring, safety projection, and episodic parameter adaptation for five scheduling actions: deny, grant, preempt, coexist, and handoff.

- An admission-adjusted multiobjective evaluation is introduced to quantify effective packet delivery, effective CPM delivery, delay, transmission energy, reliability loss, CPM freshness degradation, and RSU-load imbalance while explicitly penalizing excessive blocking and interruption.
- A controlled simulation study is conducted under vehicle-load variation, Nagel–Schreckenberg (NaSch) density variation, and RSU-capacity scaling against five literature-grounded baselines: Age of Correlated Information (AoCI)-Heuristic, RSU-Coop, Handoff-Aware, V2X-Priority, and ALTO-MAB.
- Quantitative results show that MoReSP achieves the lowest admission-adjusted system cost across all evaluated scenarios. At nominal RSU capacity, MoReSP reduces system cost by 43.6% compared with the best baseline, while under high vehicle load, it reduces cost by 54.4% at arrival scale 2.0 and maintains effective packet and CPM delivery ratios of 0.849 and 0.828, respectively.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the system model. Section 4 formulates the scheduling problem. Section 5 describes MoReSP. Section 6 presents the simulation setup and evaluation methodology. Section 7 discusses the results. Section 8 concludes the paper.

2. Related Work and Motivation

Resource scheduling in V-IoT networks has been studied from several directions, including information freshness, vehicular edge computing, RSU cooperation, mobility-aware association, reliability-sensitive V2X resource allocation, cooperative perception, and adaptive online learning. These studies provide useful building blocks, but most of them optimize a single dominant objective or operate at a different abstraction level from RSU-resident packet scheduling. The following discussion critically reviews these directions and identifies the gap addressed by MoReSP.

Age of Information (AoI)-aware and freshness-aware scheduling are closely related to V-IoT, as safety and perception information lose value with delay. Chen et al. studied age-aware radio resource management for vehicular networks using proactive deep reinforcement learning, showing that scheduling decisions can reduce stale vehicular updates [1]. Joint assignment and scheduling for Age of Correlated Information (AoCI) further showed that correlation among information sources affects scheduling performance [14]. Application-oriented AoCI optimization and the Age of Incorrect Information (AoII) metric also indicate that information usefulness depends not only on update age but also on semantic relevance and correctness [15,16]. However, freshness-oriented optimization mainly captures timeliness and does not directly address the combined effects of packet blocking, interruption, RSU imbalance, and mobility-induced link degradation. This limitation is important for CPM traffic, where a scheduler must preserve freshness without sacrificing reliability-sensitive safety traffic or overloading RSU resources.

Vehicular edge computing studies have improved the use of infrastructure-side computation and communication resources. Survey work on vehicular edge computing highlights the role of edge nodes in reducing latency and supporting mobility-aware services [2]. Collaborative task offloading has been used to improve resource sharing in vehicular multi-access edge networks [17]. Digital-twin-enabled vehicular edge computing has also been studied for task offloading and resource allocation [3], while adaptive digital twins and mobility-aware multiobjective offloading have been used to improve edge intelligence under dynamic vehicular conditions [18,19]. Meta-reinforcement learning has further been explored for fast task-offloading adaptation in edge computing [20]. These approaches are valuable for computation-centric V2X services, but they usually abstract packet-level

scheduling decisions and do not explicitly model mixed safety, CAM, and CPM traffic competing for RSU-side wireless access.

RSU cooperation and edge association have also been studied to improve connectivity and load distribution. Lin et al. proposed privacy-preserving joint edge association and power optimization for Internet-of-Vehicles systems through federated multi-agent reinforcement learning [4]. Such approaches demonstrate that distributed infrastructure coordination can improve resource use in the presence of mobility. However, edge association and power optimization do not by themselves guarantee reliable packet admission, CPM freshness, or safety-feasible action selection. A cooperative RSU policy may balance load but still make unreliable scheduling decisions if channel quality, traffic class, and interruption risk are not jointly considered.

Mobility-aware handoff and edge selection address another important part of the V-IoT scheduling problem. Ren et al. studied handoff-aware distributed computing in high-altitude-platform-assisted vehicular networks and showed that service continuity is strongly affected by mobility and reassociation events [5]. Tahir et al. further considered communication-aware consistent edge selection for mobile users and autonomous vehicles, incorporating access-point load and connectivity stability into edge-selection decisions [21]. These studies confirm the importance of mobility continuity, but handoff-aware design alone is insufficient for multi-class V-IoT scheduling because it does not fully capture the interaction among traffic urgency, CPM value, packet admission, and reliability-aware action filtering.

Reliability-sensitive V2X resource allocation has been widely explored for safety-critical vehicular communication. Zhang et al. used deep reinforcement learning for cellular V2X mode selection and resource allocation, focusing on resource assignment under V2X communication requirements [6]. Wu et al. addressed ultra-reliable low-latency communication (URLLC)-aware resource allocation for heterogeneous vehicular edge computing, emphasizing reliability and latency constraints [7]. Event-triggered reinforcement learning has also been used for joint resource allocation in URLLC V2X communication [8]. More recently, graph neural networks have been combined with deep reinforcement learning to capture topology and interference relations in V2X resource allocation [22]. These works strengthen reliability-aware V2X control, but they are generally not designed as interpretable RSU-side schedulers that jointly account for CPM freshness, mobility regime, admission decisions, and post-decision safety projection.

Cooperative perception research provides another relevant perspective because CPM traffic carries perception information rather than ordinary status updates. When2com learns when agents should communicate to improve multi-agent perception efficiency [9], while Where2comm reduces communication overhead by using spatial confidence maps for collaborative perception [10]. Probabilistic V2X data fusion and cooperative bird's-eye-view perception further show that communication-aware perception exchange can improve environmental understanding under limited bandwidth [23,24]. OPV2V provides an open benchmark for vehicle-to-vehicle cooperative perception, while V2X-Real and V2X-ReaLO extend this direction toward large-scale and realistic vehicle-to-everything perception settings [12,25,26]. Interruption-aware cooperative perception further shows that V2X communication failures can directly affect autonomous-driving perception quality [11]. These studies motivate CPM-aware scheduling, but they mainly focus on perception sharing, feature fusion, or dataset construction. They do not provide an RSU-side admission and scheduling policy that jointly considers safety/CAM/CPM coexistence, RSU load, mobility instability, and conservative reliability accounting.

Adaptive online learning methods provide lower-complexity decision alternatives for dynamic vehicular edge systems. Sun et al. proposed adaptive learning-based task

offloading for vehicular edge computing, using online decision-making to handle uncertain service conditions [13]. Such multi-armed-bandit-style methods are attractive because they require less modeling than full reinforcement learning. However, they generally do not represent the temporal coupling among queue evolution, mobility regime, CPM freshness, RSU load, and safety feasibility. This makes them useful as lightweight baselines but insufficient as complete scheduling solutions for dense and reliability-sensitive V-IoT scenarios.

Table 1 summarizes the positioning of representative studies relative to the proposed work. The comparison shows that prior studies usually emphasize one or two dimensions, such as freshness, offloading, handoff, reliability, cooperative perception, or online adaptation. In contrast, MoReSP is designed for RSU-side scheduling where traffic heterogeneity, mobility regime, reliability risk, CPM value, RSU load, and safety-feasible action correction must be considered together.

Table 1. Comparison of representative related studies and MoReSP positioning.

Study/Direction	Main Focus	Strength	Limitation Relative to MoReSP
AoI-/AoCI-aware vehicular scheduling [1]	Information freshness and update timeliness	Reduces stale vehicular information	Does not jointly model admission blocking, interruption, RSU imbalance, mobility regime, and CPM-aware safety filtering
Vehicular edge computing and digital-twin offloading [2,3,20]	Task offloading and edge-resource allocation	Improves computation and infrastructure utilization	Primarily computation-centric; packet-level safety/CAM/CPM scheduling is usually abstracted
Federated edge association and RSU coordination [4]	Edge association, power control, and distributed infrastructure coordination	Supports cooperative infrastructure use under mobility	Does not directly provide RSU-side admission control with CPM freshness and safety-feasible action projection
Handoff-aware and communication-aware edge selection [5,21]	Mobility continuity and reassociation stability	Reduces service disruption under mobility	Mobility continuity is addressed, but reliability, CPM value, class heterogeneity, and scheduling safety are not jointly optimized
Reliability-aware and URLLC-oriented V2X allocation [6–8,22]	V2X reliability, latency, resource allocation, and interference-aware learning	Strengthens reliability-sensitive V2X communication	Often focuses on physical-layer and medium-access-control resource allocation or learning-based control without interpretable RSU-side regime-aware scheduling
Cooperative perception and CPM communication [9–12,25]	Perception sharing, communication efficiency, and cooperative sensing	Improves perception utility and reduces unnecessary exchange	Does not address multi-class RSU admission, blocking/interruption penalties, and edge-load-aware scheduling
Adaptive online offloading and MAB-style learning [13]	Lightweight online decision-making under uncertainty	Low-complexity adaptation from reward feedback	Limited representation of queue dynamics, mobility regimes, safety constraints, and CPM freshness interactions
MoReSP	RSU-side mobility- and reliability-aware scheduling for safety/CAM/CPM traffic	Jointly considers mobility regime, reliability risk, CPM value, RSU load, admission control, and safety projection	Designed as a structured scheduling framework; future work can extend it toward distributed multi-RSU learning and trace-driven deployment validation

The above comparison identifies the central research gap: existing studies offer strong solutions for specific subproblems but do not provide a unified RSU-resident scheduling framework for heterogeneous V-IoT traffic under coupled mobility, reliability, CPM freshness, and RSU-load constraints. MoReSP addresses this gap by combining mobility-regime inference, structured action scoring, safety projection, and episodic adaptation into a single scheduling policy. This design allows the scheduler to select among deny, grant, preempt,

coexist, and handoff actions while explicitly penalizing excessive blocking and interruption through admission-adjusted evaluation.

3. System Model

This section defines the RSU-assisted V-IoT system considered in this work. The network consists of a set of vehicles, RSUs, wireless V2X links, and an RSU-edge layer that supports packet scheduling, handoff, service migration, and cooperative perception delivery. Vehicles generate heterogeneous safety, CAM, and CPM traffic over slotted time. Figure 1 illustrates the considered system model.

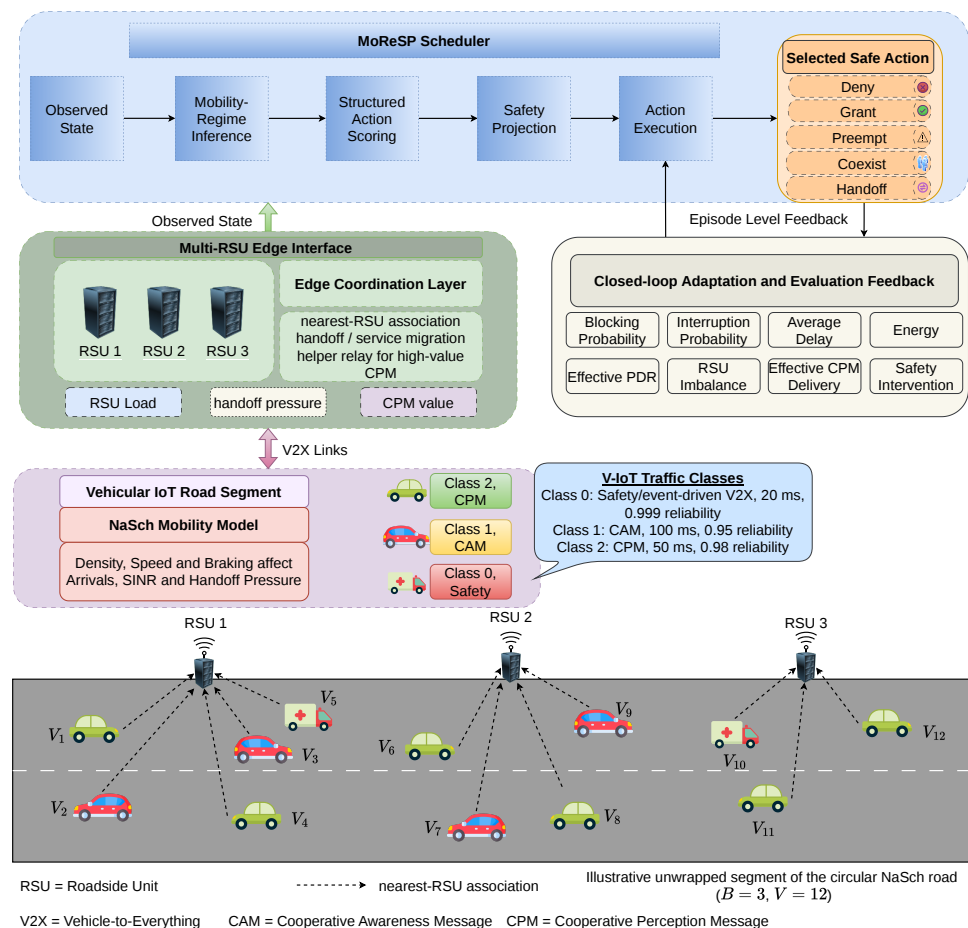


Figure 1. System model of the proposed RSU-assisted V-IoT network.

Let

$$\mathcal{V} = \{1, 2, \dots, V\} \quad (1)$$

denote the set of vehicles and

$$\mathcal{B} = \{1, 2, \dots, B\} \quad (2)$$

denote the set of RSUs. In the implemented setting, $B = 3$. The system operates over discrete time slots indexed by t , with slot duration $\Delta t = 1$ ms. Each vehicle is associated with one serving RSU at a given slot. If $x_i(t)$ denotes the position of vehicle i and x_b denotes the position of RSU b , the serving RSU is selected according to the nearest-RSU association:

$$b_i(t) = \arg \min_{b \in \mathcal{B}} d_{i,b}(t), \quad (3)$$

where $d_{i,b}(t)$ is the road distance between vehicle i and RSU b . A service migration or handoff event may occur when the serving association changes or when the current link becomes unreliable.

The traffic model contains three service classes:

$$\mathcal{C} = \{0, 1, 2\}, \quad (4)$$

where class 0 denotes safety or event-driven V2X messages; class 1 denotes CAM traffic; and class 2 denotes CPM traffic. Each class $c \in \mathcal{C}$ is characterized by its base sampling rate λ_c , deadline D_c , reliability target ξ_c , and packet size L_c . The adopted service profile is

$$\begin{aligned} (\lambda_0, D_0, \xi_0, L_0) &= (2 \text{ Hz}, 20 \text{ ms}, 0.999, 3200 \text{ bits}), \\ (\lambda_1, D_1, \xi_1, L_1) &= (10 \text{ Hz}, 100 \text{ ms}, 0.95, 2400 \text{ bits}), \\ (\lambda_2, D_2, \xi_2, L_2) &= (5 \text{ Hz}, 50 \text{ ms}, 0.98, 9000 \text{ bits}). \end{aligned} \quad (5)$$

This profile captures the strict reliability requirement of safety traffic, the periodic status-update nature of CAMs, and the larger payload size of CPMs. For vehicle i with traffic class c_i , packet arrivals follow a slot-level Bernoulli process. The arrival probability is

$$p_i(t) = 1 - \exp(-\lambda_{c_i} \lambda_{\text{scale}} \chi_i m_{c_i}(t) \Delta t), \quad (6)$$

where λ_{scale} is the offered-load scaling factor; χ_i is the device-specific load factor; and $m_{c_i}(t)$ is a class-dependent mobility multiplier. This multiplier increases the arrival rate of safety messages during unstable braking conditions and increases CPM demand under dense traffic. If $A_i(t) \in \{0, 1\}$ indicates a new packet arrival, the queue of vehicle i evolves as

$$Q_i(t+1) = [Q_i(t) - S_i(t)]^+ + A_i(t) L_{c_i}, \quad (7)$$

where $S_i(t)$ is the successfully served traffic in bits. The queue age increases while the queue remains nonempty, and a deadline violation is recorded when the queue age exceeds D_{c_i} . The exponential mapping in Equation (6) ensures that $p_i(t) \in [0, 1]$ for all evaluated values of λ_{scale} . As λ_{scale} increases, more frequent arrivals increase $Q_i(t)$ and queue age whenever the available service $S_i(t)$ is insufficient; the resulting active-queue pressure, oldest queue age, and average queue age are included in the scheduler observation and directly influence congestion- and urgency-aware action selection.

Vehicular mobility is modeled using the NaSch cellular traffic process on a circular road. The road contains $N_{\text{cell}} = 120$ cells, each with length $l_{\text{cell}} = 7.5$ m, giving

$$L_{\text{road}} = N_{\text{cell}} l_{\text{cell}} = 900 \text{ m}. \quad (8)$$

Let $v_j(t)$, $x_j(t)$, and $g_j(t)$ denote the velocity, position, and forward gap of traffic vehicle j . The mobility update is

$$\begin{aligned} v_j(t) &\leftarrow \min\{v_j(t) + 1, v_{\text{max}}\}, \\ v_j(t) &\leftarrow \min\{v_j(t), g_j(t)\}, \\ v_j(t) &\leftarrow \max\{v_j(t) - 1, 0\}, \quad \text{with probability } p_{\text{slow}}, \\ x_j(t+1) &= [x_j(t) + v_j(t)] \bmod N_{\text{cell}}, \end{aligned} \quad (9)$$

where $v_{\text{max}} = 5$ cells/slot. The extracted mobility indicators are traffic density $\rho(t)$, normalized speed $v(t)$, and braking rate $\beta(t)$.

The V2X wireless channel operates at carrier frequency $f_{\text{car}} = 5.9$ GHz. It includes distance-dependent path loss, log-normal shadowing, fast fading, interference,

and mobility-induced degradation. The noise floor is set to -94 dBm, while the shadowing and fast-fading standard deviations are 3 dB and 2 dB, respectively. Let P_i^{tx} denote the transmit power of vehicle i . The received signal level is

$$P_i^{\text{rx}}(t) = P_i^{\text{tx}} - \text{PL}(d_i(t)) + X_\sigma(t) + F_\sigma(t) - \Delta_\rho(t) - \Delta_\beta(t) - \Delta_v(t), \quad (10)$$

where $d_i(t)$ is the distance from vehicle i to its serving RSU, $X_\sigma(t)$ is the shadowing term, $F_\sigma(t)$ is the fast-fading term, and $\Delta_\rho(t)$, $\Delta_\beta(t)$, and $\Delta_v(t)$ denote density-, braking-, and speed-related channel degradation. The path loss is modeled as

$$\text{PL}(d_i(t)) = 32.4 + 20 \log_{10}(f_{\text{car}}) + 31.9 \log_{10}(d_i(t)), \quad (11)$$

where f_{car} is expressed in GHz and $d_i(t)$ is expressed in meters. In the implementation, the link distance is lower-bounded by $d_0 = 5$ m to avoid singular path-loss values at very short distances. The instantaneous signal-to-interference-plus-noise ratio (SINR) in dB is

$$\gamma_i(t) = P_i^{\text{rx}}(t) - P^{\text{noise+int}}(t), \quad (12)$$

where $P^{\text{noise+int}}(t)$ is a combined noise-plus-interference power in dBm. The slot service capacity is

$$R_i(t) = W\Delta t \min\left\{\log_2\left(1 + 10^{\gamma_i(t)/10}\right), R_{\max}\right\}, \quad (13)$$

where W is the channel bandwidth and $R_{\max}$ is the spectral-efficiency cap.

The RSU-edge layer maintains load states for all RSUs. Let $\ell_b(t) \in [0, 1]$ denote the normalized association load of RSU b , computed as the fraction of vehicles currently associated with that RSU. The average load and maximum load are

$$\bar{\ell}(t) = \frac{1}{B} \sum_{b \in \mathcal{B}} \ell_b(t), \quad \ell_{\max}(t) = \max_{b \in \mathcal{B}} \ell_b(t). \quad (14)$$

The RSU imbalance is

$$I_{\text{RSU}}(t) = \ell_{\max}(t) - \bar{\ell}(t). \quad (15)$$

At each scheduling slot, the RSU-edge layer updates the serving-RSU association of each vehicle according to the nearest-RSU association rule and derives the normalized RSU-load state from the current associations. The resulting load indicators are incorporated into the scheduler state for RSU-load-aware action selection. When a handoff action is executed, the model includes the associated service-migration latency and energy overhead. The RSU-load state synchronization and refresh are abstracted within the $\Delta t = 1$ ms RSU-edge control cycle before scheduling action selection.

For CPM traffic, a value-of-information proxy is used to represent perception usefulness. Let $V_i^{\text{CPM}}(t)$ denote the CPM value of vehicle i . It is defined as

$$V_i^{\text{CPM}}(t) = \text{clip}_{[0,1]}(\alpha_\rho \rho(t) + \alpha_\beta \beta(t) + \alpha_d \delta_i(t) + \alpha_v(1 - v(t))), \quad (16)$$

The normalized CPM value coefficients are set to $\alpha_\rho = 0.35$, $\alpha_\beta = 0.30$, $\alpha_d = 0.20$, and $\alpha_v = 0.15$. This allocation assigns greater importance to traffic density and braking activity, while retaining distance-based spatial novelty and reduced-speed effects in the CPM value estimate. The distance-related novelty factor is defined as

$$\delta_i(t) = \text{clip}_{[0,1]}\left(\frac{d_{i,b_i(t)}(t)}{L_{\text{road}}/2}\right), \quad \delta_i(t) \in [0, 1], \quad (17)$$

where $d_{i,b_i(t)}(t)$ denotes the shortest circular-road distance between vehicle i and its serving RSU $b_i(t)$. A larger value of $\delta_i(t)$ indicates greater vehicle–RSU separation and is used as a distance-based spatial-novelty proxy for CPM traffic. The coefficients α_ρ , α_β , α_d , and α_v are non-negative weights. For non-CPM packets, $V_i^{\text{CPM}}(t) = 0$.

At each slot, the scheduler selects one action from

$$\mathcal{A} = \{\text{deny}, \text{grant}, \text{preempt}, \text{coexist}, \text{handoff}\}. \quad (18)$$

The deny action rejects the current service opportunity; grant provides direct service; preempt prioritizes urgent traffic; coexist enables shared or cooperative service; and handoff triggers RSU reassociation or edge migration. Let $Y_i(t) \in \{0, 1\}$ indicate whether the scheduled transmission succeeds. The served traffic is

$$S_i(t) = Y_i(t) \min\{Q_i(t), R_i(t)\}. \quad (19)$$

The vehicle-side communication energy is calculated for successful service actions. For vehicle (i), the slot-level service energy is

$$E_i^{\text{svc}}(t) = \kappa(a(t)) \left[10^{\frac{P_i^{\text{tx}} - 30}{10}} \Delta t + \epsilon_{c_i} S_i(t) \right], \quad S_i(t) > 0, \quad (20)$$

where P_i^{tx} is the transmit power in dBm, ϵ_{c_i} is the class-dependent circuit and processing-energy coefficient in J/bit, and $S_i(t)$ is the successfully served traffic in bits. For $S_i(t) = 0$, $E_i^{\text{svc}}(t) = 0$. The adopted coefficients are $(\epsilon_0, \epsilon_1, \epsilon_2) = (1.0, 1.1, 1.5) \times 10^{-8}$ J/bit for safety, CAM, and CPM traffic, respectively.

$$\kappa(a(t)) = \begin{cases} 1.10, & a(t) = \text{handoff}, \\ 1, & \text{otherwise}, \end{cases} \quad (21)$$

where the handoff factor represents additional control and robust-allocation overhead. The episode-level vehicle-side communication energy is

$$E_{\text{ep}} = \sum_{t=1}^T \sum_{i \in \mathcal{V}} E_i^{\text{svc}}(t). \quad (22)$$

The episode-level vehicle-side communication energy is used as the energy metric in the composite system-cost formulation.

The episode throughput over T slots is

$$\Theta = \frac{\sum_{t=1}^T \sum_{i \in \mathcal{V}} S_i(t)}{T \Delta t}. \quad (23)$$

4. Problem Formulation

The scheduling problem is to determine an RSU-side policy that selects a feasible action at each slot while balancing reliability, timeliness, CPM usefulness, energy consumption, and RSU load. Let $\mathbf{o}(t)$ denote the observed system state at slot t , and let π be a scheduling policy that maps this observation to an action:

$$a(t) = \pi(\mathbf{o}(t)), \quad a(t) \in \mathcal{A}. \quad (24)$$

The objective is to minimize the expected finite-horizon admission-adjusted system cost:

$$\pi^* = \arg \min_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=1}^T C_{\text{sys}}(t) \right], \quad (25)$$

where $C_{\text{sys}}(t)$ is the slot-level composite cost. It captures delay, energy consumption, reliability loss, CPM freshness degradation, and RSU imbalance. The detailed normalized cost components are defined in the evaluation methodology.

The policy is constrained by blocking, interruption, and safety feasibility:

$$P_{\text{block}} \leq P_{\text{block}}^*, \quad (26)$$

$$P_{\text{int}} \leq P_{\text{int}}^*, \quad (27)$$

$$a(t) \in \mathcal{A}_{\text{safe}}(t), \quad \forall t, \quad (28)$$

where P_{block} and P_{int} denote the blocking and interruption probabilities, respectively. The thresholds P_{block}^* and P_{int}^* define the admissible operating limits, while $\mathcal{A}_{\text{safe}}(t) \subseteq \mathcal{A}$ denotes the set of actions satisfying the slot-level safety conditions.

The formulation defines the desired online scheduling objective but does not imply that a globally optimal policy can be obtained in closed form. The problem is difficult because the outcomes of actions depend on coupled queueing, wireless, mobility, CPM value, and RSU load dynamics. MoReSP therefore provides a structured approximate policy that performs regime-aware action scoring, safety-feasible decision correction, and episodic parameter adaptation.

5. Proposed MoReSP Framework

MoReSP is a structured online scheduling policy for the RSU-side decision problem formulated in Section 4. At each slot, the scheduler observes the current network state, derives pressure indicators, infers the mobility regime, assigns scores to the available actions in Equation (18), and executes a safety-filtered decision. The design is intentionally interpretable: each decision is shaped by measurable queue, channel, mobility, CPM, and RSU-load indicators rather than by a purely black-box mapping.

Let the normalized observation at slot t be

$$\mathbf{o}(t) = [q_{\text{act}}(t), q_{\text{old}}(t), q_{\text{avg}}(t), \gamma_{\text{avg}}(t), \zeta(t), \phi_0(t), \phi_1(t), \phi_2(t), \rho(t), v(t), \beta(t)], \quad (29)$$

where $q_{\text{act}}(t)$, $q_{\text{old}}(t)$, and $q_{\text{avg}}(t)$ denote active-queue pressure, oldest queue age, and average queue age, respectively. The terms $\gamma_{\text{avg}}(t)$ and $\zeta(t)$ denote recent SINR and success-rate indicators, $\phi_c(t)$ is the traffic fraction of class c , and $\rho(t)$, $v(t)$, and $\beta(t)$ are the density, normalized-speed, and braking indicators.

From $\mathbf{o}(t)$, MoReSP forms the feature vector

$$\mathbf{x}(t) = [U(t), R(t), F(t), G(t), M(t), E(t), C(t), H(t), L(t)], \quad (30)$$

where $U(t)$ is urgency pressure, $R(t)$ is reliability risk, $F(t)$ is fairness pressure, $G(t)$ is congestion pressure, $M(t)$ is mobility instability, $E(t)$ is emergency pressure, $C(t)$ is CPM pressure, $H(t)$ is handoff pressure, and $L(t)$ is RSU-load pressure. The reliability-risk feature is computed as

$$R(t) = \text{clip}_{[0,1]}(\alpha_1(1 - \zeta(t)) + \alpha_2\beta(t) + \alpha_3(1 - \gamma_{\text{avg}}(t)) + \alpha_4\rho(t) + \alpha_5I_{\text{win}}(t) + \alpha_6B_{\text{win}}(t)), \quad (31)$$

The normalized coefficients are set to $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6) = (0.30, 0.18, 0.12, 0.10, 0.20, 0.10)$. Accordingly, recent delivery failure and interruption history receive the largest contributions,

followed by braking activity, SINR degradation, traffic density, and recent blocking. where $I_{\text{win}}(t)$ and $B_{\text{win}}(t)$ denote recent-window interruption and blocking rates. The remaining pressure terms are computed using the same normalized design principle so that all components remain comparable within the scoring function.

The current traffic condition is assigned to one of four regimes:

$$\mathcal{R} = \{\text{sparse-stable, fast-flow, dense-congested, unstable-braking}\}. \quad (32)$$

The raw regime classifier $g(\cdot)$ uses the mobility, queue, and reliability indicators, while a smoothing operator $\mathcal{S}(\cdot)$ suppresses rapid switching:

$$r(t) = \mathcal{S}(g(\mathbf{o}(t))). \quad (33)$$

The regime variable changes the relative importance of reliability, mobility, congestion, and CPM delivery in the action-scoring process.

For each regime $r \in \mathcal{R}$, MoReSP maintains a parameter vector

$$\boldsymbol{\theta}_r = [w_u^r, w_{\text{rel}}^r, w_f^r, w_g^r, w_m^r, w_{\text{ch}}^r, \beta_{\text{co}}^r, \beta_{\text{ho}}^r, \beta_{\text{pr}}^r, \beta_{\text{dn}}^r], \quad (34)$$

where the w -terms weight urgency, reliability, fairness, congestion, mobility, and action churn, while the β -terms represent action-specific bonuses or penalties. For each action $a \in \mathcal{A}$, the score is

$$S_a(t) = f_a(\mathbf{x}(t), \boldsymbol{\theta}_{r(t)}) + B_a(r(t)) - \Omega_a(t), \quad (35)$$

where $f_a(\cdot)$ is the action-specific pressure function; $B_a(r(t))$ is the regime-dependent bias; and $\Omega_a(t)$ penalizes unnecessary action switching. The candidate action is selected as

$$a^{\text{cand}}(t) = \arg \max_{a \in \mathcal{A}} S_a(t). \quad (36)$$

The candidate action is then passed through a safety projection layer. Each action is assigned one of three feasibility states:

$$\sigma(a, t) \in \{\text{hard-feasible, soft-feasible, rejected}\}. \quad (37)$$

Let $\mathcal{A}_{\text{hard}}(t)$ and $\mathcal{A}_{\text{soft}}(t)$ denote the hard- and soft-feasible action sets. Define

$$\begin{aligned} \mathcal{A}_{\text{soft}}^<(t) &= \{a \in \mathcal{A}_{\text{soft}}(t) : \nu(a, t) < \nu_{\text{th}}\}, \\ \mathcal{A}_{\text{acc}}(t) &= \mathcal{A}_{\text{hard}}(t) \cup \mathcal{A}_{\text{soft}}^<(t). \end{aligned} \quad (38)$$

For compactness, let $\Pi_t(a) = \Pi(a, \mathcal{A}_{\text{acc}}(t))$. The safety-projected action is

$$a^{\text{safe}}(t) = \begin{cases} a^{\text{cand}}(t), & a^{\text{cand}}(t) \in \mathcal{A}_{\text{acc}}(t), \\ \Pi_t(a^{\text{cand}}(t)), & \text{otherwise.} \end{cases} \quad (39)$$

where $\nu(a, t)$ denotes the violation severity of action a , ν_{th} is the soft-acceptance threshold, and $\Pi_t(\cdot)$ returns the nearest accepted alternative. The safety layer uses success-rate, reliability-risk, congestion, mobility-instability, churn, and starvation thresholds. The soft-acceptance severity threshold is set to $\nu_{\text{th}} = 0.35$.

After each episode, MoReSP updates the regime-specific parameters using blocking, interruption, fairness, utilization, and CPM-delivery statistics. Let $\bar{P}_{\text{block}}$, $\bar{P}_{\text{int}}$, $\bar{F}$, $\bar{U}$, and $\bar{\Gamma}_{\text{cpm}}$ denote the corresponding episode-level quantities. The regime-dependent adaptation step is

$$\eta_r = \chi_r \eta_0, \quad (40)$$

where η_0 is the base step size, and χ_r increases adaptation strength in dense-congested and unstable-braking regimes. The main parameter updates are

$$w_u^r \leftarrow \text{clip}(w_u^r + \eta_r \alpha_b (\bar{P}_{\text{block}} - P_{\text{block}}^*), w_{\min}, w_{\max}), \quad (41)$$

$$w_{\text{rel}}^r \leftarrow \text{clip}(w_{\text{rel}}^r + \eta_r \alpha_r (\bar{P}_{\text{block}} - P_{\text{block}}^*), w_{\min}, w_{\max}), \quad (42)$$

and

$$w_m^r \leftarrow \text{clip}(w_m^r + \eta_r \alpha_m (\bar{P}_{\text{int}} - P_{\text{int}}^*), w_{\min}, w_{\max}). \quad (43)$$

The fixed update coefficients are $\alpha_b = 1.20$, $\alpha_r = 0.60$, and $\alpha_m = 0.90$. The base adaptation rate is $\eta_0 = 0.0875$, and the regime factors for sparse-stable, fast-flow, dense-congested, and unstable-braking conditions are 0.70, 0.60, 2.20, and 2.50, respectively. The remaining bonuses are adjusted according to the same episode-level feedback: excessive interruption reduces aggressive coexistence or preemption incentives, while insufficient CPM delivery increases coexistence and handoff incentives. The clipping bounds are $w_{\min} = 0.20$ and $w_{\max} = 2.50$, while the target limits are $P_{\text{block}}^* = 0.05$ and $P_{\text{int}}^* = 0.08$.

Algorithm 1 summarizes the MoReSP scheduling procedure.

Algorithm 1 MoReSP Scheduling Procedure

Require: Environment $\mathcal{E}$, action set $\mathcal{A}$, regime set $\mathcal{R}$, initial parameters $\{\theta_r\}_{r \in \mathcal{R}}$, safety thresholds, number of episodes E , slots per episode T

Ensure: Episode metrics and adapted regime-policy parameters

```

1: for  $e = 1, \dots, E$  do
2:   Reset environment, regime inferencer, safety layer, and short-window statistics
3:   for  $t = 1, \dots, T$  do
4:     Observe  $\mathbf{o}(t)$  and compute  $\mathbf{x}(t)$ 
5:     Infer smoothed regime  $r(t)$ 
6:     Compute  $S_a(t)$  for all  $a \in \mathcal{A}$ 
7:     Select  $a^{\text{cand}}(t) = \arg \max_{a \in \mathcal{A}} S_a(t)$ 
8:     Apply safety projection to obtain  $a^{\text{safe}}(t)$ 
9:     Execute  $a^{\text{safe}}(t)$ 
10:    Update short-window reliability, intervention, fairness, and starvation statistics
11:  end for
12:  Finalize episode metrics
13:  Adapt  $\{\theta_r\}_{r \in \mathcal{R}}$ 
14: end for
```

6. Simulation Setup and Evaluation Methodology

This section defines the experimental protocol used to evaluate MoReSP and the benchmark schemes. All simulations were conducted using a custom discrete-time V-IoT simulator implemented in Python 3.13. The simulator incorporates the multi-RSU edge environment, a NaSch-based mobility model, a stochastic packet-generation process, a V2X channel model, and packet-level scheduling operations described in Section 3. All methods are tested under the same simulator, traffic model, channel model, action space, random seeds, and post-processing pipeline. The evaluation uses five independent seeds, 300 episodes per seed, and 150 slots per episode. The main simulation parameters are summarized in Table 2.

MoReSP is compared with five benchmark schemes implemented under the same five-action scheduling space in Equation (18). AoCI-Heuristic emphasizes age and CPM freshness pressure, RSU-Coop emphasizes RSU load balancing and cooperative service, Handoff-Aware emphasizes mobility continuity, V2X-Priority emphasizes safety and latency-critical traffic, and ALTO-MAB represents lightweight adaptive online action selection. Comparison fairness is ensured by evaluating all methods under a common simulator configuration,

RSU topology, traffic and channel models, packet-generation process, action semantics, scenario settings, random-seed set, episode horizon, and metric-calculation pipeline. For each scenario–seed pair, every method is initialized independently under the same environmental configuration and evaluated over (300) episodes of (150) slots each. This common evaluation protocol enables a direct comparison of the alternative scheduling-policy designs under identical V-IoT operating conditions. These benchmarks are literature-inspired baseline families rather than exact numerical reproductions of individual prior studies.

Table 2. Main simulation parameters.

Parameter	Value/Description
Network model	RSU-assisted V-IoT with multi-RSU edge support
Number of RSUs	$B = 3$
Number of scheduled V-IoT sources	$V = 12$
NaSch mobility population	$N_{\text{mob}} = \text{round}(\rho_{\text{NaSch}} N_{\text{cell}})$
Communication direction	Vehicle $\rightarrow$ serving RSU uplink
RSU association	Nearest-RSU association on the circular road
Access-link distance	$d_{i,b_i(t)}(t) \geq 5$ m; no fixed communication-radius threshold
Delivery accounting	Successful reception at the serving RSU
Random braking probability	$p_{\text{slow}} = 0.20$
Transmit powers	Safety: 23 dBm; CAM: 20 dBm; CPM: 23 dBm
Effective service bandwidth and spectral-efficiency cap	$W = 20$ MHz; $R_{\text{max}} = 6$ bits/s/Hz
Service-migration overhead	Latency: 5 ms; energy: 1.5×10^{-4} J
Scheduling actions	{deny, grant, preempt, coexist, handoff}
Slot duration	$\Delta t = 1$ ms
Traffic classes	Safety/event-driven V2X, CAM, and CPM
Sampling rates	Safety: 2 Hz; CAM: 10 Hz; CPM: 5 Hz
Deadlines	Safety: 20 ms; CAM: 100 ms; CPM: 50 ms
Reliability targets	Safety: 0.999; CAM: 0.95; CPM: 0.98
Packet sizes	Safety: 3200 bits; CAM: 2400 bits; CPM: 9000 bits
Per-bit circuit/processing energy	Safety: 1.0×10^{-8} J/bit; CAM: 1.1×10^{-8} J/bit; CPM: 1.5×10^{-8} J/bit
Mobility model	Nagel–Schreckenberg circular-road model
Road configuration	$N_{\text{cell}} = 120$, $l_{\text{cell}} = 7.5$ m, $L_{\text{road}} = 900$ m
Maximum speed	$v_{\text{max}} = 5$ cells/slot
Carrier frequency	$f_{\text{car}} = 5.9$ GHz
Noise floor	−94 dBm
Shadowing and fading	Shadowing standard deviation: 3 dB; fast-fading standard deviation: 2 dB
Evaluation seeds	Five independent random seeds
Simulation horizon	300 episodes and 150 slots per episode

The evaluation considers three main scenario sweeps and three diagnostic sweeps. The vehicle-load scenario varies the arrival-scale parameter as

$$\lambda_{\text{scale}} \in \{0.8, 1.2, 1.6, 2.0, 2.5, 3.0\}. \quad (44)$$

The traffic-density scenario varies the NaSch density as

$$\rho_{\text{NaSch}} \in \{0.10, 0.20, 0.35, 0.50, 0.65, 0.80\}. \quad (45)$$

The RSU-capacity scenario varies the RSU capacity scale as

$$\kappa_{\text{RSU}} \in \{0.50, 0.75, 1.00, 1.25, 1.50\}. \quad (46)$$

The diagnostic sweeps vary the RSU-to-RSU cooperation rate, CPM-load scale, and block-budget scale over {0.50, 0.75, 1.00, 1.25, 1.50}. The main figures report vehicle-

load, NaSch-density, and RSU-capacity results because these directly stress packet arrival pressure, mobility-density effects, and infrastructure availability.

Let N_{arr} , N_{blk} , N_{pre} , and N_{del} denote the number of packet arrivals, blocked arrivals, preempted transmissions, and delivered packets, respectively. The blocking and interruption probabilities are

$$P_{\text{block}} = \frac{N_{\text{blk}}}{N_{\text{arr}}}, \quad (47)$$

and

$$P_{\text{int}} = \frac{N_{\text{pre}}}{\max(N_{\text{arr}}, 1)}. \quad (48)$$

The raw packet delivery ratio (PDR) is

$$\text{PDR}_{\text{raw}} = \frac{N_{\text{del}}}{N_{\text{arr}}}, \quad (49)$$

and the raw CPM delivery ratio, CPM_{raw} , is computed using delivered and generated CPM packets.

To prevent blocked or interrupted traffic from being hidden by raw delivery statistics, the evaluation uses conservative admission-adjusted delivery indices. The service-admission factor is

$$\eta_{\text{svc}} = (1 - P_{\text{block}})(1 - P_{\text{int}}). \quad (50)$$

The effective packet delivery and CPM delivery indices are

$$\text{PDR}_{\text{eff}} = \text{PDR}_{\text{raw}} \eta_{\text{svc}} \quad (51)$$

and

$$\text{CPM}_{\text{eff}} = \text{CPM}_{\text{raw}} \eta_{\text{svc}}. \quad (52)$$

These indices are not conventional physical-layer PDRs; they are admission-adjusted service-reliability measures that penalize delivery failure, blocking, and interruption. The admission-adjusted reliability loss is

$$L_{\text{rel}}^{\text{adm}} = 1 - \text{PDR}_{\text{eff}}. \quad (53)$$

The main system-level metric is the admission-adjusted system cost. It combines normalized delay, normalized energy, reliability penalty, CPM freshness cost, and RSU-imbalance cost. The delay component is

$$C_{\text{delay}} = \text{clip}\left(\frac{D_{\text{avg}}}{5}, 0, 1\right), \quad (54)$$

where D_{avg} is the average delay in milliseconds. The CPM freshness component is

$$C_{\text{cpm}} = 1 - \text{CPM}_{\text{eff}}. \quad (55)$$

The reliability component is

$$C_{\text{rel}} = \text{clip}\left(0.55L_{\text{rel}}^{\text{adm}} + 0.25P_{\text{block}} + 0.20P_{\text{int}}, 0, 1\right). \quad (56)$$

Let $\mathcal{D}$ denote the set of scenario–scheme–seed records in the reporting table. The normalized vehicle-side communication-energy component is

$$C_{\text{energy}} = \frac{E_{\text{ep}} - \min_{z \in \mathcal{D}} E_{\text{ep}}^{(z)}}{\max_{z \in \mathcal{D}} E_{\text{ep}}^{(z)} - \min_{z \in \mathcal{D}} E_{\text{ep}}^{(z)}}, \quad (57)$$

where $E_{ep}^{(z)}$ is the mean episode-level energy for record z . If the denominator is zero, C_{energy} is set to zero. The RSU-imbalance component C_{rsu} is normalized using the same min-max procedure. The final system cost is

$$C_{sys} = 0.20C_{delay} + 0.10C_{energy} + 0.35C_{rel} + 0.25C_{cpm} + 0.10C_{rsu}. \quad (58)$$

The nominal weight vector is $(w_D, w_E, w_R, w_C, w_I) = (0.20, 0.10, 0.35, 0.25, 0.10)$, where the weights sum to one. These normalized design weights assign the greatest importance to admission-adjusted reliability and CPM freshness, while retaining delay, energy, and RSU-load balance as contributing system-level factors. Their influence is further examined through the weight-sensitivity analysis in Section 7. The largest weights are assigned to reliability and CPM freshness because the target application is reliability-sensitive cooperative V-IoT scheduling rather than throughput-only access. All reported curves show the mean across five seeds. Error bars indicate two-sided 95% Student's t confidence intervals across five independent seeds. Cost-decomposition plots report the five weighted components of C_{sys} , and all figures use a fixed scheme-color convention across scenarios.

7. Results and Discussion

This section evaluates MoReSP under RSU-capacity scaling, NaSch-density variation, and vehicle-load variation against AoCI-Heuristic, RSU-Coop, Handoff-Aware, V2X-Priority, and ALTO-MAB. Lower values are preferred for admission-adjusted system cost, reliability loss, and delay, whereas higher values are preferred for effective packet and CPM delivery. Unless otherwise stated, all numerical values are five-seed means corresponding to the plotted points and are reported to three decimal places.

Figure 2 shows the effect of RSU-capacity scaling. MoReSP obtains the lowest admission-adjusted system cost throughout the sweep. At $\kappa_{RSU} = 0.5$, MoReSP records a cost of 0.254, compared with 0.412 for the best baseline, Handoff-Aware, corresponding to a 38.4% reduction. At nominal capacity, $\kappa_{RSU} = 1.0$, MoReSP reduces the cost to 0.227, compared with 0.402 for RSU-Coop, giving a 43.6% reduction. At $\kappa_{RSU} = 1.5$, MoReSP records 0.219, whereas RSU-Coop remains at 0.404, corresponding to a 45.8% reduction. These results show that MoReSP uses additional RSU capacity more effectively than the baselines.

At nominal RSU capacity, MoReSP also achieves a lower admission-adjusted reliability loss of 0.193, while RSU-Coop, AoCI-Heuristic, Handoff-Aware, V2X-Priority, and ALTO-MAB record 0.514, 0.538, 0.542, 0.544, and 0.555, respectively. The effective packet delivery ratio of MoReSP is 0.807, compared with 0.486 for RSU-Coop. Similarly, the effective CPM delivery ratio is 0.787 for MoReSP and 0.475 for RSU-Coop. The cost decomposition in Figure 2d further shows that MoReSP reduces the total cost mainly through lower reliability and CPM penalties. Its total cost of 0.227 consists of delay cost 0.021, energy cost 0.047, reliability penalty 0.037, CPM freshness cost 0.053, and RSU-imbalance cost 0.069. In contrast, RSU-Coop has a total cost of 0.402, with reliability and CPM penalties of 0.133 and 0.131, respectively.

The reduction in these two components follows from the joint use of queue, channel, mobility, CPM, and RSU-load indicators in the decision process. When recent delivery success, SINR, density, braking activity, or short-window blocking and interruption rates indicate elevated service risk, MoReSP adjusts the action ranking using its reliability, congestion, and mobility pressures. The subsequent safety projection preserves only feasible actions under the prevailing operating condition, thereby reducing ineffective scheduling decisions that contribute to blocking and interruption. For CPM traffic, the CPM-pressure and value-of-information indicators maintain the importance of useful perception

packets during action selection. Consequently, improved successful service and lower admission disruption jointly reduce reliability and CPM penalties.

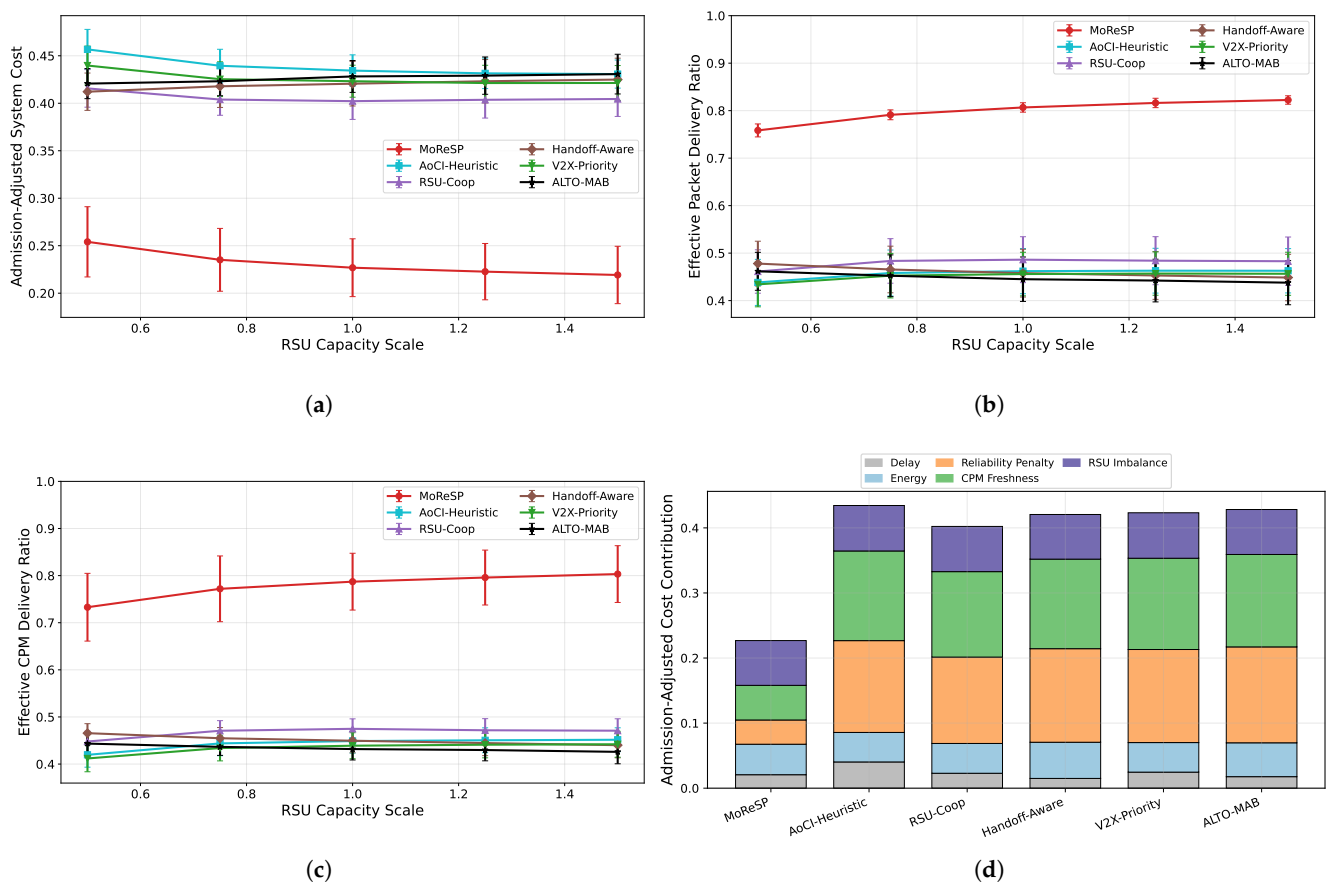


Figure 2. Impact of RSU-capacity scaling on system performance. (a) Admission-adjusted system cost versus RSU capacity scale. (b) Effective packet delivery ratio versus RSU capacity scale. (c) Effective CPM delivery ratio versus RSU capacity scale. (d) Cost-component decomposition at nominal RSU capacity, $\kappa_{RSU} = 1.0$.

Figure 3 evaluates traffic-density variation. In Figure 3a, MoReSP gives the lowest system cost at all density values. At $\rho_{NaSch} = 0.1$, MoReSP achieves 0.116, compared with 0.476 for RSU-Coop, giving a 75.6% reduction. At $\rho_{NaSch} = 0.5$, MoReSP records 0.295, compared with 0.404 for RSU-Coop, corresponding to a 27.0% reduction. At $\rho_{NaSch} = 0.8$, MoReSP records 0.424, while the best baseline, ALTO-MAB, records 0.471, giving a 10.0% reduction. The smaller margin at high density is expected because contention and mobility instability become dominant.

Figure 3b confirms that MoReSP preserves reliability more effectively as density increases. At $\rho_{NaSch} = 0.1$, its reliability loss is 0.022, compared with 0.705 for RSU-Coop. At $\rho_{NaSch} = 0.5$, MoReSP records 0.330, while Handoff-Aware and RSU-Coop record 0.531 and 0.539, respectively. At $\rho_{NaSch} = 0.8$, MoReSP records 0.555, compared with 0.628 for Handoff-Aware, giving an 11.6% reduction. Figure 3c shows that MoReSP does not always minimize delay. At $\rho_{NaSch} = 0.8$, MoReSP delay is 3.291 ms, while RSU-Coop and ALTO-MAB record 2.699 ms and 2.742 ms, respectively. This indicates a deliberate reliability-delay trade-off: MoReSP accepts moderate delay growth to reduce the likelihood of failed or ineffective service decisions.

Figure 3d shows the cost breakdown at $\rho_{NaSch} = 0.5$ and supports this interpretation. MoReSP records a total cost of 0.295, with delay cost 0.062, energy cost 0.038, reliability penalty 0.064, CPM freshness cost 0.093, and RSU-imbalance cost 0.038. RSU-Coop

records 0.404, mainly due to higher reliability and CPM penalties of 0.132 and 0.141. Thus, the density-sweep advantage of MoReSP is mainly due to reliability and CPM preservation rather than delay minimization alone.

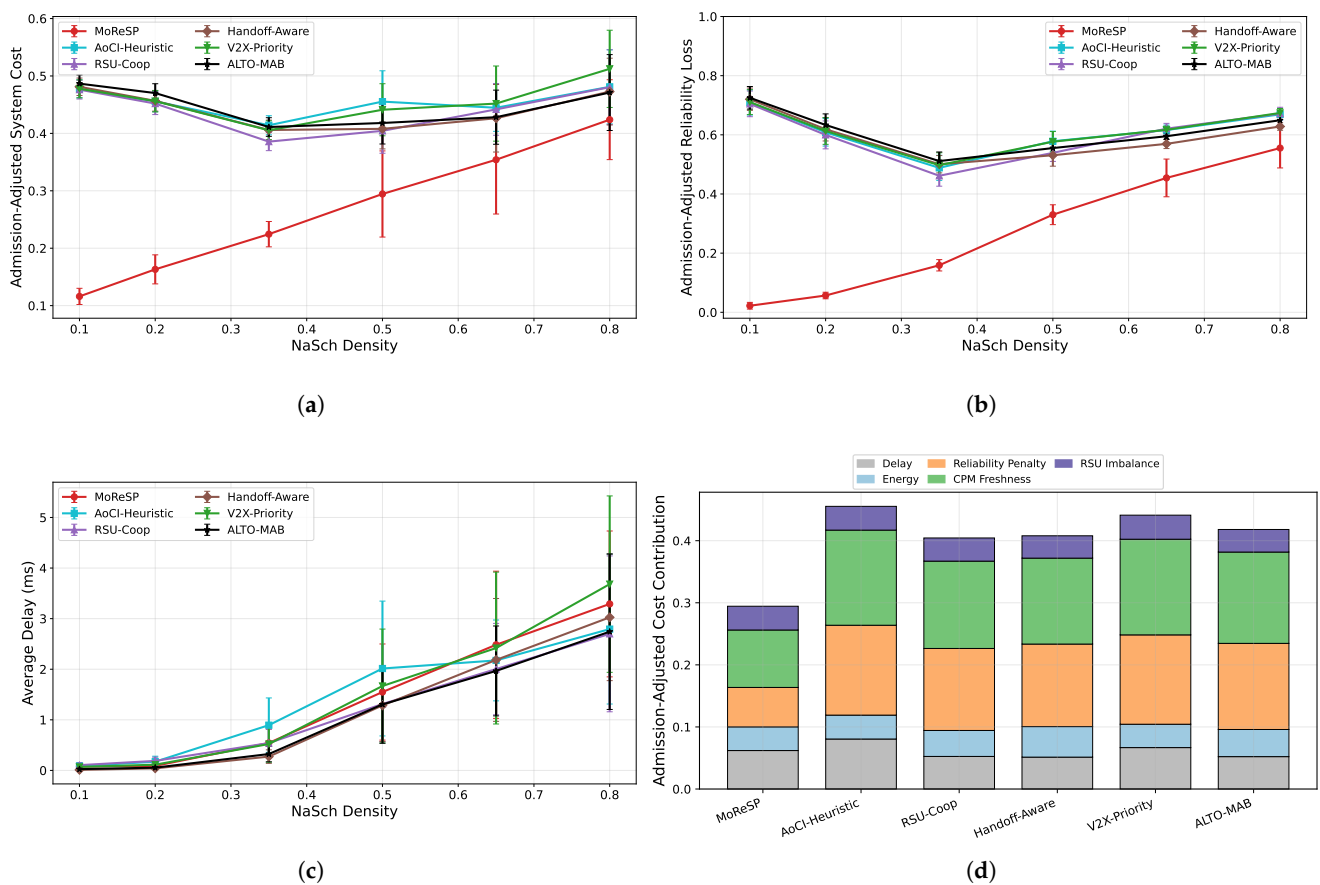


Figure 3. Impact of NaSch density on system performance. (a) Admission-adjusted system cost. (b) Admission-adjusted reliability loss. (c) Average delay. (d) Cost-component decomposition at $\rho_{NaSch} = 0.5$.

This trend is consistent with the mobility-regime mechanism. As traffic density increases and vehicle motion becomes less stable, the regime classifier changes the relative importance assigned to reliability, congestion, and mobility conditions in the action-scoring process. The safety projection further evaluates whether coexistence remains feasible under the observed reliability risk, recent success rate, mobility instability, and congestion pressure. Hence, MoReSP accepts moderate delay growth at high density when necessary to avoid unreliable service decisions, while preserving CPM delivery when the prevailing state supports coexistence or handoff scheduling.

Figure 4 evaluates vehicle-load variation. In Figure 4a, MoReSP remains the lowest-cost scheme across the full arrival-scale sweep. At $\lambda_{scale} = 0.8$, MoReSP records 0.114, compared with 0.483 for RSU-Coop, giving a 76.4% reduction. At $\lambda_{scale} = 2.0$, MoReSP records 0.187, compared with 0.411 for RSU-Coop, corresponding to a 54.4% reduction. At $\lambda_{scale} = 3.0$, MoReSP records 0.271, while RSU-Coop records 0.407, giving a 33.5% reduction. The pronounced reduction at light load reflects the admission-adjusted composite metric, where MoReSP incurs lower blocking, interruption, and CPM-freshness penalties than RSU-Coop.

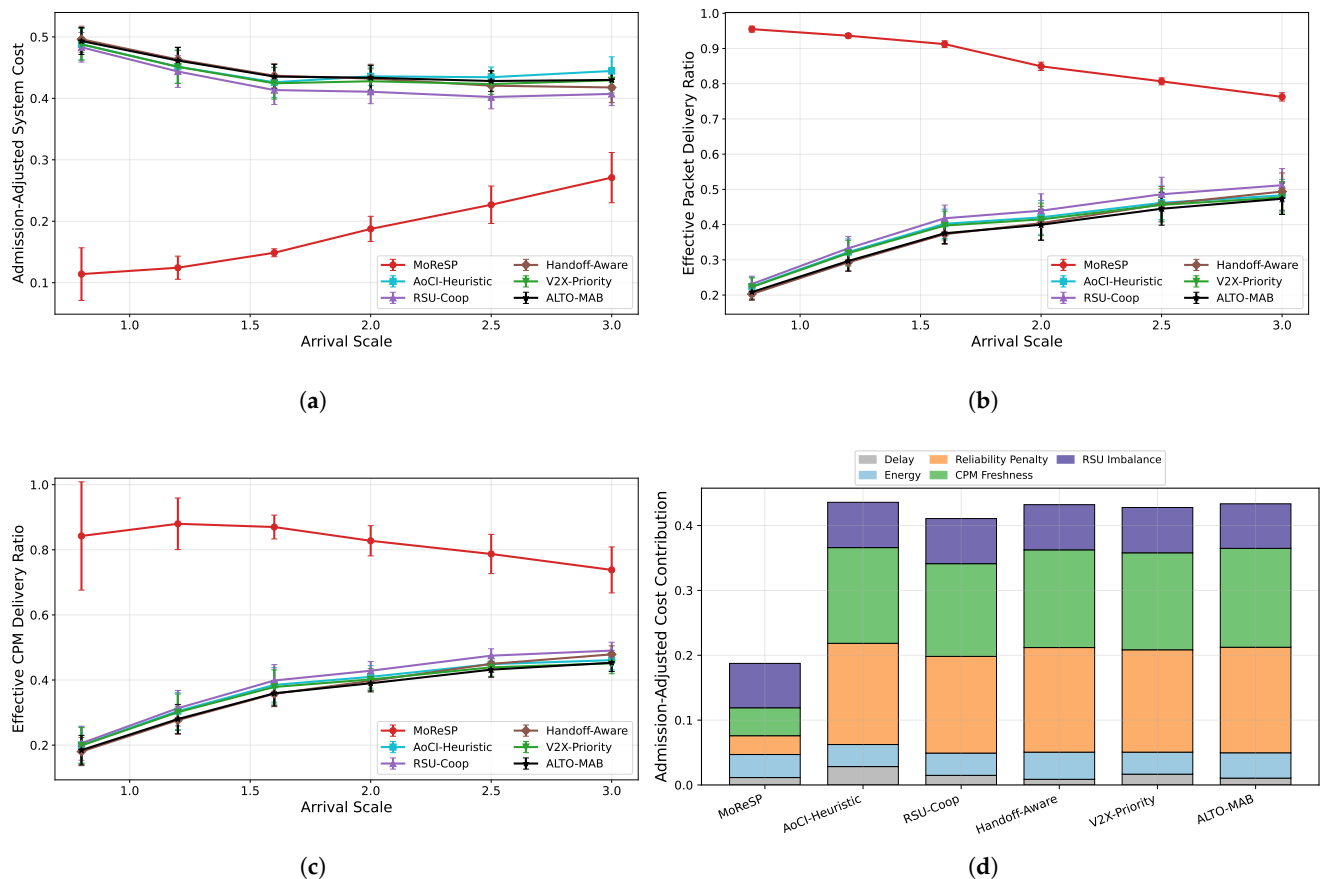


Figure 4. Impact of vehicle-load variation through arrival-scale adjustment. (a) Admission-adjusted system cost. (b) Effective packet delivery ratio. (c) Effective CPM delivery ratio. (d) Cost-component decomposition at $\lambda_{scale} = 2.0$.

Figure 4b,c show that the lower system cost of MoReSP is accompanied by stronger effective delivery. At $\lambda_{scale} = 2.0$, MoReSP achieves an effective packet delivery ratio of 0.849, compared with 0.440 for RSU-Coop. At the same load, MoReSP achieves an effective CPM delivery ratio of 0.828, compared with 0.428 for RSU-Coop. At $\lambda_{scale} = 3.0$, MoReSP maintains effective packet and CPM delivery ratios of 0.763 and 0.738, while RSU-Coop records 0.512 and 0.490, respectively.

Figure 4d shows the cost-component decomposition at $\lambda_{scale} = 2.0$. MoReSP has a total cost of 0.187, with delay cost 0.011, energy cost 0.036, reliability penalty 0.029, CPM freshness cost 0.043, and RSU-imbalance cost 0.068. RSU-Coop records a total cost of 0.411, mainly because its reliability and CPM penalties increase to 0.149 and 0.143. AoCI-Heuristic, Handoff-Aware, V2X-Priority, and ALTO-MAB record total costs of 0.436, 0.432, 0.428, and 0.434, respectively.

These values indicate that the high-load gain of MoReSP is primarily due to improved reliability and CPM delivery under admission pressure. Under increased arrival load, queue growth, and recent blocking or interruption events increase the urgency and reliability-risk pressures used by the policy. Episodic parameter adaptation then strengthens the emphasis on reliability and mobility protection, while a low CPM delivery ratio increases the preference for CPM-supporting coexistence and handoff decisions. This closed-loop adjustment improves the service-admission factor and the timely delivery of CPM traffic, which directly lowers the reliability and CPM components of the composite cost.

MoReSP achieves the lowest admission-adjusted system cost in all evaluated scenarios. Its largest gains appear under vehicle-load stress, where it reduces cost by 54.4% at

$\lambda_{\text{scale}} = 2.0$ and maintains nearly twice the effective packet and CPM delivery ratios of the strongest baseline. Under RSU-capacity scaling, it reduces cost by 43.6% at nominal capacity and 45.8% at the highest capacity point. Under NaSch-density variation, it remains the lowest-cost and lowest-reliability-loss scheme, although it incurs moderate delay growth at high density. These results show that reliable V-IoT scheduling requires coordinated admission control, mobility awareness, CPM preservation, reliability-risk management, and RSU-load balancing.

The results show that the observed cost reduction is produced by the coordinated use of reliability-risk estimation, mobility-regime inference, CPM-aware action scoring, safety-feasible scheduling, and episodic adaptation. These mechanisms jointly reduce blocking and interruption while preserving effective packet and CPM delivery under changing RSU capacity, traffic density, and vehicle-load conditions.

Sensitivity to Composite-Cost Weights

To assess the sensitivity of the results to the weighting coefficients in Equation (58), the composite cost was recomputed from the same seed-level delay, energy, reliability, CPM, and RSU-imbalance components using alternative normalized weight profiles. The component definitions and normalization procedure were retained, while only the aggregation weights were changed. The analysis covers the (17) operating points in the three primary sweeps: vehicle-load variation, NaSch-density variation, and RSU-capacity scaling. In Table 3, w_D , w_E , w_R , w_C , and w_I denote the weights of delay, energy, reliability, CPM, and RSU-imbalance components, respectively. The mean and minimum reductions are calculated relative to the lowest-cost baseline at each operating point.

Table 3. Composite-cost sensitivity under alternative weight profiles.

Profile	$(w_D, w_E, w_R, w_C, w_I)$	MoReSP Rank-1 Points	Mean Reduction (%)	Minimum Reduction (%)
Nominal	(0.20, 0.10, 0.35, 0.25, 0.10)	17/17	46.7	10.0
Reliability emphasis	(0.15, 0.05, 0.45, 0.25, 0.10)	17/17	51.6	12.2
CPM emphasis	(0.15, 0.05, 0.30, 0.40, 0.10)	17/17	50.2	9.5
Latency–energy emphasis	(0.35, 0.20, 0.20, 0.15, 0.10)	17/17	33.2	2.3
RSU-load emphasis	(0.15, 0.10, 0.30, 0.20, 0.25)	17/17	35.9	10.0

As shown in Table 3, MoReSP achieves the lowest mean composite cost at all (17) operating points under every tested weight profile. Its mean reduction relative to the best baseline ranges from (33.2%) under latency–energy emphasis to (51.6%) under reliability emphasis. The smallest observed reduction is (2.3%), obtained at $\rho_{\text{NaSch}} = 0.8$ under the latency–energy profile. These results indicate that the main conclusion is robust to meaningful changes in the relative importance assigned to delay, energy, reliability, CPM delivery, and RSU-load balance.

8. Conclusions

This paper presented MoReSP, a mobility- and reliability-aware scheduling framework for RSU-assisted V-IoT networks. MoReSP combines mobility-regime inference, structured action scoring, safety projection, and episodic adaptation to support admission-sensitive scheduling for heterogeneous safety, CAM, and CPM traffic. The framework was evaluated under RSU-capacity scaling, NaSch-density variation, and vehicle-load scaling using admission-adjusted metrics that penalize excessive blocking and interruption. The results showed that MoReSP consistently achieved the lowest admission-adjusted system cost among all compared schemes. At nominal RSU capacity, MoReSP reduced the system

cost from 0.402 for the best baseline to 0.227, corresponding to a 43.6% reduction. Under vehicle-load stress, MoReSP reduced the cost by 54.4% at arrival scale 2.0 and by 33.5% at arrival scale 3.0. It also maintained stronger effective delivery performance, achieving effective packet and CPM delivery ratios of 0.849 and 0.828 at an arrival scale of 2.0, compared with 0.440 and 0.428 for the best baseline. The results confirm that reliable V-IoT scheduling requires joint control of admission, mobility condition, reliability risk, CPM delivery, and RSU load. Future work will extend MoReSP toward distributed multi-RSU coordination, constrained reinforcement learning, graph-based adaptation, trace-driven mobility validation, standardized 5G/6G sidelink channels, and hardware-in-the-loop edge scheduling.

Author Contributions: Conceptualization, M.F.S. and A.I.; methodology, M.F.S. and A.I.; software, M.F.S.; validation, M.F.S. and A.I.; formal analysis, M.F.S. and A.I.; investigation, M.F.S. and A.I.; writing—original draft preparation, M.F.S. and A.I.; writing—review and editing, A.I.; supervision, A.I. All authors have read and agreed to the published version of the manuscript.

Funding: 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. KFU263571].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

Acknowledgments: The authors acknowledge the support of their respective institutions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chen, X.; Wu, C.; Chen, T.; Zhang, H.; Liu, Z.; Zhang, Y.; Bennis, M. Age of information aware radio resource management in vehicular networks: A proactive deep reinforcement learning perspective. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 2268–2281. [\[CrossRef\]](#)
2. Liu, L.; Chen, C.; Pei, Q.; Maharjan, S.; Zhang, Y. Vehicular edge computing and networking: A survey. *Mob. Netw. Appl.* **2021**, *26*, 1145–1168.
3. Xie, Y.; Wu, Q.; Fan, P. Digital twin vehicular edge computing network: Task offloading and resource allocation. In *Proceedings of the 2024 7th International Conference on Information Communication and Signal Processing (ICICSP)*; IEEE: New York, NY, USA, 2024; pp. 1137–1141.
4. Lin, Y.; Bao, J.; Zhang, Y.; Li, J.; Shu, F.; Hanzo, L. Privacy-preserving joint edge association and power optimization for the internet of vehicles via federated multi-agent reinforcement learning. *IEEE Trans. Veh. Technol.* **2023**, *72*, 8256–8261.
5. Ren, Q.; Abbasi, O.; Kurt, G.K.; Yanikomeroglu, H.; Chen, J. Handoff-aware distributed computing in high altitude platform station (HAPS)-assisted vehicular networks. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 8814–8827. [\[CrossRef\]](#)
6. 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*, 6380–6391.
7. Wu, Q.; Wang, W.; Fan, P.; Fan, Q.; Wang, J.; Letaief, K.B. URLLC-awared resource allocation for heterogeneous vehicular edge computing. *IEEE Trans. Veh. Technol.* **2024**, *73*, 11789–11805. [\[CrossRef\]](#)
8. Khan, N.; Coleri, S. Event-triggered reinforcement learning based joint resource allocation for ultra-reliable low-latency V2X communications. *IEEE Trans. Veh. Technol.* **2024**, *73*, 16991–17006. [\[CrossRef\]](#)
9. Liu, Y.C.; Tian, J.; Glaser, N.; Kira, Z. When2com: Multi-agent perception via communication graph grouping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 13–19 June 2020; pp. 4106–4115.
10. Hu, Y.; Fang, S.; Lei, Z.; Zhong, Y.; Chen, S. Where2comm: Communication-efficient collaborative perception via spatial confidence maps. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 4874–4886. [\[CrossRef\]](#)
11. Ren, S.; Lei, Z.; Wang, Z.; Dianati, M.; Wang, Y.; Chen, S.; Zhang, W. Interruption-aware cooperative perception for V2X communication-aided autonomous driving. *IEEE Trans. Intell. Veh.* **2024**, *9*, 4698–4714.

12. Xiang, H.; Zheng, Z.; Xia, X.; Xu, R.; Gao, L.; Zhou, Z.; Han, X.; Ji, X.; Li, M.; Meng, Z.; et al. V2X-Real: A large-scale dataset for vehicle-to-everything cooperative perception. In Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024; pp. 455–470.
13. Sun, Y.; Guo, X.; Song, J.; Zhou, S.; Jiang, Z.; Liu, X.; Niu, Z. Adaptive learning-based task offloading for vehicular edge computing systems. *IEEE Trans. Veh. Technol.* **2019**, *68*, 3061–3074. [\[CrossRef\]](#)
14. He, Q.; Dán, G.; Fodor, V. Joint assignment and scheduling for minimizing age of correlated information. *IEEE/ACM Trans. Netw.* **2019**, *27*, 1887–1900. [\[CrossRef\]](#)
15. Yin, B.; Zhang, S.; Cheng, Y. Application-oriented scheduling for optimizing the age of correlated information: A deep-reinforcement-learning-based approach. *IEEE Internet Things J.* **2020**, *7*, 8748–8759. [\[CrossRef\]](#)
16. Maatouk, A.; Kriouile, S.; Assaad, M.; Ephremides, A. The age of incorrect information: A new performance metric for status updates. *IEEE/ACM Trans. Netw.* **2020**, *28*, 2215–2228. [\[CrossRef\]](#)
17. Qiao, G.; Leng, S.; Zhang, K.; He, Y. Collaborative task offloading in vehicular edge multi-access networks. *IEEE Commun. Mag.* **2018**, *56*, 48–54. [\[CrossRef\]](#)
18. Zhang, K.; Cao, J.; Zhang, Y. Adaptive digital twin and multiagent deep reinforcement learning for vehicular edge computing and networks. *IEEE Trans. Ind. Inform.* **2021**, *18*, 1405–1413. [\[CrossRef\]](#)
19. Cao, B.; Li, Z.; Liu, X.; Lv, Z.; He, H. Mobility-aware multiobjective task offloading for vehicular edge computing in digital twin environment. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 3046–3055. [\[CrossRef\]](#)
20. Wang, J.; Hu, J.; Min, G.; Zomaya, A.Y.; Georgalas, N. Fast adaptive task offloading in edge computing based on meta reinforcement learning. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *32*, 242–253. [\[CrossRef\]](#)
21. Tahir, N.; Parasuraman, R.; Sun, H. Communication-aware consistent edge selection for mobile users and autonomous vehicles. In *Proceedings of the 2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*; IEEE: New York, NY, USA, 2024; pp. 1–7.
22. Ji, M.; Wu, Q.; Fan, P.; Cheng, N.; Chen, W.; Wang, J.; Letaief, K.B. Graph neural networks and deep reinforcement learning-based resource allocation for V2X communications. *IEEE Internet Things J.* **2024**, *12*, 3613–3628. [\[CrossRef\]](#)
23. Shan, M.; Narula, K.; Worrall, S.; Wong, Y.F.; Perez, J.S.B.; Gray, P.; Nebot, E. A novel probabilistic V2X data fusion framework for cooperative perception. In *Proceedings of the 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*; IEEE: New York, NY, USA, 2022; pp. 2013–2020.
24. Xu, R.; Tu, Z.; Xiang, H.; Shao, W.; Zhou, B.; Ma, J. CoBEVT: Cooperative bird’s eye view semantic segmentation with sparse transformers. *arXiv* **2022**, arXiv:2207.02202.
25. Xiang, H.; Zheng, Z.; Xia, X.; Zhao, S.Z.; Gao, L.; Zhou, Z.; Cai, T.; Zhang, Y.; Ma, J. V2X-ReaLO: An open online framework and dataset for cooperative perception in reality. *arXiv* **2025**, arXiv:2503.10034.
26. Xu, R.; Xiang, H.; Xia, X.; Han, X.; Li, J.; Ma, J. OPV2V: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication. In *Proceedings of the 2022 International Conference on Robotics and Automation (ICRA)*; IEEE: New York, NY, USA, 2022; pp. 2583–2589.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.