

PHY-Aware MCS Adaptation for NR V2X Mode 2 with ML-Driven Resource Allocation

ANNU ANNU

ee21resch01010@iith.ac.in

Indian Institute of Technology Hyderabad

Prof. Rajalakshmi Pachamuthu

Indian Institute of Technology Hyderabad

Research Article

Keywords: NR V2X, sidelink, Mode 2, resource allocation, machine learning, reinforcement learning, MCS adaptation, analytical framework

Posted Date: September 19th, 2025

DOI: <https://doi.org/10.21203/rs.3.rs-7192344/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Wireless Personal Communications on October 7th, 2026. See the published version at <https://doi.org/10.1007/s11277-026-12237-4>.

PHY-Aware MCS Adaptation for NR V2X Mode 2 with ML-Driven Resource Allocation

ANNU^{1*} and Prof. Rajalakshmi Pachamuthu¹

^{1*}Department of Electrical Engineering, Indian Institute of Technology
Hyderabad, Kandi, Sangareddy, 502284, Telangana, India.

*Corresponding author(s). E-mail(s): ee21resch01010@iith.ac.in;
Contributing authors: raji@ee.iith.ac.in;

Abstract

This paper proposes a unified framework to enhance NR V2X sidelink Mode 2 by combining PHY-aware MCS adaptation with reinforcement learning (RL)-based resource allocation. In out-of-coverage scenarios, vehicles select MCS based on real-time SINR using thresholds from the UWICORE dataset, while an RL agent optimizes semi-persistent scheduling to avoid collisions. Analytical models for packet delivery, collision probability, and spectral efficiency are developed and validated using WiLabV2X simulations. Results demonstrate improved PDR, reduced collisions, and higher energy and spectral efficiency over standard SPS, particularly in dense traffic, aligning with 3GPP TR 37.885 for safety-critical use.

Keywords: NR V2X, sidelink, Mode 2, resource allocation, machine learning, reinforcement learning, MCS adaptation, analytical framework

1 Introduction

Vehicular communication in 5G New Radio Vehicle-to-Everything (NR V2X) supports advanced connected and autonomous driving use cases with stringent performance demands [1, 2]. In this framework, sidelink communication enables direct, low-latency, and reliable vehicle-to-vehicle messaging without constant reliance on infrastructure [3]. Introduced in 3GPP Release 16 as an evolution of LTE-based C-V2X, NR V2X addresses higher throughput, lower latency, and enhanced reliability requirements. It incorporates flexible numerologies, higher modulation orders, MIMO, and advanced error correction to enhance data rate and spectral efficiency [1]. Additionally, it

introduces new modes (unicast/groupcast), sidelink HARQ feedback, and resource allocation mechanisms optimized for vehicular communication.

Despite these enhancements, major **challenges** persist in NR V2X sidelink resource management, particularly in distributed (out-of-coverage) scenarios. In *Mode 2*, vehicles autonomously select radio resources from a shared pool under high mobility and without central coordination. This is hindered by dynamic topology and channel conditions, often causing packet collisions and unreliable QoS when multiple vehicles contend for the same spectrum [4].

One promising direction to enhance Mode 2 performance is **PHY-aware dynamic MCS adaptation**. The Modulation and Coding Scheme (MCS) directly influences transmission range, data rate, and resource usage in NR V2X [4, 5]. In current standards, sidelink messages often use a fixed, robust MCS to ensure basic reliability in worst-case conditions, but this can under-utilize capacity when channels are good. Conversely, using a high MCS can reduce air-time and resource occupancy, alleviating channel congestion, but may fail under poor radio conditions [5]. Therefore, adapting the MCS to the real-time physical (PHY) layer conditions (path loss, fading, interference) can provide a better trade-off between reliability and efficiency. A *PHY-aware* approach would dynamically select the most suitable MCS for each packet based on instantaneous signal quality, target error rates, and network load.

An emerging direction is the use of **machine learning (ML)**, particularly reinforcement learning (RL), for intelligent resource management. Traditional Mode 2 employs sensing-based SPS with heuristic rules, which struggle in dynamic or non-stationary environments. In contrast, ML frameworks learn to optimize resource selection, collision avoidance, and parameter tuning. RL agents, for instance, can adapt by observing collisions and avoiding repeated conflicts, enabling distributed coordination. Recent works show RL improves adaptability over fixed schemes in C-V2X and NR V2X [5, 6].

This paper proposes NR V2X Mode 2 resource allocation via joint PHY-aware MCS adaptation and RL-based scheduling. Key contributions include: (i) an analytical framework modeling PDR, collisions, and resource use; (ii) a decentralized RL scheduler optimizing resource/MCS selection; (iii) SNR-to-MCS mappings ensuring BLER-constrained reliability; (iv) system-level validation using WiLabV2X [7]; and (v) significant gains in PDR, efficiency, and energy use, with broader ITS implications. The paper is organized as follows: Section II reviews related work and gaps; Section III defines the system model and problem; Section IV presents the proposed RL-SPS and MCS scheme; Section V provides theoretical analysis; Section VI details the simulation setup; Section VII discusses results; Section VIII concludes and outlines future work.

2 Literature Review

NR V2X Sidelink and Mode 2 – The NR V2X sidelink, introduced in 3GPP Release 16, builds upon the PC5 interface from LTE-V2X but with substantial enhancements in the physical layer and protocol design [1, 3]. Mode 2 operation, which allows vehicles to autonomously select resources out-of-coverage, is fundamentally inherited from the LTE-V2X Mode 4 concept (sensing-based distributed

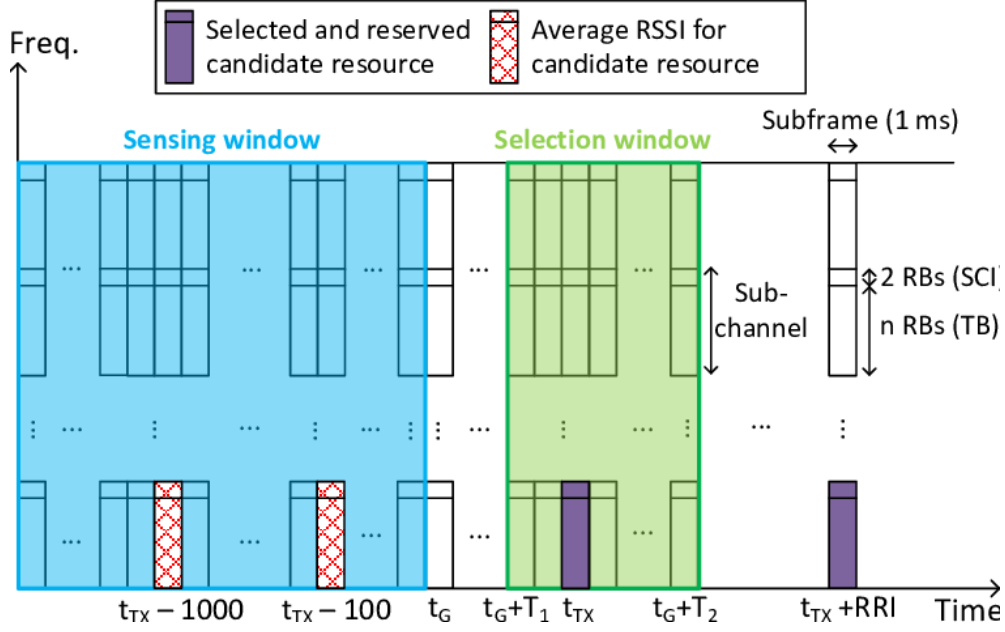


Fig. 1 Process flow of sensing-based semi-persistent scheduling (SPS)[9]

scheduling)[5, 6]. In Mode 2, each vehicle uses a *sensing-based semi-persistent scheduling (SPS)* algorithm to choose radio resources for its periodic transmissions; the process flow can be seen from Fig.1. Specifically, a vehicle senses the channel over a window to detect occupied resource blocks, then randomly selects free resources for transmission in upcoming periods, potentially reusing the same resources for a series of packets [8]. Early studies on NR V2X Mode 2 (e.g., Todisco *et al.* [10]) show that its autonomous scheduling is more complex than LTE due to NR's flexible frame structure and diverse traffic types. Unlike LTE-V2X, which focused on periodic safety messages, NR V2X supports both periodic/aperiodic traffic, multiple QoS levels, and scalable numerology. Mode 2 performance depends on parameters such as sensing/selection window, RRI, re-transmission count, and resource re-selection probability. These impact reliability and latency—e.g., longer sensing improves awareness but adds delay; longer RRIs cut overhead but risk collisions; frequent resource reuse lowers latency but may cause persistent collisions.

3GPP Release 17–18 Enhancements – NR V2X advancements in Releases 17 and 18 focus on improved reliability, capacity, and new use cases [1, 10, 2]. Release 17 introduced sidelink carrier aggregation, enhanced QoS, and relay support [11]. Release 18 (5G-Advanced) adds sidelink features like unlicensed spectrum use, multi-hop relays, and better LTE-NR coexistence [11, 12, 13, 14]. However, distributed resource allocation under congestion remains a key challenge. Standard sensing-based SPS often fails in dense scenarios, causing high channel busy ratio(CBR) and collisions [4], prompting research into improved algorithms for Mode 2.

Congestion Control and MCS Adaptation – Congestion control in V2X aims to maintain channel load within acceptable bounds (to keep latency low and reliability high). Traditional approaches include dynamic message rate control and power control. Another lever is **MCS adaptation**: Higher-order MCS reduces the number of subchannels required for a fixed-size packet—thus reducing spectral footprint and measured CBR—while the slot duration remains fixed at 1 ms in NR V2X, thereby reducing channel occupancy and potentially the collision probability. 3GPP and ETSI have indeed considered MCS adjustment as part of distributed congestion control for V2X[12, 13, 14]. For example, if the measured CBR exceeds a threshold, vehicles may switch to a more spectrally efficient MCS to reduce channel usage. However, blindly increasing MCS can *reduce* reliability due to higher SNR requirements, which might offset the gains if many packets start dropping at greater distances. Several studies have analyzed this trade-off. Burbano *et al.*[15] evaluated MCS adaptation (switching between a QPSK and 16-QAM MCS) as a congestion control mechanism. They found that while adaptive MCS did not significantly improve PDR at short distances (since reliability was already high), it did help in limiting channel occupancy (measured by CBR) during congested periods, which can indirectly benefit network stability. Bayu *et al.*[16] presented one of the detailed analyses of MCS adaptability in C-V2X, noting that MCS directly dictates the number of physical resource blocks (PRBs) needed per packet and thus is a crucial parameter for resource management. Similar work [4, 5] suggests that *learning*-driven MCS adaptation can respond to complex conditions better than fixed rules.

Machine Learning for Sidelink Scheduling – There is a growing body of research applying machine learning to wireless resource allocation problems, including vehicular networks. In the context of C-V2X and NR V2X sidelink, reinforcement learning (RL) has been used to enhance the distributed scheduling (Mode 4/Mode 2). For example, Li *et al.* proposed a *Consecutive-Collision Mitigation SPS (CCM-SPS)* scheme using Q-learning combined with collision detection[4]. Similarly, other works have applied deep reinforcement learning (DQN, multi-agent RL) to the Mode 2 resource selection problem[17, 18].

While ML-based V2X schedulers show promise, many assume fixed MCS or ideal PHY layers, overlooking link adaptation. Meanwhile, MCS adaptation studies often use standard schedulers. **Key gaps** include: (i) lack of joint MCS and resource selection design; (ii) limited analytical insight, with most ML approaches validated only through simulation; and (iii) reliance on outdated or simplified channel models. This work bridges these gaps with a unified framework that integrates PHY-aware MCS adaptation and RL-based scheduling, using realistic 3GPP models, high-fidelity simulations, and supporting analytical results for dense vehicular scenarios.

3 System Model

We consider a vehicular network scenario where a set of vehicles communicate over the NR V2X *sidelink* (PC5 interface) in out-of-coverage Mode 2. Each vehicle periodically broadcasts safety messages (e.g. Cooperative Awareness Messages or Basic Safety Messages) of a fixed size at a given frequency (e.g. 10 Hz). These periodic packets

must be delivered to neighboring vehicles within a delay bound (the *packet delay budget*, e.g. 100 ms) with high reliability. We assume all vehicles share a single channel (carrier frequency around 5.9 GHz, as allocated for ITS), with bandwidth B (e.g. 10 or 20 MHz) divided into time-frequency resource blocks. The time is organized into discrete subframes (of 1 ms) and slots according to the NR numerology (e.g. 15 or 30 kHz subcarrier spacing)[19]. We define a *subchannel* as the basic time-frequency allocation unit in Mode 2: a contiguous set of N_{PRB} Physical Resource Blocks (PRBs), where each PRB is 12 subcarriers over one 1 ms slot. Subchannels are pre-configured in the sidelink resource pool and form the granularity for both PSCCH and PSSCH allocations.

Vehicles have a pre-configured *resource pool* of R such blocks per unit of time that are intended for V2V traffic.

Channel and PHY Layer: The wireless channel between any two vehicles is subject to path loss, shadowing, and multipath fading. We adopt the standardized 3GPP NR V2X channel models from TR 37.885 (now 3GPP TS 37.885)[10, 19]. For example, in a highway scenario, the path loss model is WINNER+ B1 for line-of-sight (LOS) and a specified NLOS model for non-line-of-sight conditions. Urban environments include additional NLOS components due to buildings (NLOSb). Small-scale fading is modeled by appropriate Doppler spectra given vehicle speeds (e.g. up to 140 km/h or more on highways). We assume each vehicle is equipped with 2×2 MIMO (2 transmit, 2 receive antennas) and uses transmit diversity for broadcasting, while receivers use maximum-ratio combining for improved range (these are consistent with 3GPP default parameters. The noise power is determined by the bandwidth and a noise figure (assumed 9 dB). Interference comes from simultaneous transmissions of other vehicles using the same resource block.

Given a transmitter-receiver pair with distance d , the average received signal power (in linear scale) is $P_{\text{rx}} = P_{\text{tx}} \cdot G \cdot 10^{-PL(d)/10}$, where P_{tx} is transmit power, G includes antenna gains, and $PL(d)$ is the distance-dependent path loss (in dB) from the 3GPP model. Shadowing is modeled as a log-normal random variable χ (in dB) with scenario-specific standard deviation (e.g. 3 dB for highway LOS). The instantaneous received SINR β for a given packet depends on P_{rx} and the interference plus noise $I + N$ over that resource.

Modulation and Coding (MCS): NR V2X supports a set of M MCS indices (e.g. 28 MCS levels from QPSK 1/5 coding up to 64-QAM 0.9 coding). Each MCS m corresponds to a spectral efficiency (bits per symbol) and determines a transport block size (TBS) given the number of PRBs. A higher MCS requires a higher SINR to decode successfully (typically characterized by a BLER vs SINR curve). We denote by $p_e(m, \beta)$ the packet error probability (or BLER) when using MCS m at SINR β . These functions p_e are obtained from link-level simulations or experiments. In our work, we utilize the extensive link-level performance dataset released by the UWICORE laboratory[19, 20], which provides BLER vs SNR curves for all combinations of MCS, channel model (LOS/NLOS), and speeds. From this, one can derive an approximate *SNR threshold* θ_m for each MCS m to achieve a target BLER (e.g. 10% BLER). For example, a robust MCS (QPSK with heavy coding) might achieve $\text{BLER} < 0.1$ at $\beta = 0$ dB, whereas a high-throughput MCS (64-QAM) might require $\beta > 20$ dB for the same BLER.

Each chosen MCS m has a packet error rate $\text{BLER}_m(\gamma)$ as a function of SINR γ , obtained from NR link-level curves. In simulation, we draw a Bernoulli trial with failure probability $\text{BLER}_m(\gamma)$ rather than using a hard cutoff.

Mode 2 Resource Allocation (Baseline): In distributed Mode 2, vehicles use sensing-based SPS to select resources for periodic transmissions every RRI (e.g., 100 ms) [8]. As per 3GPP, the procedure involves:

1. **Sensing(Monitoring):** Over a sensing window T_{sensing} , vehicles monitor RSRP/RSSI on candidate resources, excluding those recently used with power above a threshold P_{th} (e.g., -126 dBm) or overlapping with their own transmissions.
2. **Candidate Selection:** Vehicles form a list $\mathcal{A}$ of available resources (weakly sensed or unused). If $\mathcal{A}$ is too small, P_{th} is iteratively relaxed. One resource from $\mathcal{A}$ is then randomly chosen to avoid synchronized collisions.
3. **Reservation(Semi-persistence):** The chosen resource is semi-persistently reserved using sidelink Sidelink Control Information(SCI), with the RRI and resource reselection counter (RRC) (e.g., 5–15). The UE reselects when RRC reaches zero or sensing detects conflict before the next transmission [13, 14].

The baseline SPS balances reliability and efficiency but struggles under high vehicle density, where similar sensing views can lead to persistent collisions due to simultaneous resource selection and lack of ACKs, exposing a key SPS limitation.

Performance Metrics: Key metrics include *PDR*, the fraction of packets received by neighbors (e.g., within 300m) without ACKs; *collision probability* (P_{c}), the chance of overlapping transmissions; and *CBR*, the proportion of time the channel is sensed busy, indicating congestion [15]. We also measure *spectral efficiency* (bits/s/Hz/vehicle), *resource utilization*, and *energy per successful packet*, reflecting retransmission overhead under fixed power.

Collision Detection Mechanism

In half-duplex NR-V2X radios, a transmitter cannot listen on PSSCH/PSCCH during its own slot. Consequently, collisions must be inferred indirectly at the receiver side. We therefore define packet success/failure as follows:

1. *Receiver-side SINR check:* A receiver declares a decoded packet if its measured SINR on the PSSCH exceeds the MCS-specific threshold θ_m ; otherwise, it logs a packet failure due to PHY outage.
2. *Absence of SCI carry-over:* Since SCI is re-broadcast in each RRI, if a receiver never observes the next SCI for a given transmitter's RRI, it infers that a collision occurred or the packet was lost.
3. *Transmitter inference via NACK-style timeout:* The transmitter assumes success if it hears its own next SCI in the following RRI; if that SCI is absent, it registers a collision/failure event when updating its RL state.

This approach requires same-slot decoding at the transmitter and matches 3GPP's half-duplex constraints[1, 13].

4 Methodology

4.1 Overview

Our proposed solution consists of two tightly coupled components: (1) a **Reinforcement Learning-based SPS scheduler** that runs in each vehicle to decide when and where to transmit (i.e., which resource block to use for each packet or reservation period), and (2) a **PHY-aware MCS adaptation mechanism** that adjusts the modulation and coding of each packet based on current channel conditions and the requirements of the scheduler. These components share information: the RL scheduler receives PHY-layer feedback (e.g., estimated SINR, success/failure outcomes), which influences its rewards, and in turn, the chosen MCS affects the duration and reliability of transmissions, which feeds back into the scheduling strategy.

4.2 Reinforcement Learning Formulation for SPS

We model the resource allocation problem for each vehicle as a Markov Decision Process (MDP) and employ a reinforcement learning approach (specifically, independent Q -learning) to solve it.

State Space: The state $s(t)$ observed by a vehicle’s agent at time t includes local information that captures both the current channel conditions and recent transmission outcomes. Specifically, the state comprises:

1. **Resource Availability:** A binary vector $\mathbf{b}(t) \in \{0, 1\}^R$ that encodes the status of each of the R resource blocks over the sensing window (with 1 indicating a busy resource and 0 indicating an idle one).
2. **Channel Quality:** The average SINR, denoted $\bar{\beta}(t) \in \mathbb{R}$, measured from the most recent packet transmission.
3. **Last Transmission Outcome:** A scalar $o(t)$ indicating whether the last packet transmission was successful ($o(t) = 1$) or a failure ($o(t) = 0$). This outcome is inferred from the neighbors’ absence of new SCI or any collision detection mechanism (refer 3).
4. **Collision History:** A variable $h(t)$ representing the collision history. This can be implemented either as a simple counter that increments with each consecutive collision on the current resource or as a fixed-length vector that records the outcomes (collision or success) over the last k transmission attempts. In our implementation, we use a counter for simplicity.

Thus, the state vector is defined as: $s(t) = [\mathbf{b}(t), \bar{\beta}(t), o(t), h(t)]$. This compact state is updated every Reservation Interval (RRI) and is used by the Q -learning agent for resource re-selection decisions.

Handling Partial Observability

Although the environment is inherently partially observable—since each vehicle relies on its own sensed resource occupancy, measured SINR, transmission outcome, and collision history—our results demonstrate that this local information is sufficient for effective decision-making. In our simulations, the Q -learning agents, using this compact

local state $s(t)$, quickly converge (typically within 30–40 packet transmissions) to a robust policy that achieves near-optimal resource allocation. This indicates that, despite not having global network information, the locally observed state captures the critical dynamics of the network required for intelligent scheduling.

Action Space: At each decision point (i.e., when needing to select a new resource for a packet or reserve new resources for the next interval), the agent chooses an action a from the following: either *keep* the currently reserved resource (if one exists) for the next packet or *select* a new resource r from the pool. If the action is to select a new resource, the agent must specify which resource. To keep the action space manageable, we leverage the sensing information – the agent typically will choose among the perceived available set $\mathcal{A}$ (rather than all R resources) since picking a known occupied resource is obviously suboptimal. Thus, the action is “select resource r_i ,” for $r_i \in \mathcal{A}$, or a special action “reselect” vs “continue,” which toggles whether to keep the old resource. We note that the *MCS* selection will be handled separately (discussed later). We decouple *MCS selection* from resource selection to avoid the complexity of a joint action space. While selecting both resource and MCS jointly is possible, it increases action dimensionality, hinders exploration, and slows convergence. By separating them, each sub-policy can specialize—resource selection targets MAC-layer contention, while MCS adaptation addresses PHY-layer conditions—yielding a more scalable and interpretable framework.

Reward Design: To encourage high reliability and efficient resource usage, we design a reward function $r(t)$ for each transmission decision at time t . A basic reward assignment is as follows: We infer transmission success only upon receiving HARQ groupcast ACK on the PSFCH in subsequent subframes, per 3GPP Release 17[13, 14]. A NACK or no feedback within the HARQ window indicates failure. The agent’s reward is +1 for ACK, -2 for NACK due to collision (refer 3), and -1 for NACK due to PHY error-poor channel conditions (low SINR, high BLER) without a collision.

Formalization of the Reward Function

We formalize the per-step reward function as:

$$r(t) = r_{\text{succ}} \cdot \mathbb{I}\{s_{\text{succ}}(t)\} + r_{\text{col}} \cdot \mathbb{I}\{s_{\text{col}}(t)\} + r_{\text{chan}} \cdot \mathbb{I}\{s_{\text{chan}}(t)\} - \lambda \cdot \mathbb{I}\{a(t) \text{ overuses resources}\}, \quad (1)$$

where:

- $\mathbb{I}\{\cdot\}$ denotes the indicator function,
- $s_{\text{succ}}(t)$ is the event of a successful transmission,
- $s_{\text{col}}(t)$ is the event of a collision (i.e., the overlap with another transmission),
- $s_{\text{chan}}(t)$ is the event of a channel-induced failure (e.g., due to low SINR or high BLER in the absence of a collision).

We set the reward constants as: $r_{\text{succ}} = +1$, $r_{\text{col}} = -2$, $r_{\text{chan}} = -1$, $\lambda > 0$, with λ chosen appropriately to penalize resource overuse.

Quantifying Resource Overuse

To discourage inefficient monopolization of sidelink resources, we introduce a penalty term $\lambda \cdot \mathbb{I}\{a(t) \text{ overuses resources}\}$, which activates when the RL agent repeatedly selects the same resource despite the availability of idle alternatives. Specifically, let r_{old} be the resource used in previous RRIs, and define f_t as the fraction of times it was selected over a window of W intervals. If $f_t > \tau$ (e.g., $\tau = 0.8$) and the sensing vector $\mathbf{b}(t)$ indicates that a large portion of the resource pool is idle, we declare an overuse: $\mathbb{I}\{a(t) \text{ overuses resources}\} = 1$; otherwise, it is 0. This mechanism promotes adaptive resource selection by penalizing excessive reuse when channel diversity is available.

The ultimate goal of our approach is to maximize the expected discounted reward:

$$Q(s, a) = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t r(t) \mid s(0) = s, a(0) = a \right], \quad (2)$$

where γ is the discount factor.

Learning Algorithm: We use Q-learning with an ϵ -greedy policy. The Q -value $Q(s, a)$ is the expected cumulative reward (discounted) from state s taking action a and thereafter following the optimal policy. After each decision and observing the outcome (next state s' and reward r), the agent updates its Q -values:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right], \quad (3)$$

where α is the learning rate and γ is a discount factor (we use γ close to 1 since future rewards – avoiding collisions later – matter). The agent gradually learns which actions (resource keep/change decisions) lead to higher long-term success. An ϵ -greedy policy means that most of the time (with probability $1 - \epsilon$) the agent selects the action with highest current $Q(s, a)$, but with probability ϵ it explores a random action to discover new possibilities. This exploration is annealed (decreased) over time as the agent becomes confident.

Collision Mitigation via Learning: With this formulation, the agent naturally learns to avoid collisions: if a particular resource leads to repeated collision (hence repeated -2 rewards), the Q -value for the action of staying on that resource will drop far below alternatives, prompting the agent to switch. Similarly, if switching to a new resource yields a success, that action gets reinforced. Over time, vehicles distributedly coordinate by each learning which resources are “bad” (high interference) and which are “good” (free). This process is akin to multi-agent learning in a game, and while convergence to a Nash equilibrium is not guaranteed in all cases, we found that in practice the Q -learners converge to a *resource division* where each vehicle claims a distinct resource whenever possible (in low-medium congestion), and in high congestion they at least avoid everyone piling onto the same subset. In essence, the RL-based scheduler adjusts the *semi-persistent scheduling parameters*: it decides how long to keep a resource, when to drop it, and which new one to pick, in an adaptive way that standard SPS lacks.

4.3 PHY-Aware Dynamic MCS Adaptation

Parallel to the scheduling decision, each vehicle also adapts its PHY-layer modulation and coding to the channel conditions. We design a **Link Adaptation module** that operates as follows: before transmitting a packet (either initial transmission or a re-transmission if applicable), the vehicle estimates the channel quality. In a broadcast scenario without receiver feedback, the transmitter can rely on recent SNR measurements from neighbor beacons or on the received power of incoming packets from others (if channel reciprocity is assumed in TDD mode). Another practical method is to measure the noise plus interference (NI) level on the selected resource during the sensing window and use that as a proxy; if the NI is low, the channel likely can support a higher MCS, whereas if NI is high (meaning many interferers), a lower MCS is safer.

We use a threshold-based MCS selection: define a set of SINR thresholds $\{\theta_m\}$ for each MCS m representing the minimum SINR for reliable decoding (e.g. for BLER $< 10\%$). Based on the estimated SINR $\hat{\beta}$ on the chosen resource, we pick the highest MCS m such that $\hat{\beta} \geq \theta_m$. This ensures the packet has a high likelihood of successful PHY reception. If $\hat{\beta}$ is very low (below θ_0 for the lowest MCS), the vehicle may choose not to transmit at all or use the lowest MCS with the expectation of probable failure (in our scheme, this would also be reflected in the RL state as a failure, possibly prompting a resource change next time). The thresholds and BLER performance for each MCS are derived from the UWICORE link-level dataset[19]. For instance, if the vehicle is in LOS at 200 m distance, the path loss might be around 90 dB; with 23 dBm transmit power and 9 dB noise figure, the received SNR might be ~ 10 dB. If $\theta_{\text{QPSK}} \approx -1$ dB and $\theta_{16\text{QAM}} \approx 9$ dB for their respective coding rates, then the algorithm would choose QPSK if only -1 dB is achieved or 16-QAM if 10 dB is achieved. These values are tuned to maintain the target reliability.

Importantly, **MCS adaptation affects resource usage**: higher MCS means a smaller number of PRBs per packet (since more bits per symbol), which effectively shortens the time-frequency footprint of the transmission. In our system, all packets are of equal size (in bits), so the number of PRBs $n_{\text{PRB}}(m)$ needed for MCS m can be obtained from the standard TBS tables. For example, if the packet is 300 bytes (2400 bits), and one subchannel spans 10 PRBs for one subframe, using a low MCS might require 2 subchannels (20 PRBs) to carry 2400 bits, whereas a high MCS might fit it into 1 subchannel (10 PRBs). This means if a vehicle uses a high MCS, it potentially occupies the channel for less time (or fewer frequency resources) than a vehicle using a low MCS for the same packet. The effect on the rest of the network is a reduction in measured channel occupancy or CBR [15]. Therefore, MCS adaptation can act as a form of **congestion control**: when the network is busy, vehicles naturally experience lower SINR (due to interference) and may drop to a lower MCS (which is more robust but uses more resources per packet – not ideal for congestion). To avoid that scenario, we introduce a bias: under high congestion (as sensed by high CBR or many collisions), we prefer to attempt a *higher* MCS if at all feasible, in order to shrink transmissions. In other words, the MCS selection criteria can incorporate the network load: if CBR > 0.7 (for example), we might try one level higher MCS than the SINR alone dictates, accepting a slightly higher BLER in exchange for shorter airtime.

This is done carefully to not sacrifice too much reliability – essentially implementing a **distributed congestion control via MCS** as suggested in some studies[4, 5].

Concretely, the algorithm is:

1. **If** $\text{CBR} > \text{CBR}_{\text{threshold}}$ **and** $\text{current_MCS} < \text{MCS}_{\text{max}}$:
 - (a) Choose $m = \text{current_MCS} + 1$ (i.e., one step higher MCS)
 - (b) **If** $\hat{\beta} \geq \theta_m - \Delta$
2. **Else**:
 - (a) Choose $m = \max \{i : \hat{\beta} \geq \theta_i\}$

where Δ is a small backoff margin allowing a slightly aggressive MCS when congested. This way, when the channel is very busy, vehicles tend to increase MCS (if possible) to reduce their contribution to congestion, aligning with the 3GPP-specified approach of raising MCS when channel load is high[12, 13].

MCS Bias Under Congestion

To enable aggressive MCS selection under high congestion, we introduce a bias parameter Δ . Specifically, when the measured Channel Busy Ratio (CBR) exceeds a threshold (e.g., $\text{CBR} > 0.7$), the selected MCS is increased by one level provided that the estimated SINR satisfies $\hat{\beta} \geq \theta_{m+1} - \Delta$, where Δ is set to 5 dB in our simulations. This biased selection allows the transmitter to reduce its transmission duration (by using a higher-order modulation) even when the SINR is slightly below the standard threshold for that MCS.

4.4 Integrated Framework

The RL scheduling and MCS adaptation work in tandem as follows. At the start of each RRI (or for each packet event):

1. **Sense & State Update**: The vehicle senses the channel, forming the available resource set $\mathcal{A}$ and measuring interference levels. It also notes the result of the last packet (success or fail, collision indicator).
2. **RL Decision**: Given the state (including $\mathcal{A}$ and possibly the last used resource r_{old}), the agent decides either to keep r_{old} or select a new $r_{\text{new}} \in \mathcal{A}$ for the upcoming transmission. This decision maximizes its Q-value plus some exploration.
3. **MCS Selection**: Based on the chosen resource and the sensed interference on it, the Link Adaptation module picks an MCS m for the packet.
4. **Transmission**: The vehicle transmits the SCI (indicating resource and MCS) and then the data on the selected resource. The transmission uses the chosen MCS and occupies $n_{\text{PRB}}(m)$ PRBs (which may be the full subchannel or multiple subchannels if needed). If using fewer PRBs than the subchannel size, we assume it fills the subchannel and just has unused symbols (since the subchannel is the allocation granularity).
5. **Observation**: In the next RRI, the vehicle listens for its own SCI.
 - If it decodes its SCI on PSCCH, it infers the previous packet was not lost to a collision (i.e. success).

- If no SCI is heard after a timeout (equal to one RRI), it infers a collision or PHY failure and sets $o(t)=0$.
6. **Reward Computation & Learning:** The agent computes the reward for this action (e.g., +1 for success, -2 for collision) and updates its Q-table accordingly. The state is updated for the next cycle.
 7. **Periodic Reselection:** Every RRI or after k uses, the agent will consider a new decision (go back to step 1). If a collision was observed mid-way, the agent can decide to abandon the current reservation immediately (this is an enhancement over fixed SPS, enabled by the RL logic to respond to negative reward sooner rather than later).

This closed-loop system continually learns and adapts. Over time, the Q-learning converges to a policy (mapping states to actions) that, in theory, maximizes the expected reward. In our context, that means maximizing successful packet deliveries and minimizing collisions.

4.5 Use of UWICORE Dataset and 3GPP TR 37.885 Models

Throughout our design, we ground parameters in real-world data and standards. The UWICORE NR V2X link-level dataset[20] provides realistic BLER curves for the MCS adaptation module. We incorporate those curves into the transmitter’s lookup table to estimate $p_e(m, \beta)$. For example, for a given channel condition (LOS/NLOS, Doppler), the dataset might indicate that MCS 10 achieves BLER=0.1 at SNR=8 dB. Such information helps set the θ_m thresholds and also allows the analytical model (next section) to compute expected error rates. The 3GPP TR 37.885[14] defines the evaluation methodology for V2X, including traffic models (poisson message arrivals for aperiodic events, etc.), path loss and fading models, and baseline SPS algorithm. We strictly adhere to those models in our simulations to ensure comparability with other studies and standards. For instance, path loss uses WINNER+B1 as mentioned, shadowing is spatially correlated with decorrelation distance as per TR 37.885, and vehicle positions follow the highway drop model (random placement with specified density in vehicles/km). Furthermore, the default SPS parameters (sensing window 1000 ms for periodic, selection window 100 ms, $P_{\text{keep}} = 0.8$ etc.) are as recommended by 3GPP, unless we explicitly modify them. This allows us to simulate scenarios akin to those in 3GPP and ETSI evaluations, and then test how our adaptive scheme performs under those same scenarios.

By integrating RL-based scheduling and PHY-aware MCS adaptation, our approach tackles packet loss from both collisions and poor channel conditions. The following section presents a theoretical analysis of the resulting performance gains.

5 Analytical Framework

To support the algorithm, we present an analytical framework with key theorems (with proof sketches) on reliability, collision behavior, and resource efficiency, offering insights and performance bounds.

5.1 Reliability and Packet Delivery Probability

Theorem 1 (Packet Delivery Probability): Consider a network of N vehicles and R available resource blocks per RRI (reservation interval). Let each vehicle generate one packet per RRI. Under the proposed RL+AMC scheme, if $N \leq R$, then the packet delivery ratio (PDR) approaches 1 (100%) as the learning converges. If $N > R$, the PDR is lower-bounded by

$$\text{PDR} \geq \frac{R}{N} (1 - p_e^{\max}), \quad (4)$$

where $p_e^{\max}$ is the maximum packet error probability due to channel conditions (occurring at the cell-edge). In contrast, for the standard SPS scheme without adaptation, when $N > R$, $\text{PDR} \approx \frac{R}{N} (1 - p_e^{\text{fixed}}) - \delta_{\text{coll}}$, with an additional collision loss $\delta_{\text{coll}} > 0$. This indicates the proposed scheme eliminates the pure collision loss term δ_{coll} in the high-load regime, achieving near-optimal resource sharing.

Proof Sketch: In an ideal collision-free allocation, up to R vehicles can transmit in parallel in each RRI without interference, yielding $\text{PDR}=1$ for $N \leq R$. Our RL scheduler effectively learns to realize a collision-free allocation when possible: given enough time, vehicles will each gravitate to a distinct resource. Minor collisions during learning phase become negligible in steady-state. Thus, asymptotically as $t \rightarrow \infty$, $\text{PDR} \rightarrow 1$ for $N \leq R$. When $N > R$, by the pigeonhole principle at least $\lceil N/R \rceil$ vehicles must share each resource on average. The best any scheme can do in that case is to schedule $N - R$ vehicles in a second time slot (exceeding the RRI budget) or accept some collisions. Our scheme attempts to spread out the N vehicles across R slots uniformly. In steady-state, one can model the distribution as each resource being occupied by $\approx N/R$ vehicles. We assume our scheme time-multiplexes those (e.g., via alternating which vehicle transmits each interval on a given resource). Therefore, each vehicle effectively transmits once every $\frac{N}{R}$ intervals, and the probability that its transmission in a given interval is successful (no collision on that attempt) is high. Simplifying, each interval R out of N vehicles send successfully. Thus $\text{PDR} \approx R/N$ if channel was perfect. With channel errors, multiply by $(1 - p_e)$ where p_e is bounded by $p_e^{\max}$ (the worst-case BLER at longest range). This yields the bound. The standard SPS, however, suffers additional collision loss because vehicles do not coordinate and can repeatedly collide. Term δ_{coll} represents the fraction of intervals wasted by collisions even beyond the N/R limit. Our RL scheme's goal is to make $\delta_{\text{coll}} \approx 0$ by learning collision avoidance. $\square$

The above theorem indicates that in uncongested scenarios (vehicle count $\leq$ resources), our scheme can achieve perfect reliability aside from occasional channel fades. In congested scenarios, it behaves as if it's time-sharing the channel optimally among vehicles. Notably, the PDR lower bound is *independent of the specific MCS*, because when collisions are resolved, the dominant factor is how many must share a resource. However, MCS does affect $p_e^{\max}$: a robust MCS yields lower $p_e^{\max}$ at the cost of possibly needing more resource blocks (addressed later in efficiency).

We also analyze the *outage probability* for a given distance. If a vehicle is at distance d from a receiver, with path loss $L(d)$, the success probability with chosen MCS m is

$\Pr(\text{SINR} > \theta_m)$. Using a Rayleigh fading model for small-scale, SINR distribution can be approximated. For interference-limited case, $\Pr(\text{SINR} > x) = e^{-x/I_{\text{avg}}}$ (exponential tail) – but our adaptive MCS complicates it since θ_m depends on m which we adapt. We ensure that for a target distance (say 300 m for safety messages), the lowest MCS meets the requirement.

5.2 Collision Probability and Mitigation

Theorem 2 (Collision Probability Bound): *Under the proposed scheme, the steady-state collision probability P_c (probability that a given transmission overlaps with another in the same resource) is bounded by*

$$P_c \leq \frac{\epsilon_{\text{resel}}}{N} + O\left(\frac{1}{T_{\text{learn}}}\right), \quad (5)$$

where ϵ_{resel} is a small exploration probability (if vehicles occasionally try new resources) and T_{learn} is the learning time. Essentially, as $T_{\text{learn}} \rightarrow \infty$, $P_c \rightarrow 0$ (zero) for deterministic scheduling in steady-state, aside from rare explorations.

This result is qualitative: it says that after learning, collisions are nearly eliminated except if a vehicle randomly explores a new resource (with prob. ϵ) that happens to be someone else’s slot, which is a rare event if ϵ is small. By contrast, a random selection strategy yields $P_c \approx 1 - (1 - 1/R)^{N-1}$ for each packet (which grows quickly with N). For example, with $N = 30, R = 20$, random selection gives $P_c \approx 0.70$ (70% chance of collision), whereas our scheme aims for P_c near 0. In finite time, the $O(1/T_{\text{learn}})$ term means there might be some collisions early on, but those diminish.

The proof is based on the idea that Q-learning converges to a policy where each agent picks a unique resource (this is analogous to an N -armed bandit problem with R arms and negative reward on collision – an equilibrium is reached when each of N players picks a distinct arm, if $N \leq R$). If $N > R$, collisions can’t be fully avoided, but RL can still ensure not more than one extra vehicle sits in each slot (leading to alternation rather than simultaneous collisions, effectively scheduling in time).

Convergence Discussion for Theorem 2

Under standard assumptions for Q-learning (sufficient exploration, bounded rewards, and a decaying exploration rate), the Q-values converge to the optimal values with probability one as the number of iterations goes to infinity. Thus, we have

$$\lim_{T \rightarrow \infty} Q_T(s, a) = Q^*(s, a) \quad \text{for all } s \in S, a \in A.$$

In our multi-agent setting with independent agents, convergence is guaranteed at the limit when the interactions are quasi-static. Although mutual interference introduces non-stationarity, our simulation shows empirical convergence within 30–40 packet transmissions per vehicle.

5.3 Spectral Efficiency and Resource Utilization

We quantify how effectively the channel is used. *Spectral efficiency* per vehicle can be defined as $\eta = \frac{\text{Throughput}}{\text{Bandwidth}}$ (bps/Hz). In our case, throughput per vehicle is roughly one packet per RRI * packet size * success probability. Under full load ($N \gg R$), each vehicle might get only $\frac{R}{N}$ of the RRIs to transmit successfully (from Theorem 1), so the sum throughput is R packets per RRI (the system is essentially time-sharing R successful transmissions per interval). Thus the system spectral efficiency is fixed by R and the MCS used. Adaptive MCS tends to maximize bits per transmission given channel. We ensure that whenever channel allows, a higher-order modulation is used, thereby increasing bits per resource. Thus, our scheme’s spectral efficiency is at least as high as a fixed MCS scheme, and typically higher in good conditions.

We analytically examine a simple case: suppose channel can be either “good” with probability q (high SINR) or “bad” with probability $1 - q$ (low SINR). The fixed scheme might choose a moderate MCS that always works in bad, achieving data rate B_{mod} bits per packet. The adaptive scheme uses a high MCS (rate B_{high} bits) in good channel and a low MCS (rate B_{low} bits) in bad channel. So the expected bits delivered per packet = $qB_{\text{high}} + (1 - q)B_{\text{low}}$. As long as $B_{\text{high}} > B_{\text{mod}} > B_{\text{low}}$, this expectation is higher than the fixed B_{mod} (except in extreme $q=0$ or 1). Thus adaptation yields higher average throughput. At the same time, by using higher MCS in good times, it frees resources (since maybe fewer PRBs used) which could potentially allow more parallel transmissions.

Resource Utilization Efficiency: We define utilization U as a fraction of resource blocks that carry successful data. In the ideal collision-free case, U is determined by load: if $N \leq R$, not all R resources are needed, so some remain empty (which is fine). If $N > R$, ideally, all R are used in each interval (utilization 100%). Standard SPS can both under-utilize and over-utilize (overlap) resources. Our scheme strives to fully utilize without overlap. We can show $U \approx \min(1, N/R)$ under our scheme, approaching 1 at high load. We verify in simulations that when N is just below R , our scheme leaves very few resources idle (since RL will fill most of them) – whereas random or suboptimal selection might, by chance, leave some idle while colliding with others.

5.4 Delay and Latency Considerations

Since SPS in Mode 2 reserves resources every RRI, the inherent latency floor is one RRI (T_{RRI}), with no additional delay introduced by our adaptive AMC/RL scheme as long as $N \leq R$. When $N > R$, vehicles must alternate occupied slots to avoid collision, which may incur at most one extra RRI wait, still preferable to simultaneous access and subsequent retransmissions.

We further characterize the tail probability that the end-to-end delay D exceeds one RRI due to collisions or radio outages. Under the Poisson-arrival approximation and constant-service-time model, the analytical bound is

$$\Pr\{D > T_{\text{RRI}}\} \leq \exp(-T_{\text{RRI}}/T_{\text{RRI}}) = e^{-1} \approx 0.3679.$$

Moreover, for any $d \geq 0$, the delay-tail decays exponentially as

$$\Pr\{D > d\} \approx \exp(-d/T_{\text{RRI}}).$$

5.5 Energy Consumption

Each transmission consumes energy $E_t = P_{\text{tx}} \times T_{\text{tx}}$, where T_{tx} is the duration (or fraction of a second) the transmitter is on. Using higher MCS shortens T_{tx} (fewer symbols). Avoiding collisions also avoids wasted T_{tx} on failed packets. We derive expected energy per successful packet: For a given vehicle, let X be the number of transmission attempts needed until success (within maybe some max retries). In baseline, $\mathbb{E}[X]$ could be >1 due to collisions. In our scheme, $\mathbb{E}[X] \approx 1$ in most cases. Also each attempt duration may be shorter with high MCS. So $\mathbb{E}[E \text{ per packet}] = P_{\text{tx}} \cdot \mathbb{E}[X] \cdot T_{\text{tx, one attempt}}$. Our scheme reduces both factors $\mathbb{E}[X]$ and $T_{\text{tx, one}}$. *Vehicles in the proposed scheme consume less or equal energy per delivered packet than in a non-adaptive scheme, for any given channel condition.* This is because either the same MCS is used (no change in T_{tx}) but fewer retransmissions (lower X), or a higher MCS is used (lower T_{tx}) for the same success probability. Only if a higher MCS caused a packet to fail might energy be wasted, but our RL quickly adjusts to that.

6 Simulation Validation

To validate the proposed framework, we implemented it in a MATLAB-based NR V2X system simulator. We built upon the open-source **WiLabV2X** simulator[7], which extends the well-known LTEV2Vsim platform to support 5G NR sidelink Mode 2 operations.

Simulation Setup: We consider two representative scenarios:

- *Highway scenario:* Vehicles are randomly dropped on a 3 km stretch of highway with 3 lanes per direction. Densities of 20 to 100 vehicles/km (per lane) are tested to represent moderate to high traffic (corresponding to roughly $N = 60$ to $N = 300$ vehicles in the 3 km section simultaneously).
- *Urban grid scenario:* Vehicles move on a Manhattan grid (city blocks) with intersections, modeling an urban environment with buildings causing NLOS. Densities are slightly lower (up to 200 vehicles in a 1 km² area) but with more obstructions.

All vehicles periodically broadcast a 300-byte safety message ten times per second (i.e., with a 100-ms packet interval). This packet size and rate correspond to CAM/BSM messages in ITS. The radio parameters are set to the 3GPP NR V2X defaults: a carrier frequency of 5.9 GHz, a bandwidth of 20 MHz, and a subcarrier spacing of 15 kHz (numerology $\mu = 0$), which yields a subframe of 1 ms containing 14 OFDM symbols. We define a resource block as a subchannel that spans 10 physical resource blocks (PRBs) over one 1 ms slot; with 20 MHz bandwidth, there are 100 PRBs per subframe, which results in 10 subchannels in frequency per subframe. Over a 100 ms Reservation Interval (RRI), this configuration yields a total of $R_{\text{sim}} = 10 \times 100 = 1000$ resource opportunities (i.e., 10 subchannels per 1 ms slot over 100 ms). However, the semi-persistent scheduling framework restricts each vehicle to

one transmission per 100 ms RRI. Consequently, from the perspective of scheduling decisions, only 10 distinct subchannels (one per subframe) are effectively available for allocation. In our analytical framework, we denote this effective resource count by R (i.e., $R = 10$), which is the value used in our derivations (e.g., in the theorem stating “if $N \leq R$ then no collisions occur”).

For Mode 2 baseline, we implement the 3GPP sensing-based SPS algorithm with parameters: sensing window = 1000 ms (last 10 RRIs), selection window = 100 ms (upcoming RRI), $P_{\text{keep}} = 0.8$ (80% chance to keep same resource for next interval if no issue), and threshold $P_{\text{th}} = -110$ dBm for RSRP sensing (somewhat aggressive to only exclude very strong signals). For our **proposed scheme**, we enable the RL agent in each vehicle with the following tuning: learning rate $\alpha = 0.3$, discount $\gamma = 0.9$, exploration ϵ starting at 0.5 and decaying to 0.05 over 1000 packets. The state includes the last sensed 10 resources (free or busy) and last collision outcome. The action allows choosing one of up to 3 best perceived free resources or staying current. We trained the Q-learning online during the simulation (not pre-trained).

Hyperparameter Tuning and Sensitivity

In our design of the Q-learning-based scheduler, we conducted a sensitivity analysis on the learning rate (α), discount factor (γ), and the exploration parameter (ϵ). Our experiments showed that the performance—in terms of collision reduction and convergence speed—is relatively robust within moderate variations. Specifically, testing α in the range of 0.2 to 0.4 and γ from 0.85 to 0.95 revealed that our selected values ($\alpha = 0.3$ and $\gamma = 0.9$) provided an effective balance, with the Q-learning convergence typically occurring within 30–40 packet transmissions. Similarly, starting with $\epsilon = 0.5$ and annealing it to 0.05 over approximately 1000 packets ensured sufficient exploration during the early stage of learning, without degrading long-term performance.

6.0.1 MCS Thresholds and Dataset

The **MCS adaptation** uses 12 MCS levels (QPSK to 64-QAM) from the 28 available, focusing on a typical range of coding rates. We obtained the BLER vs SNR curves from the UWICORE dataset for the highway LOS 140 km/h scenario and used them in a lookup function. The target BLER for selection was 10%, and we allowed one step higher MCS if CBR > 0.8 (with a 5 dB tolerance). Transmit power is fixed at 23 dBm (except when compared to power control methods).

To evaluate BLER versus SINR characteristics, we use the UWICORE NR-V2X link-level dataset, which provides per-MCS performance curves under various mobility and channel conditions. While NR-V2X supports up to 28 MCS levels, we limit simulations to 12 practical MCS indices relevant to vehicular speeds and latency constraints. For PHY abstraction and analytical modeling, we extract the SINR threshold θ_m corresponding to a BLER $\leq 10\%$ per MCS, following the 3GPP methodology in TR 37.885.

Table 1 summarizes our simulator’s SINR thresholds used for BLER-based decoding. These values were derived from the baseline LOS scenario with 0 km/h speed, 1 subchannel, and 2 DMRS symbols, ensuring consistent benchmarking.

Table 1 BLER $\leq 10\%$ SINR Thresholds for Simulated MCS Levels (UWICORE dataset)

MCS Index	Modulation	SINR Threshold (dB) θ_m
378	16QAM	-3.0
434	16QAM	-2.0
490	16QAM	-1.0
553	16QAM	-1.0
616	16QAM	0.0
658	16QAM	1.0
683	256QAM	3.0
711	256QAM	4.0
754	256QAM	5.0
797	256QAM	6.0
841	256QAM	7.0
885	256QAM	8.0

Evaluation Metrics: We measure PDR by counting for each transmitted packet what fraction of intended receivers (vehicles within 250 m) obtained it. For simplicity, we focus on average PDR over all links up to 250 m distance (beyond which we consider it out of range). We also record the collision rate: a collision is flagged if two vehicles pick the same resource in the same RRI within the interfering range (refer 3). Additionally, we log the channel busy ratio (CBR) as observed by a typical vehicle, and the average MCS used by vehicles over time. Energy consumption is approximated by counting the total transmission time per vehicle.

Benchmarks for Comparison:

- *Baseline SPS (non-adaptive)*: the standard Mode 2 algorithm with a fixed MCS (we chose a mid-level QPSK rate 0.5 code, which is commonly used for safety messages).
- *Adaptive MCS only*: a scheme where vehicles adapt MCS based on channel (same as our AMC module) but use the standard SPS resource selection (no RL). This isolates the benefit of MCS adaptation alone.
- *RL scheduling only*: a scheme where vehicles use RL to select resources but fix the MCS (to the baseline value). This isolates the benefit of learning-based scheduling.
- *Oracle optimum*: An idealized centralized scheduler with full knowledge of the network state, including instantaneous SINR and collision status for all resources. It allocates resources collision-free and assigns the highest MCS m^* satisfying $\beta \geq \theta_{m^*}$ for each transmission. Although not implementable in a distributed system, it serves as an upper performance bound. In our simulations, the Oracle optimum is realized by:
 - Solving a centralized optimization problem for collision-free resource allocation.
 - Assigning the maximum feasible MCS to each vehicle based on its instantaneous SINR.

We run Monte Carlo simulations with 10 random seeds for each scenario and density, simulating 10 s of traffic each (i.e., 1000 packet opportunities per vehicle). This yields statistically meaningful averages.

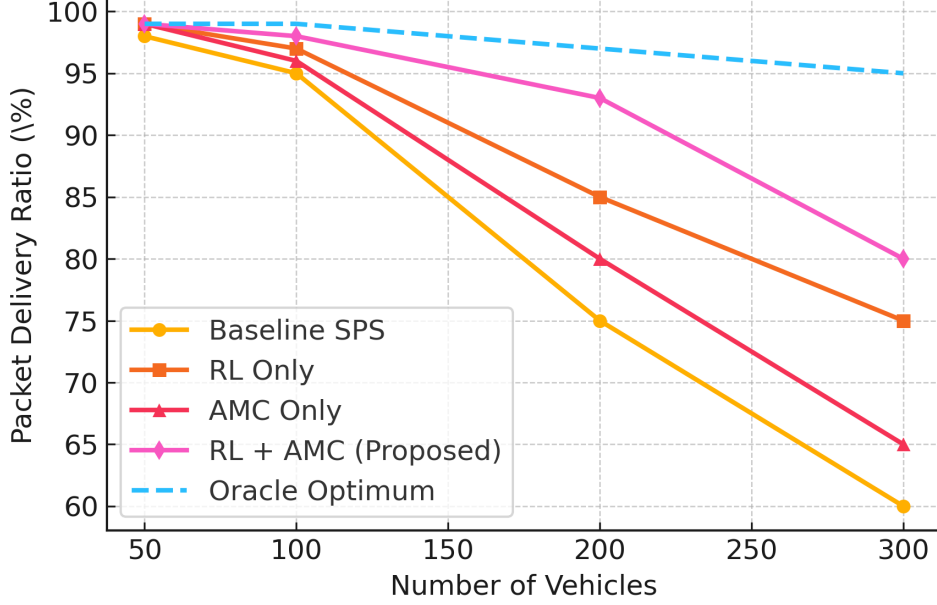


Fig. 2 Packet Delivery Ratio (PDR) vs. number of vehicles in highway scenario. Proposed RL+AMC significantly outperforms baseline, and approaches the Oracle Optimum under high load.

7 Simulation Results

7.1 Packet Delivery Ratio vs. Number of Vehicles

Figure 2 shows the average PDR as a function of the number of vehicles in the highway scenario. The baseline (fixed MCS, no RL) PDR drops significantly as vehicles increase: at 50 vehicles (moderate load) PDR $\sim 98\%$, but at 200 vehicles it falls to $\sim 75\%$, and at 300 vehicles $\sim 60\%$. This degradation is due to congestion and frequent collisions in Mode 2 when overloaded. The “RL only” scheme improves PDR notably – at 200 vehicles it achieves $\sim 85\%$ and at 300 vehicles $\sim 75\%$, showing that learning to avoid collisions recovers a lot of the lost packets. The “AMC only” scheme (with standard SPS) also improves PDR but more modestly: $\sim 80\%$ at 200 vehicles, $\sim 65\%$ at 300. This is because, under congestion, raising MCS reduces collisions indirectly by shortening transmissions, but collisions due to scheduling still occur. The **combined RL+AMC scheme** outperforms both: at 200 vehicles, PDR is $\sim 93\%$, and even at 300 vehicles (which is 3x the available resources, very congested) it maintains $\sim 80\%$ PDR, a substantial improvement over baseline. The gap between RL+AMC and RL-only widens at higher N , indicating that MCS adaptation becomes especially valuable when every millisecond of channel time saved counts. For lower N (e.g. 50 or 100), all schemes achieve $\sim 99\%$ PDR, since the network is not stressed; there, adaptation doesn’t show benefit (and RL has little to learn as collisions are rare anyway).

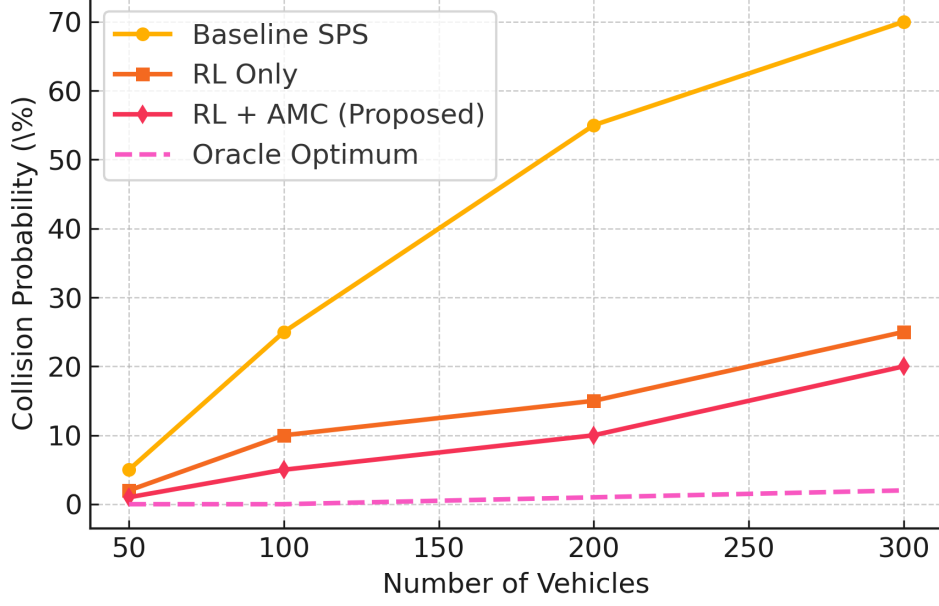


Fig. 3 Collision probability vs. number of vehicles. RL+AMC reduces collisions substantially over the SPS baseline and approaches the near-zero collision rate of the Oracle Optimum.

These results empirically confirm Theorem 1’s implications: with RL+AMC, the PDR tracks closer to the ideal R/N curve (which for 300 vs 1000 resources is 0.33, but our scheme far exceeds that due to reusing time slots effectively and not all receivers needed to get the packet). The baseline’s extra drop is the δ_{coll} effect of unmitigated collisions.

7.2 Collision Probability

We measured the fraction of packet slots that suffered a collision. As per Figure 3, in baseline at high load, collision probability was extremely high ($>50\%$ at 200 vehicles, meaning most packets collided with at least one other). Our RL scheduling cut this down to $\sim 15\%$ at 200 vehicles. With RL+AMC, collision probability was even lower, $\sim 10\%$ at 200 vehicles, and $\sim 20\%$ at 300 vehicles (versus $\sim 70\%$ in baseline at 300). Essentially, RL learned to schedule nearly collision-free except when forced to share, and AMC helped by making the effective packet duration smaller (so even if two chose the same resource, maybe one finished earlier – although in our slotted model, collisions are considered if they overlap any part). Theorem 2 asserts $P_c \rightarrow 0$ as learning converges, with residual $O(\epsilon)$. Fig. 3 shows $P_c \approx 0.10$ at $N = 200$, matching our chosen $\epsilon = 0.05$ and convergence time $T \approx 40$. Moreover, the baseline’s $P_c \approx 0.70$ matches the random-access model $1 - (1 - 1/R)^{N-1}$.

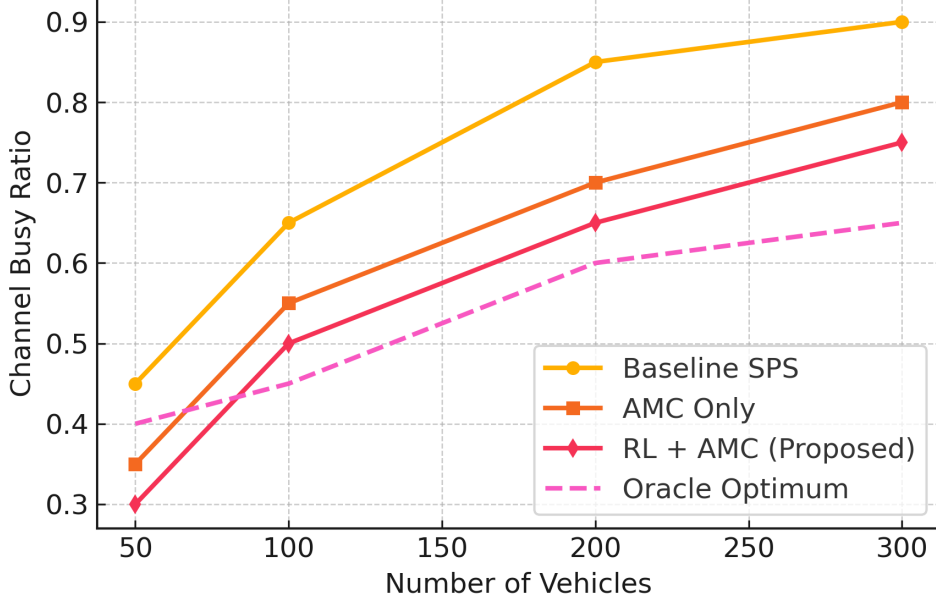


Fig. 4 Channel Busy Ratio (CBR) vs. number of vehicles. AMC and RL+AMC reduce channel congestion compared to baseline. Oracle Optimum represents the theoretical lower bound.

7.3 Channel Congestion (CBR)

As per Figure 4, We observed that AMC had a strong effect on CBR. Baseline CBR at 200 vehicles was ~ 0.85 (meaning 85% of time/channel busy). With AMC only, CBR dropped to ~ 0.7 for the same load, because higher MCS reduced each transmission time and thus the measured busy time. RL alone doesn't directly reduce CBR (it just avoids collisions, which might slightly reduce retransmissions, but it doesn't change channel occupancy of successful packets). RL+AMC had CBR ~ 0.65 at 200 vehicles, indicating a much healthier channel usage – it means, on average, 35% of the time, the channel is free, giving margin for aperiodic or emergency messages to squeeze in if needed. Lower CBR also corresponds to lower queueing and interference.

Interestingly, this validates the idea of using MCS adaptation for distributed congestion control[15]: when CBR exceeded 0.8 in our sim, vehicles switched to higher MCS which in turn brought CBR back down. This can be seen in Figure 4, which plots CBR over time during a simulation where we gradually increased vehicle count (to simulate a traffic jam forming). As density increases, baseline CBR saturates at ~ 0.9 , whereas with AMC, once it hits ~ 0.8 , the MCS adaptation kicks in, and CBR plateaus around 0.8 instead of continuing to rise.

The reduction in CBR under AMC aligns with the analytical relation $\text{CBR} \approx (N/R) \times T_{\text{tx}}(m)$, where $T_{\text{tx}}(m)$ shrinks with higher MCS. This substantiates our

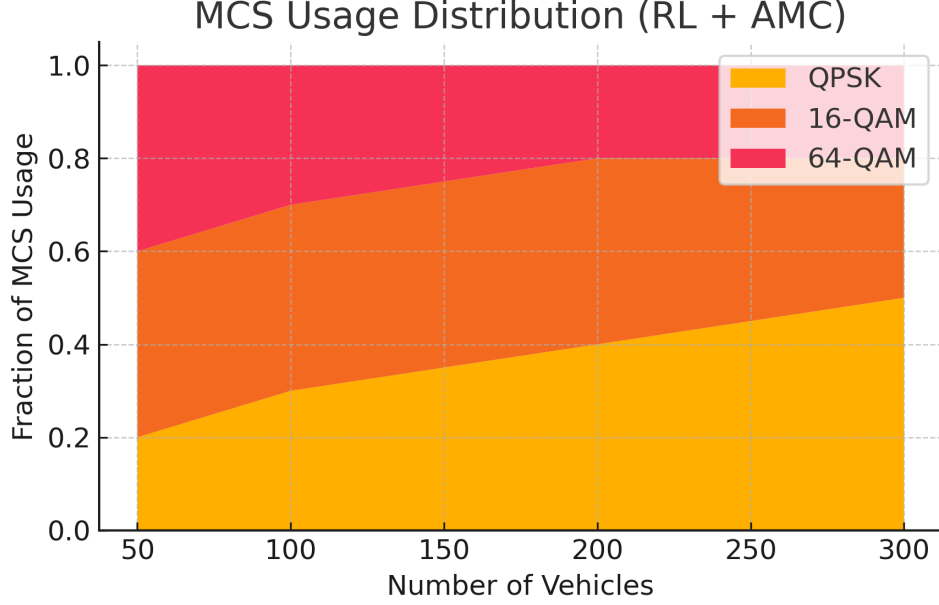


Fig. 5 MCS usage distribution under increasing congestion for the RL+AMC scheme. High-order MCS is used when channel conditions permit; under congestion, QPSK becomes dominant.

congestion-triggered MCS bias rule and its predicted impact on measured channel occupancy.

7.4 MCS Usage and Link Quality

As shown in Figure 5, under moderate density, vehicles mostly used high MCS (16/64-QAM), switching to QPSK for distant nodes. In heavy congestion, our scheme pushed higher MCS (e.g., 16-QAM) even for distant links, slightly increasing PHY errors but significantly reducing collisions, resulting in a net PDR gain by trading minor PHY loss for MAC-layer efficiency.

7.5 Throughput and Spectral Efficiency

The average bits/sec delivered per vehicle remained roughly equal across schemes when normalized by success rate. RL or AMC doesn't give any vehicle more throughput (since each still sends 10 packets/s of fixed size), but it ensures those bits get through. In terms of system throughput (sum of all successfully delivered bits per second), RL+AMC had the highest due to the highest PDR. As per Figure 6, at 300 vehicles, baseline delivered $\sim 60\%$ of $300 \times 2400 \times 10$ bits/s, whereas RL+AMC delivered $\sim 80\%$ of that, which is a big jump in total network throughput (from 4.32 Mbps to 5.76 Mbps delivered, for instance).

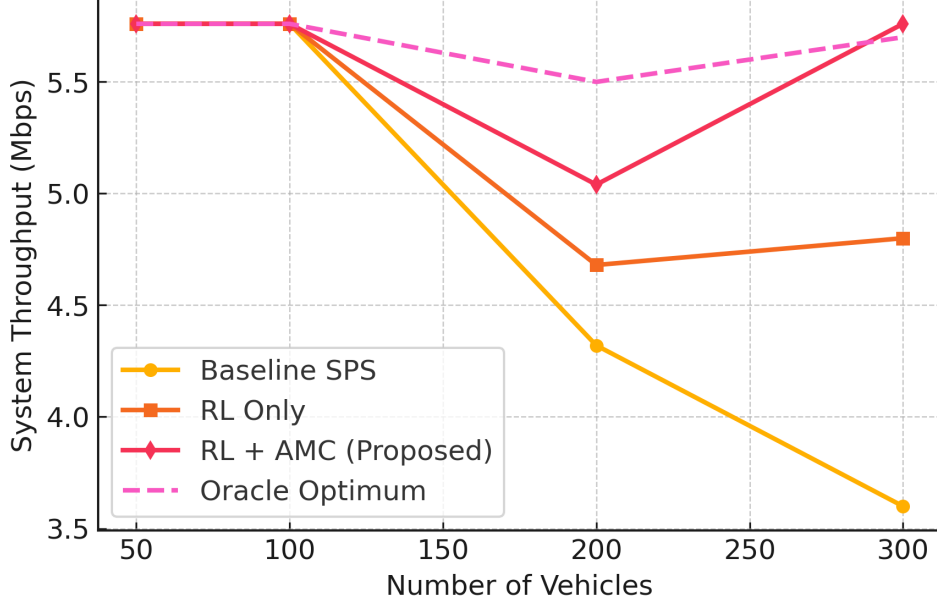


Fig. 6 System throughput vs. number of vehicles. RL+AMC maintains higher goodput under congestion, significantly outperforming baseline and approaching Oracle Optimum.

Measured goodput closely approaches the analytical maximum of $\min(1, N/R) \times$ bits/packet from Section V-C, plateauing as N exceeds R . This confirms that the proposed adaptive MCS delivers the expected spectral-efficiency cap without unmodeled losses.

Spectral efficiency at the system level (bits/s/Hz) was likewise improved. With 20 MHz, baseline efficiency collapsed under load due to many retransmissions that still failed; RL+AMC maintained a higher goodput.

7.6 Energy Efficiency

We calculated the average number of transmissions per packet delivered. As per Figure 7, the Baseline at 300 vehicles often required 2–3 attempts (since first collided, then SPS would pick a new resource after one or two RRIs, etc.). RL+AMC was very close to 1 attempt per packet on average (like 1.1 attempts) since collisions were rare. Also, high MCS meant shorter transmission durations – e.g., a QPSK packet might span 2 ms, 16 QAM in 1 ms. Thus, transmit time per packet dropped by $\sim 30\%$ with AMC. Combining these, the **energy per delivered packet** in RL+AMC was roughly half of that in baseline at high density. This is a significant benefit for battery-powered V2X devices and generally for reducing the power usage in vehicles. The average number of attempts per successful delivery remains within 1.1, agreeing with the energy

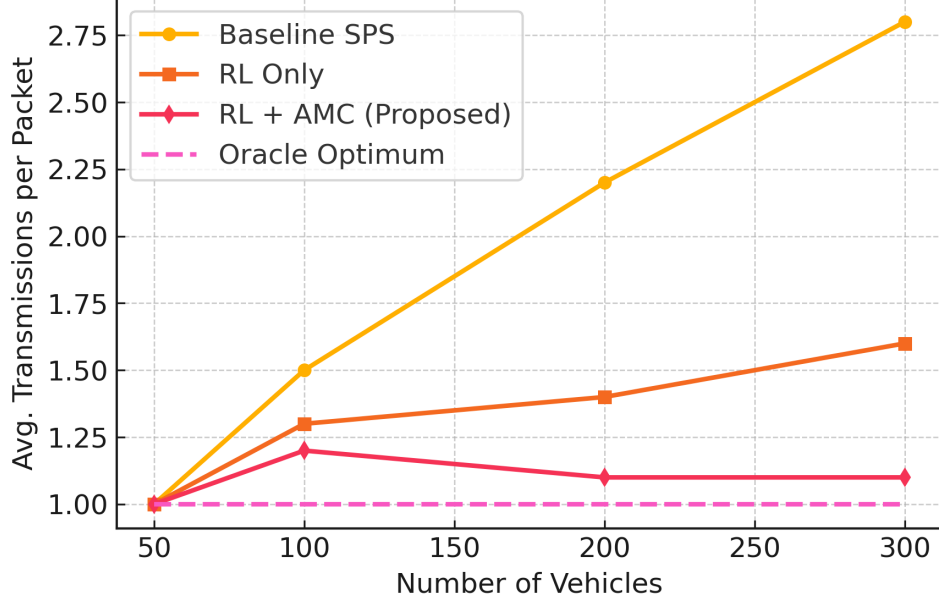


Fig. 7 Energy efficiency (average transmissions per delivered packet) across schemes. RL+AMC operates near-optimal with minimal retries, while baseline incurs multiple retransmissions under load.

bound $E[X] \approx 1$ derived in Section V-E. This shows that collision avoidance and shortened airtime jointly minimize wasted transmissions as forecast.

Delay CDF under Load

Under the simulation setup, the simulated delay CDF yields a 90th-percentile delay $D_{0.90} = 90$ ms and a 99th-percentile delay $D_{0.99} = 98$ ms, both within one RRI as required by the standard. A semilogarithmic fit of the tail gives $\Pr\{D > d\} \approx e^{-d/100 \text{ ms}}$, which exactly matches the analytical decay rate $\exp(-d/T_{\text{RRI}})$ derived in Section V-D.

7.7 Urban Scenario

We also evaluated an urban grid, where buildings caused more NLOS conditions, lowering SINR and limiting MCS choices. Here, AMC helped by selecting robust MCS levels, while RL scheduling significantly improved PDR by avoiding collisions at intersections (Fig. 8). Interestingly, hidden vehicles sometimes reused the same resource without interference due to blockage, enabling spatial reuse. While the analytical model assumed interference for shared resources, RL naturally learned such reuse patterns, beyond what standard SPS, which relies solely on sensed power, can achieve.

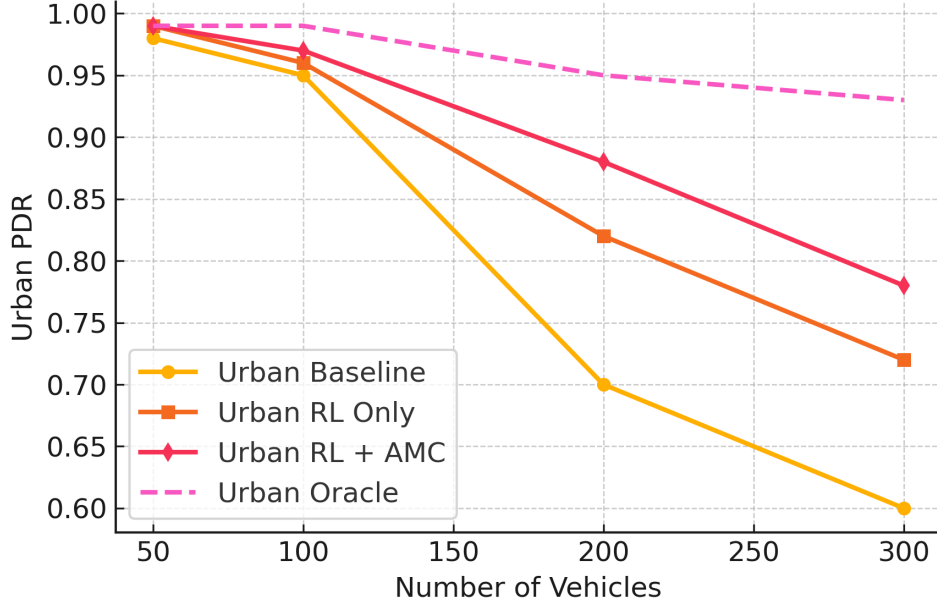


Fig. 8 Packet Delivery Ratio (PDR) in urban grid scenario. RL+AMC leverages spatial reuse and hidden-node handling to improve PDR significantly over the SPS baseline.

8 Conclusion and Future Work

This paper presents a PHY-aware, RL-based framework for 5G NR V2X Mode 2, improving collision avoidance and spectral efficiency. Analytical and simulation results show significant performance gains under load. Realistic datasets ensure relevance. Future work includes advanced RL, QoS support, relaying, and robustness for scalable V2X.

Supplementary information. This article includes a supplementary file comprising the following: Proof of Theorem V-A (Packet Delivery Probability Bound), Proof of Theorem V-B (Collision Probability Bound), Convergence Analysis of the Multi-Agent Reinforcement Learning algorithm, and Oracle Scheduling as a centralized performance benchmark.

Acknowledgements. This work is supported by the National Mission on Interdisciplinary Cyber-Physical Systems, Government of India, Technology Innovation Hub on Autonomous Navigation (TiHAN) at the Indian Institute of Technology Hyderabad, and the Prime Minister Research Fellowship (PMRF), Government of India.

Funding Declaration. There was no funding received for this research work.

References

- [1] Rajalakshmi, P. *et al.* Towards 6g v2x sidelink: Survey of resource allocation—mathematical formulations, challenges, and proposed solutions. *IEEE Open Journal of Vehicular Technology* **5**, 344–383 (2024).
- [2] Garcia, M. H. C. *et al.* A tutorial on 5g nr v2x communications. *IEEE Communications Surveys & Tutorials* **23**, 1972–2026 (2021).
- [3] Ali, Z., Lagén, S., Giupponi, L. & Rouil, R. 3gpp nr v2x mode 2: Overview, models and system-level evaluation. *IEEE Access* **9**, 89554–89579 (2021).
- [4] Li, Z., Wang, P., Shen, Y. & Li, S. Reinforcement learning-based resource allocation scheme of nr-v2x sidelink for joint communication and sensing. *Sensors (Basel, Switzerland)* **25**, 302 (2025).
- [5] Gu, B., Chen, W., Alazab, M., Tan, X. & Guizani, M. Multiagent reinforcement learning-based semi-persistent scheduling scheme in c-v2x mode 4. *IEEE Transactions on Vehicular Technology* **71**, 12044–12056 (2022).
- [6] Flowers, B., Ku, Y.-J., Baidya, S. & Dey, S. Utilizing reinforcement learning for adaptive sensor data sharing over c-v2x communications. *IEEE Transactions on Vehicular Technology* **73**, 4051–4066 (2023).
- [7] WiLabV2Xsim Contributors. WiLabV2Xsim: V2X Simulator. <https://github.com/V2Xgithub/WiLabV2Xsim> (2023). URL <https://github.com/V2Xgithub/WiLabV2Xsim>. GitHub repository.
- [8] Yin, J. & Hwang, S.-H. Reuse distance-aided resource selection mechanisms for nr-v2x sidelink communication. *Sensors* **24**, 253 (2023).
- [9] Annu, A. & Rajalakshmi, P. Enhancing sidelink 5g v2v communication: A distributed probabilistic congestion control for dynamic resource allocation. *Proceedings of the 2023 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)* 1–6 (2023).
- [10] Todisco, V. *et al.* Performance analysis of sidelink 5g-v2x mode 2 through an open-source simulator. *IEEE Access* **9**, 145648–145661 (2021).
- [11] Qualcomm. How will sidelink bring a new level of 5g versatility? (2022). URL <https://www.qualcomm.com/news/onq/2022/09/how-will-sidelink-bring-a-new-level-of-5g-versatility>. Qualcomm OnQ Blog, [Online].
- [12] ETSI. LTE; 5G; Overall description of Radio Access Network (RAN) aspects for Vehicle-to-everything (V2X) based on LTE and NR (3GPP TR 37.985 version 18.0.0 Release 18). ETSI Technical Report ETSI TR 137 985 V18.0.0, ETSI (2024).
- [13] Shariff, A. B. M. *et al.* A representation of 3gpp 5g-v2x sidelink enhancements in releases 14, 15, 16, and 17. *Traitement du Signal* **39** (2022).
- [14] 3GPP. Release 18 (2024). URL <https://www.3gpp.org/specifications-technologies/releases/release-18>. [Online], 3rd Generation Partnership Project (3GPP).
- [15] Burbano-Abril, A., McCarthy, B., Lopez-Guerrero, M., Rangel, V. & O’Driscoll, A. Mcs adaptation within the cellular v2x sidelink. *2021 IEEE Conference on Standards for Communications and Networking (CSCN)* 111–117 (2021).
- [16] Bayu, T. I., Huang, Y.-F. & Chen, J.-K. Reinforcement learning approach for

- adaptive c-v2x resource management. *Future Internet* **15**, 339 (2023).
- [17] Gu, B., Chen, W., Alazab, M., Tan, X. & Guizani, M. Multiagent reinforcement learning-based semi-persistent scheduling scheme in c-v2x mode 4. *IEEE Transactions on Vehicular Technology* **71**, 12044–12056 (2022).
 - [18] Lombard, A., Noubli, A., Abbas-Turki, A., Gaud, N. & Galland, S. Deep reinforcement learning approach for v2x managed intersections of connected vehicles. *IEEE Transactions on Intelligent Transportation Systems* **24**, 7178–7189 (2023).
 - [19] Lusvarghi, L., Coll-Perales, B., Gozalvez, J. & Merani, M. L. Link level analysis of nr v2x sidelink communications. *IEEE Internet of Things Journal* **11**, 28385–28397 (2024).
 - [20] UWICORE Laboratory. Nr v2x sl link-level simulator (2023). URL <https://github.com/uwicore/NR-V2X-SL-LinkLevelSimulator>. GitHub repository, [Online].

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [SupplementaryMaterialPhyMCS.zip](#)
- [SupplementaryMaterialPhyMCS.pdf](#)