Hierarchical Reinforcement Learning-Based Routing Algorithm With Grouped RSU in Urban VANETs

Qin Yang, *Graduate Student Member, IEEE*, and Sang-Jo Yoo, *Member, IEEE*

***Abstract*— The rapid growth of the Internet of Vehicles (IoV) has generated significant interest in routing techniques for vehicular ad hoc networks (VANETs) in both academic and industrial communities. To address the complexity of urban environments and dynamic vehicle mobility, we propose a hierarchical Q-learning-based routing algorithm with grouped roadside unit (RSU) for VANETs. RSUs are grouped, and a Q-vector containing group information is exchanged through vehicle-to-everything (V2X) communications. Q-vector-based road-segment (QVRS) control messages are periodically broadcasted to refresh the V2X evaluation metric, which considers vehicle positions, velocities, directions, and communication conditions. To adapt to the nonstationary vehicular environment, a multi-agent reinforcement learning (RL) algorithm is performed on RSUs at each intersection to achieve distributed learning and local decisions. The hierarchical Q-learning algorithm trains group Q-table and local Q-table individually for reaching destinations on each RSU. The optimal data routing behavior is conducted with two separate Q-tables by utilizing the integrated V2X metric as the reward function. Simulation results demonstrate that our proposed method reduces broadcasting overhead, prolongs path lifetime and maintains a high packet delivery ratio and low average end-to-end delay. The incorporation of group design in our method accelerates the learning process, which facilitates more efficient communication in VANETs.**

***Index Terms*— Data routing, distributed learning, intelligent transportation systems (ITS), reinforcement learning (RL), roadside unit (RSU), vehicular ad hoc networks (VANETs).**

I. Introduction

THE rapid growth of urban areas with large numbers of vehicles on roads has led to significant challenges for transportation systems. Intelligent transportation systems (ITS) have been identified as promising solutions to address these challenges [1]. ITS typically rely on various data sources such as sensors, cameras, and global positioning system (GPS) devices to collect real-time traffic and transportation data, which are processed and analyzed to generate insights and optimize transportation systems. In recent years, the Internet of Things (IoT) has gained significant attention as a means to connect and integrate various devices and systems [2], [3]. One important application of IoT is Internet of Vehicles (IoV), which is a system that enables communication and information exchange between vehicles and other entities such as infrastructures and pedestrians. The IoV has the potential to revolutionize transportation by enabling a range of safer, efficient, and comfort applications, such as collision avoidance, traffic management, and entertainment. Vehicular ad hoc networks (VANETs) are essential components of both IoV and ITS [4]. VANETs are wireless networks that enable vehicles to communicate with each other, roadside units (RSUs), and other infrastructure elements through vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-everything (V2X) communications, thereby facilitating data exchange and improving traffic management and safety [5].

In ITS, routing strategies play a crucial role in efficiently delivering data generated from vehicles, roadside devices, and infrastructure network servers to specific vehicles, vehicle groups, or vehicles on particular roads [6], [7]. Routing is not limited to user data transmission between vehicles; it also extends to various ITS applications. For example, emergency vehicle preemption warnings can be transmitted to vehicles on specific road segments, allowing emergency vehicles like ambulances and fire trucks to receive the right-of-way, thus ensuring faster response times. Infrastructures can provide information on traffic conditions, weather, and road hazards to achieve cooperative awareness through communication with vehicles. Real-time traffic information, road closures, and other variable message signs can be relayed to drivers, assisting them in making informed decisions about routes, detours, and travel times. Greedy perimeter stateless routing (GPSR) [8] and ad hoc on-demand distance vector (AODV) [9] -based routing methods are commonly used in mobile ad hoc networks (MANETs) and VANETs. They both utilize on-demand routing and distance vector algorithms to determine the shortest path to a destination node, and support both unicast and multicast traffic. However, they both suffer from a high network overhead, where GPSR does not scale efficiently in case of larger networks with high node density, and AODV causes high route discovery delay. The RSU-assisted routing approaches overcome these limitations. By utilizing RSUs as intermediaries, the RSU's knowledge and capabilities can be leveraged to make informed routing decisions; thereby reducing the overhead to improve

Manuscript received 12 June 2023; revised 14 November 2023; accepted 8 January 2024. Date of publication 25 January 2024; date of current version 1 August 2024. This work was supported by the Ministry of Science and ICT (MSIT), South Korea, through the Information Technology Research Center (ITRC) Support Program supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP) under Grant IITP-2021-0-02052. The Associate Editor for this article was G. Mao. *(Corresponding author: Sang-Jo Yoo.)*

Qin Yang is with the Department of Electrical and Computer Engineering, Inha University, Incheon 22212, South Korea.

Sang-Jo Yoo is with the Department of Electrical and Computer Engineering, Inha University, Incheon 22212, South Korea (e-mail: sjyoo@inha.ac.kr).

Digital Object Identifier 10.1109/TITS.2024.3353258

the network performance [10]. Artificial intelligence (AI) is currently used in various application domains because of its strong potential to enhance conventional data-driven methods [11]. Decision trees and naïve Bayes have been used to improve the robustness of routing decisions [12], [13]. Particle swarm optimization (PSO) is used for vehicle neighborhood exploration to recognize the identity and the number of neighbors [14]. Ant colony optimization (ACO) was studied to determine the shortest routing path in [15].

Recently, reinforcement learning (RL) has emerged as a promising approach for developing dynamic and adaptive routing solutions that can learn and optimize routing decisions based on environmental conditions and network performance [16], [17]. RL is a machine-learning technique that focuses on learning through trial-and-error interactions with an environment. In the context of VANETs, RL can be used to develop routing methods that have the ability to adapt to changing network conditions and traffic patterns, thereby leading to more efficient and reliable communication [18]. In RL, an agent operates within a set of states $\mathcal{S}$, where each state represents the current situation or condition of the agent and its environment, a set of actions $\mathcal{A}$, where each action is a decision made by the agent in a particular state, and a reward function $\mathcal{R}$, which provides feedback to the agent regarding the quality of its decision. Therefore, an agent, such as a vehicle or RSU, takes actions as next-hop routing nodes based on the current state of the environment to maximize a cumulative reward that reflects the network performance, such as packet delivery ratio, network overhead, and latency.

Considering the characteristics of urban VANETs, RL can not only adapt to a dynamic environment, but also leverage RSUs as relays to enhance the quality of service (QoS) requirements in vehicular networks. This paper presents a hierarchical Q-learning-based routing algorithm with grouped RSU (HQGR) for urban VANETs. The main contributions of this study are as follows:

- We propose a grouped RSU-based routing scheme for large-scale urban areas by dividing them into different groups based on the positions of the RSUs. At each intersection, the group perspective view of the entire topology and the local perspective view within a specific group are integrated to exchange information with neighbors for comprehensive analysis and reduction of communication resources.
- We present a hierarchical RSU-to-RSU (R2R) distributed multi-agent Q-learning algorithm that trains group and local Q-tables separately. The group Q-table represents the expected sum of the rewards (also known as the expected return) for reaching the destination group, whereas the local Q-table captures the destination RSU. This hierarchical approach accelerates the learning process and promotes more efficient data forwarding.
- We develop a Q-vector-based V2X broadcasting mechanism using a Q-vector-based road-segment (QVRS) control message. The mechanism updates the V2X evaluation metric to provide more informed V2V routing decisions, and is utilized as a reward that assists with R2R routing decisions. It achieves the traffic recognition and network overhead control for each road segment.
- We design a novel V2X evaluation metric to predict vehicle positions and increase the awareness of traffic conditions, which is considered as the reward function in Q-learning. To form an integrated V2X path reward, we consider the delivery delay, hop count, link quality, and expected lifetime.

The remainder of this paper is structured as follows: Section II introduces the related work. In Section III, we elaborate our motivation and describe the system models. Section IV presents the details of the proposed RL-based routing algorithm with grouped RSUs for urban VANETs. Simulation results are presented and discussed in Section V. Finally, we conclude this paper in Section VI.

II. Related Work

In this section, we analyze the existing routing strategies in VANETs from various perspectives. Several routing algorithms have been developed to coordinate V2V communications [19], [20], [21]. In particular, the authors in [22] proposed a decentralized vehicle cluster management algorithm to maximize the overall transmission rate. In [23], the authors introduced a distance and signal quality-aware routing method called DSQR to make forwarding decisions based on mid-area node selection. They considered the direction and distance of neighboring nodes, and the link quality. However, it is worth noting that most V2V routing strategies in VANETs do not rely on a central authority or infrastructure for routing decisions. This results in suboptimal routing decisions, increased overhead, and potential conflicts among vehicles owing to a lack of coordinated control and optimization.

RSU-based routing protocols have been proposed to address the challenges associated with intersections. These protocols rely on vehicles or infrastructure elements located at intersections for determining the routing direction by selecting a sequence of intersections [24]. RSU-based routing protocols aim to optimize routing decisions, improve traffic efficiency, reduce congestion, and enhance overall VANET performance. In existing research, several protocols [25], [26] have focused on traffic-aware routing through defined control messages using the RSU. In [27], the authors exploited real-time traffic and network-status measurements. They selected intersections based on the neighbor's predicted position, average received signal strength indicator (RSSI) value, and mobility information recentness. The authors in [28] proposed a scheme for an RSU placement problem that reduced network latency while ensuring good network capacity. However, these methods still have limitations, such as reliance on fixed infrastructure and the need for frequent updates to adapt to changing traffic conditions.

AI technologies have demonstrated significant advantages in VANETs routing. In [29], the authors used the random forest technique to explore the neighborhood of vehicles, along with a support vector machine (SVM) for the classification of nodes based on their ability to forward messages. The use of long short-term memory (LSTM) mentioned in [30]

enhanced the trust level of vehicles in routing decisions. This is because of the ability of LSTM to predict traffic flow stochastically and retain vehicle information for a longer duration. In [31], the authors investigated driver behavior and demonstrated that convolutional neural network (CNN) is a suitable technique for predicting driving routes, thereby mitigating the impact of rapid changes in case of route failures. However, training a fixed model in the real world using these supervised learning techniques in vehicular networks is challenging. Even after training, the dynamic nature of the environment poses difficulties in maintaining model adaptability over time. Hence, RL is considered a preferable approach because it can adapt to nonstationary environments. In addition, the implementation of large AI models in the cloud can result in access delays and hardware resource utilization issues.

In addition to many routing strategies using RL [32], [33], [34], [35], the Q-learning algorithm, which is a value-based temporal difference RL technique, achieves outstanding performance in VANETs through distributed local learning and decision-making. The authors in [36] proposed an offline Q-learning routing approach that utilized real-time GPS data from taxis. This approach used fixed Q-tables that were carried by vehicles during the simulation process. In [37], a link reliability calculation model was developed and considered as a parameter for the Q-learning algorithm and used the bandwidth factor as the learning rate, whereas the authors in [38] focused on evaluating the link reliability model between vehicles and used the ratio of link duration as the learning rate in the Q-learning algorithm based on [39]. The authors in [40] modeled the routing problem as a constrained Markov decision process (MDP) and recast it as an optimization problem based on constrained satisfaction problems that can be learned using an extended Q-learning algorithm.

Several studies have explored the advantages of using a Q-learning algorithm with infrastructure elements to address the routing challenges in vehicular networks. In [41], the authors proposed an optimized Q-greedy geographical forwarding approach based on V2V Q-learning for vehicle selection and R2R Q-learning for RSU selection for traffic-aware routing solutions. The authors in [42] determined the discount factor in the R2R Q-learning algorithm based on vehicle density and distance on the road. They implemented a greedy forwarding strategy on road segments for V2V routing. In this strategy, the vehicle node selects the nearest neighbor to the target intersection or target vehicle as the subsequent relay for packet delivery. Additionally, they reduced communication overhead by designing a congestion control mechanism. Subsequently, the authors in [43] followed the discount factor design on Q-learning for RSU selection, proposed a V2V score for greedy selection among vehicles, and extended this approach in [44]. In [45], the authors proposed a value-based and model-free approach that defined the R2R Q-routing function based on the shortest distance and higher connectivity distribution. The V2V Q-routing function was based on the difference in vehicle speed and vehicle moving direction.

III. System Model

A. Motivation

Several challenges persist in existing RL-based routing methods owing to the specific characteristics of vehicular networks, including vast urban areas and frequent topology changes caused by high vehicle mobility. First, RSUs were introduced as potential solutions to improve the network performance. However, using individual R2R Q-learning routing can be challenging in a large topology because it requires a large number of Q-tables for multi-agent environments and a large dimensional space for each Q-table. This increases the complexity of the Q-learning process, further leading to a slower convergence time. Second, some existing methods used the V2V Q-learning routing approach for road segments, which may not be optimal for vehicular networks because of the high mobility of vehicles. The convergence of V2V Q-tables can be time consuming, and if vehicles move to different road segments, the Q-tables become obsolete, further leading resource wastage. Finally, link quality, path quality, latency, hop-by-hop processing overhead, and path lifetime are all crucial factors for ensuring reliable V2X communications, which are related to mobility and communication conditions.

To address the aforementioned issues, at first we propose a grouped-RSU-based routing scheme for a large-scale topological environment. By grouping the RSUs, the number of Q-tables representing groups and the size of each table can be reduced, thereby improving the convergence time of the Q-learning process. A separate Q-table representing RSUs is learned to enhance the regional features. In addition, group-based RSU allows less information exchange among nearby RSUs, further leading to high efficiency and less network overhead. Secondly, we propose a hierarchical Q-learning architecture for R2R routing. For each RSU, we utilize group Q-learning towards the destination group and local Q-learning within a group towards the destination RSU. Finally, we design an integrated V2X evaluation metric that considers both vehicle mobility and communication conditions. This metric considers the vehicle positions, velocities, movements, directions, and channel status. We present an integrated V2X metric as the reward function in the Q-learning algorithm to make RSUs aware of traffic congestion and network overhead control at each intersection.

B. Group-Based RSU-Assisted Topology Model

Here, we introduce a group-based RSU-assisted topology model as depicted in Fig. 1, in which a large-scale urban road area is depicted. One RSU is deployed at each intersection. Two RSUs that are connected by a road segment are the neighboring RSUs. Each RSU has at least one neighboring RSU based on geography. On each road segment, the vehicles move in two directions at different speeds and movements.

Based on the position of intersections, we divide the area into several groups $\mathbb{G} \triangleq \{G_1, G_2, \cdots, G_g, \cdots, G_{NG}\}$, where $g = 1, 2, \cdots, NG$ and NG is the total number of groups. G_g represents a group that contains multiple RSUs, such as $\boldsymbol{G_g} \triangleq \left\{\boldsymbol{R_1^{G_g}}, \boldsymbol{R_2^{G_g}}, \cdots, \boldsymbol{R_r^{G_g}}, \cdots, \boldsymbol{R_{NR_g}^{G_g}}\right\}$, where NR_g is the

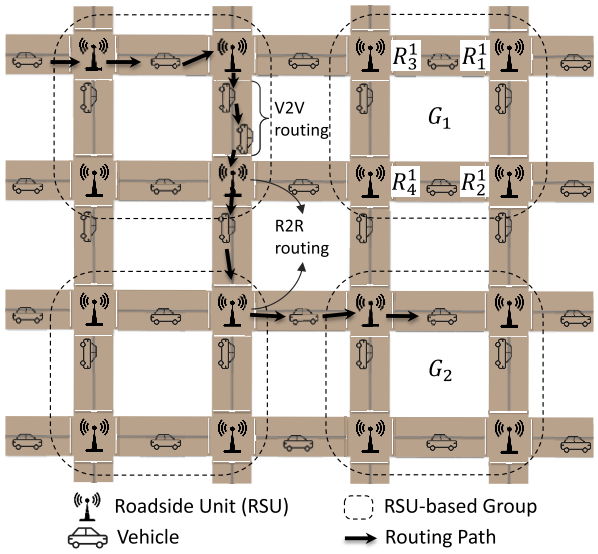

Fig. 1. Network scenario.

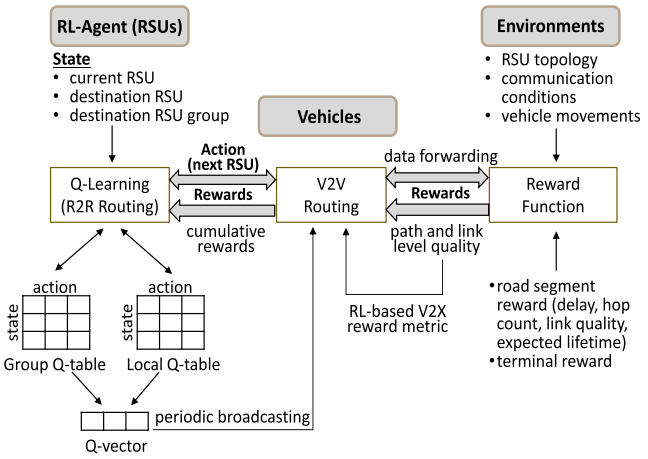

Fig. 2. Proposed system model block diagram.

total number of RSUs in the group G_g. $R_r^{G_g}$ stands for the RSU's id in $\boldsymbol{G_g}$ with index $r = 1, 2, \cdots, NR_g$. Therefore, in the global view, the entire area is subdivided into multiple disjointed RSU groups, whereas in the local view, multiple RSUs are placed in one group. The arrow in Fig. 1 indicates the routing path in which the data packet is delivered through a multi-hop transmission involving a sequence of vehicles and RSUs. In a road network topology, group configuration varies depending on the designer's perspective regarding the considered topologies. The size of a group is also determined based on topology characteristics. Once set, the group size should remain constant during learning. However, for specific learning objectives (e.g., high-traffic hours or late-night scenarios), group composition can be adjusted, even if the topology remains unchanged.

C. Q-Learning Model

RL is a subfield of machine learning techniques that deals with learning how to make decisions by interacting with the environment and receiving feedback in the form of rewards. The Q-learning algorithm is an RL algorithm that uses a variant of the Bellman equation. The Q-learning algorithm is used to determine the optimal policy in an MDP without explicitly knowing the transition probabilities. This indicates that Q-learning can solve problems without any prior knowledge of the environment, which we utilize in our scenario. The Q-learning algorithm uses the Q-value (also known as the Q-function) to measure the expected total reward that an agent can receive by performing a particular action a in a particular state s. Each agent starts with an initial Q-table and then begins to explore the environment. The Q-value in the Q-table is updated as in

$$Q(s,a) = Q(s,a) + \alpha\{r + \gamma \max_{\forall a'} \left(Q\left(s', a'\right)\right) - Q(s,a)\} \quad (1)$$

where $Q(s,a)$ is the Q-value of state-action pair (s,a), $\max_{\forall a'} \left(Q\left(s', a'\right)\right)$ calculates the maximum expected future reward by taking the best possible action a' in the next state s'. α is a learning rate, r is a receiving reward, and γ is a discount factor that determines the importance of future rewards. During the learning process, the Q-value is updated based on the immediate reward received by the agent after performing an action. If the Q-value becomes high for a particular state-action pair, the agent expects to receive a high total reward by taking that action in that state to achieve the optimal policy. In other words, a higher Q-value indicates that the agent has learned which state-action pair is more likely to receive a high reward in the long term. Therefore, the agent should choose the action with the highest Q-value at a given state.

D. Proposed RL-Based Routing System Model

Fig. 2 depicts the block diagram of the proposed RL-based routing system model. First, each RSU acts as an agent and maintains two types of Q-tables based on the different states. The group Q-table stores and updates global Q-values for each destination group and neighboring RSU action pair, whereas the local Q-table updates local Q-values for the destination RSU within the destination group. Subsequently, based on the group and local Q-tables, a Q-vector is generated at each RSU and periodically broadcasted through the vehicles by interacting with the environment. This broadcasting mechanism is used to evaluate the V2X reward metric between two neighboring RSUs by considering the communication conditions and vehicle movements. The V2X metric is designed to measure the quality of V2X path and link performance at both levels within a road segment. We incorporate four key elements: delivery delay, hop count, minimum link quality, and minimum expected lifetime. When broadcasting, vehicles use this metric to measure the performance of all potential links and paths, enabling them to identify the next optimal relay node. Detailed explanations are available in the following section. Finally, for data routing, the next RSU is selected based on its maximum Q-value or random action to balance the exploration and exploitation using a group or local Q-table. The V2X metric is used for selecting the next vehicle to reach the next RSU, and as the reward function to update

the Q-tables of RSU. This process is repeated until the data is successfully delivered to destinations where a terminal reward is provided.

IV. Hierarchical Q-Learning-Based Routing Algorithm With Grouped RSU (HQGR)

This section elaborates on the details of the proposed method, which comprises three parts. The first subsection introduces the proposed HQGR group and local Q-table definitions. The second subsection explains the Q-vector-based V2X broadcasting mechanism, including the design of the integrated V2X evaluation metric. In the third subsection, the proposed group-based hierarchical Q-learning routing mechanism with the V2X metric utilization and Q-table updates is described.

A. HQGR Group and Local Q-Table Definitions

In our scenario, we select the RSUs as the agents for RL. The set of agents is defined as $\mathbb{R} \triangleq \{R_1, R_2, \cdots, R_i, \cdots, R_{NR}\}$, where NR is the total number of RSUs. The environment comprises the entire network topology. The destination vehicle of the data is denoted as v_d and the nearest RSU of v_d is represented as R_d (i.e., destination RSU). The destination group to which the destination RSU, R_d belongs, is denoted as G_d. In this study, we assume that a location service server is utilized such as Amazon Location Service. The location service server can identify where the destination vehicle is located on the road segment and relay this information to the source RSU. Since RSU placement information is shared among all RSUs, the source RSU can determine the destination RSU and the destination group to which the destination RSU belongs. Any location service platforms or algorithms that provide real-time access to all vehicle locations can be applied to our approach to determine the destination RSU. In the proposed Q-learning method, each RSU acts as an agent in the RL and maintains two types of Q-tables: a group Q-table and a local Q-table. A group Q-table is used to determine the next group to deliver data to the destination when the current agent R_i does not belong to the destination group. A local Q-table is used to determine the next-neighbor RSU when agent R_i is within the destination group.

The set of states of agent R_i is defined as in

$$\mathcal{S}_i \triangleq \begin{cases} \{G_1, \cdots, G_g, \cdots, G_{NG}\}, & \text{if } R_i \notin G_d \\ \left\{R_1^{G_d}, \cdots, R_r^{G_d}, \cdots, R_{NR_d}^{G_d}\right\}, & \text{if } R_i \in G_d \end{cases} \tag{2}$$

In the group Q-table, because all groups can be the destination group, $\mathcal{S}_i$ includes all the groups. For the local Q-table, the state is the destination RSU, and the set of states includes all RSUs in G_d for representing all possible destinations. The destination can be adaptable and flexible depending on various real-world applications. It may involve a source transmitting data to a specific vehicle, or the destination can extend to one or more RSUs, or even multiple RSU groups for disseminating information within specific areas. Simultaneously, as data reaches the destination RSUs, this information can be broadcast to all vehicles on the corresponding road segments, thereby facilitating the delivery of emergency warnings and traffic information to promote cooperative awareness.

Group State	Action			
	R_1^i	⋯	R_n^i	⋯
G_1	$Q_i^G(G_1, R_1^i)$	⋯	$Q_i^G(G_1, R_n^i)$	⋯
⋮	⋮	⋮	⋮	⋮
G_g	$Q_i^G(G_g, R_1^i)$	⋯	$Q_i^G(G_g, R_n^i)$	⋯
⋮	⋮	⋮	⋮	⋮

(a)

Local State	Action			
	R_1^i	⋯	R_n^i	⋯
$R_1^{G_d}$	$Q_i^L(R_1^{G_d}, R_1^i)$	⋯	$Q_i^L(R_1^{G_d}, R_n^i)$	⋯
⋮	⋮	⋮	⋮	⋮
$R_r^{G_d}$	$Q_i^L(R_r^{G_d}, R_1^i)$	⋯	$Q_i^L(R_r^{G_d}, R_n^i)$	⋯
⋮	⋮	⋮	⋮	⋮

(b)

R_i 's Q-Vector	
Group	Value
G_1	$GQ_i(G_1, *)$
⋮	⋮
G_g	$GQ_i(G_g, *)$
⋮	⋮

(c)

Fig. 3. Forms of (a) group Q-table, (b) local Q-table, and (c) Q-vector.

The actions that RSU R_i can take for selecting the next road-segment RSUs in a routing path to reach the destination group or destination RSU, are defined as $\mathcal{A}_i \triangleq \{R_1^i, \cdots, R_n^i, \cdots, R_{NNR^i}^i\}$, which represent the neighboring RSUs of R_i from the current RSU intersection to the neighboring RSU intersections. NNR^i is the number of neighboring RSUs of R_i. R_n^i represents the $n-$th neighboring RSU of R_i.

In the Q-learning algorithm, the Q-table is essentially a matrix that stores the estimated expected return for each action that an agent can perform in each state of the environment. Fig. 3(a) and Fig. 3(b) depict a group Q-table and a local Q-table of RSU R_i, respectively. The rows of the table represent the states corresponding to all possible destination groups or RSUs, and the columns represent actions corresponding to the neighboring RSUs. $Q_i^G(G_g, R_n^i)$ represents the group Q-value of R_i when the current state is G_g and the action is R_n^i. $Q_i^L\left(R_r^{G_d}, R_n^i\right)$ represents the local Q-value of R_i when the destination RSU is $R_r^{G_d}$ by performing action R_n^i.

We define a Q-vector for each RSU R_i, in which $GQ_i\left(G_g, *\right)$ represents the expected maximum return at R_i to each destination group G_g. The Q-vector is derived from the group Q-table and local Q-table, as shown in (3) and (4). The Q-vector form is depicted in Fig. 3(c).

$$GQ_i\left(G_g, *\right) = \max_{\forall n \in \{1, \cdots, NNR^i\}} Q_i^G(G_g, R_n^i) \quad for\ \forall G_g \neq G_{(i)} \tag{3}$$

$$GQ_i\left(G_{(i)}, *\right) = \operatorname*{avg}_{\forall r \in \{1, \cdots, NR_{(i)}\}} \left\{ \max_{\forall n \in \{1, \cdots, NNR^i\}} Q_i^L\left(R_r^{G_{(i)}}, R_n^i\right) \right\} \tag{4}$$

TABLE I
QVRS MESSAGE STRUCTURE

1st part	Time stamp	Sender RSU address	Sender RSU sequence	Sender RSU Q-vector
2nd part	Vehicle address	Vehicle position	Vehicle velocity	Vehicle direction
3rd part	Delivery delay	Hop count	Minimum link quality	Minimum expected lifetime
4th part	V2X path reward			

where $G_{(i)}$ is the group id of RSU R_i and $NR_{(i)}$ is the number of RSUs of group $G_{(i)}$. $GQ_i\left(G_g, *\right)$ denotes the maximum group Q-value that R_i can achieve at a particular group G_g among all possible actions R_n^i with $\forall\ n \in \left\{1, \cdots, NNR^i\right\}$. $\underset{\forall n \in \{1,\cdots,NNR^i\}}{max} Q_i^L\left(R_r^{G_{(i)}}, R_n^i\right)$ represents the maximum local Q-value that R_i can achieve at a particular RSU among all possible actions R_n^i inside the RSU R_i group $G_{(i)}$, and $GQ_i\left(G_{(i)}, *\right)$ denotes that it takes the average of all maximum local Q-values among all possible RSUs $R_r^{G_{(i)}}$ with $r \in \{1, \cdots, NR_{(i)}\}$ in the group $G_{(i)}$.

B. Q-Vector-Based V2X Broadcasting Mechanism

Broadcasting is necessary within RSUs and vehicles to share information and refresh the link status of the network environment, which in turn helps to ensure that packets are delivered to their intended destinations in a timely and reliable manner. To enable efficient and effective communication in the network, we propose a Q-vector-based broadcasting mechanism, in which each RSU broadcasts a designed QVRS control message to its neighboring RSUs through vehicles. Broadcasting is limited to each road segment independently to decrease the network overhead.

Table I lists the structure of a QVRS message, which comprises four parts. The first part, generated by each RSU, includes a time stamp, the sender RSU's sequence number, address, and Q-vector. The second part represents the vehicle movement information and is included in each vehicle on the road segment (i.e., between the sender RSU and the neighboring RSU). When a vehicle receives a broadcast message on a road segment, it records its identification address, location, speed, and direction. The third part of this process involves path quality evaluation, which initiates at the sender RSU and continues through subsequent vehicles that receive the message. This evaluation includes updating and recording path quality in comparison to the previous relay nodes for the same QVRS message, considering factors like accumulated delivery delay, hop count, minimum link quality, and minimum expected lifetime of a path. The final part is the integrated V2X path reward computed by the vehicle using the quality metric. When any intermediate vehicle receives the QVRS, it replaces the values of the QVRS message except in the first part. Finally, the vehicles rebroadcast the QVRS message until they reach the neighboring RSU, as depicted in Fig. 4. The metrics used to calculate the integrated V2X path reward is explained in subsequent subsections.

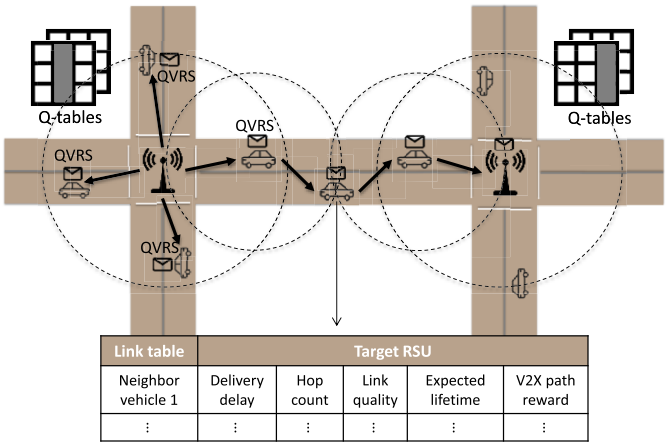

Fig. 4. V2X broadcasting mechanism.

1) Delivery Delay: The delivery delay $DD_{R_i}\left(v_{n\to m}\right)$ refers to the time it takes for a packet to be transmitted from the sender RSU R_i to the receiver vehicle v_m through the previous intermediate node v_n. This is the sum of the delivery delay $DD_{R_i}\left(v_{k\to n}\right)$ from R_i to v_n and the one-hop delivery delay $OD_{v_n}(v_m)$ from v_n to v_m.

$$DD_{R_i}\left(v_{n\to m}\right) = DD_{R_i}\left(v_{k\to n}\right) + OD_{v_n}(v_m) \tag{5}$$

The one-hop delivery delay $OD_{v_n}(v_m)$ represents the time it takes for a packet to traverse a single network hop, which is determined by several factors, including the transmission delay $d_{v_n}^{tx}(v_m)$, propagation delay $d_{v_n}^{pr}(v_m)$, queueing delay $d_{v_n}^{que}(v_m)$ from v_n to v_m, and store-carry and forward delay $d_{v_m}^{SCF}$ of v_n when there are no neighboring nodes within its communication range. These delays are calculated as in

$$OD_{v_n}(v_m) = d_{v_n}^{tx}(v_m) + d_{v_n}^{pr}(v_m) + d_{v_n}^{que}(v_m) + d_{v_m}^{SCF} \tag{6}$$

$$d_{v_n}^{tx}(v_m) = \frac{l_{msg}}{\mu_{nm}},\ d_{v_n}^{pr}(v_m) = \frac{dist_{nm}}{c},\ d_{v_n}^{que}(v_m) = \frac{ql_m}{\mu_{nm}} \tag{7}$$

where l_{msg} is the length of the message from vehicle v_n, μ_{nm} is the transmission rate from vehicle v_n to vehicle v_m, $dist_{nm}$ is the distance between two nodes, c is the propagation speed over the medium, ql_m is the queue length at v_m.

2) Hop Count: Hop count $HC_{R_i}\left(v_{n\to m}\right)$ indicates the number of intermediate nodes, or 'hops' from sender RSU R_i to a current receiver v_m through the previous intermediate node v_n. Each time a packet is forwarded by a vehicle, the hop count increases by one. Generally, a lower hop count is preferred as it implies a more direct and efficient route, with a lower chance of congestion or data loss.

3) Link Quality: We capture the minimum link quality along the path of the QVRS message from R_i to v_m as $minLQ_{R_i}\left(v_{n\to m}\right)$. The minimum value between $minLQ_{R_i}\left(v_{k\to n}\right)$ and the measured link quality between v_n and v_m is determined using (8).

$$minLQ_{R_i}\left(v_{n\to m}\right) = \min\left[minLQ_{R_i}\left(v_{k\to n}\right),\ lq_{v_n}\left(v_m\right)\right] \tag{8}$$

The link quality $lq_{v_n}(v_m)$ measures the reliability and stability of a communication link between v_n and v_m and can be calculated as in

$$lq_{v_n}(v_m) = \frac{P_{v_n}^{rcv}(v_m) - P_{min}^{rcv}}{P_{min}^{rcv}} \tag{9}$$

where P_{min}^{rcv} is the minimum decodable power (dBm) based on the coding rate, channel bandwidth, and other factors. $P_{v_n}^{rcv}(v_m)$ is v_m's received power (dBm) of the message from the previous node v_n.

4) Expected Lifetime: The expected lifetime refers to the estimated path lifetime from the QVRS sender RSU to the vehicle for ad hoc communication. $minELT_{R_i}(v_{n\to m})$ represents the minimum expected lifetime of the path among all links from R_i to v_m through the previous intermediate node v_n as in

$$minELT_{R_i}(v_{n\to m}) = \min\left[minELT_{R_i}(v_{k\to n}),\ elt_{v_n}(v_m)\right] \tag{10}$$

where $elt_{v_n}(v_m)$ represents the expected link lifetime between v_n and v_m, which is determined by the position, velocity, and movement of the vehicle. The distance between nodes v_n and v_m at time t is denoted as $dist_{nm}(t)$. Communication is disconnected when v_m is out of v_n's transmission range d_{tx}. The $dist_{nm}\left(t + elt_{v_n}(v_m)\right) \geq d_{tx}$, $elt_{v_n}(v_m)$ is obtained in

$$dist_{nm}\left(t + elt_{v_n}(v_m)\right)^2 = A \times (elt_{v_n}(v_m))^2 + B \times (elt_{v_n}(v_m)) + C \tag{11}$$

with

$$A = \left(V_{x_m} - V_{x_n}\right)^2 + \left(V_{y_m} - V_{y_n}\right)^2$$
$$B = 2[(x_m - x_n)\left(V_{x_m} - V_{x_n}\right) + (y_m - y_n)\left(V_{y_m} - V_{y_n}\right)]$$
$$C = (x_m - x_n)^2 + (y_m - y_n)^2$$

where V_m, V_n correspond to velocities of v_m and v_n at time t, respectively. V_{x_m}, V_{y_m}, V_{x_n}, V_{y_n} denote the velocity components of v_m and v_n in the X and Y directions, respectively. The communication range d_{tx} is derived from the path loss model with log-normal shadowing as

$$P_{v_n}^{rcv} = P_{tx} + 10log_{10}K - 10\beta log_{10}\left(\frac{d_{tx}}{d_0}\right) + X_\sigma$$
$$\text{with } K = \frac{G_t G_r \lambda^2}{(4\pi)^2} \tag{12}$$

where P_{tx} is the transmission power; β is the path loss exponent; d_0 is reference distance, which is equal to 1 m. G_t and G_r are antenna gains for transmitter and receiver, respectively; λ is the wavelength. X_σ represents a zero-mean Gaussian random variable with a variance of σ^2, often referred to as the shadowing variance.

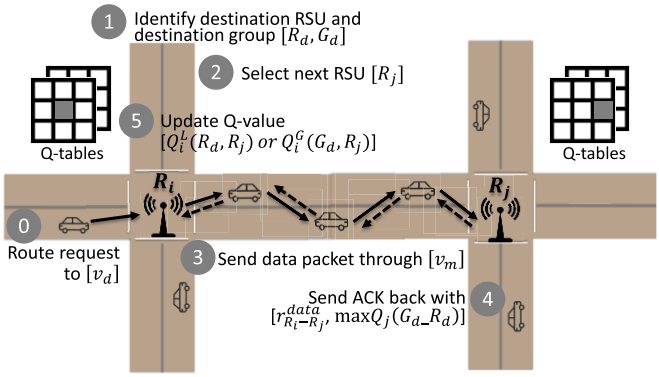

Fig. 5. Data routing process.

5) Integrated V2X Evaluation Metric: The integrated V2X path reward $r_{R_i}(v_{n\to m})$ from RSU R_i to vehicle v_m on the road segment through the previous intermediate node v_n is considered as a comprehensive representation of the four factors, as a longer delivery delay and hop count are undesirable, whereas a better link quality and longer expected lifetime are desirable.

$$r_{R_i}(v_{n\to m}) = -\omega_1 \frac{DD_{R_i}(v_{n\to m})}{\max DD} - \omega_2 \frac{HC_{R_i}(v_{n\to m})}{max\ HC} - \omega_3 \frac{max\ minLQ - minLQ_{R_i}(v_{n\to m})}{max\ minLQ} - \omega_4 \frac{max\ minELT - minELT_{R_i}(v_{n\to m})}{max\ minELT} \tag{13}$$

where the variables $\max DD$, $max\ HC$, $max\ minLQ$, and $max\ minELT$ represent the predetermined maximum values for normalization. The weight factors ω_1, ω_2, ω_3, and ω_4 are used to balance the importance of the four factors in the metric. We maintain the value of $r_{R_i}(v_{n\to m})$ negative.

C. Proposed Group-Based Hierarchical Q-Learning Routing Mechanism

During the broadcasting of QVRS messages, each vehicle updates and stores information regarding its neighboring vehicles in its link table. Using this link table, each vehicle can determine the optimal neighboring node for forwarding data packets to its next-neighbor RSUs. If vehicle v_m receives multiple QVRS messages of R_i from different neighboring vehicles, it selects the neighboring vehicle v_n that generated the highest $r_{R_i}(v_{n\to m})$ value from the sender R_i to the v_m. Eventually, the neighboring RSU, R_j receives multiple QVRS messages of R_i from different neighboring vehicles. $r_{R_i}(v_m \to R_j)$ represents the maximum path reward on the segment from R_i to R_j through R_j's neighbor vehicle v_m. Then, the maximum path reward among all possible routes from R_i to R_j, $r_{R_i-R_j}$ is derived as in (14).

$$r_{R_i-R_j} = \max_{\forall v_m \in V_{ngb}^{R_j}} r_{R_i}(v_m \to R_j) \tag{14}$$

where $V_{ngb}^{R_j}$ is the set of neighboring vehicles of RSU R_j. It should be noted that the path rewards for two directions on the same road segment may differ from each other. When the

Algorithm 1 Q-Vector-Based V2X Broadcasting and Table Updating Mechanism

Input: RSU topology, broadcast interval, vehicle movement
Output: Q-tables, V2X link table
Initialize group & local Q-tables.
while the broadcast interval ends, RSU R_i **do** // RSU broadcasts Q-vector. **do**
  Create its Q-vector based on group & local Q-tables using (3)-(4);
  Broadcast QVRS message;
end while
if a vehicle receives QVRS message **and then**
  it is on the one of the segment of R_i **then** // Vehicle rebroadcasts Q-vector.
  Calculate V2X path reward using $r_{R_i}(v_{n\to m})$ (5)-(13);
  if $r_{R_i}(v_{n\to m})$ is larger than previous received path reward **then**
    Update V2X link table;
    Update vehicle parameters and path reward in QVRS;
    Rebroadcast QVRS message;
  end if
end if
// Neighbor RSU receives Q-vector.
if a neighbor RSU R_j receives QVRS message **then**
  Calculate V2X path reward $r_{R_i}(v_m \to R_j)$ using (5)-(13);
  if $r_{R_i}(v_m \to R_j)$ is larger than previous received path reward **then**
    Update V2X link table;
    Calculate $r_{R_i-R_j}$ using (14);
    Update R_j Q-tables with $r_{R_i-R_j}$ using (15);
    Stop rebroadcast QVRS;
  end if
  end if
end if

QVRS messages arrive at neighboring RSU R_j, the entire row in R_j's group Q-table (i.e., all RSU groups) corresponding to action R_i is updated using

$$Q_j^G(G_g, R_i) = (1-\alpha_G) \times Q_j^G\left(G_g, R_i\right) + \alpha_G\left\{r_{R_i-R_j} + \gamma G Q_i\left(G_g, *\right)\right\} \ \forall G_g \quad (15)$$

where α_G represents the learning rates for updating global Q-tables and $GQ_i\left(G_g, *\right)$ denotes the R_i's Q-vector for group G_g. The Q-vector-based V2X broadcasting and Q-table updating mechanism algorithm is presented in Algorithm 1.

When a vehicle generates a data packet to send, it must first determine the source RSU, denoted as R_s, and then deliver the packet to one of its neighboring vehicles to R_s based on its link table. Subsequently, the source RSU identifies the nearest RSU to the destination vehicle as the destination RSU, R_d, and checks to which group R_d belongs to, i.e., G_d. Any RSU, R_i between R_s and R_d must determine the next RSU to forward the data packet. If R_i is outside of the destination group G_d, then the next RSU R_j can be determined using its group Q-table; If R_i is within the destination group G_d, then it determines the next RSU R_j using its local Q-table, as in (16).

$$R_j = \begin{cases} \underset{\forall R_n^i \in R_{nbg}^i}{\text{argmax}}\ Q_i^G(G_d, R_n^i), & \text{if } R_i \notin G_d \\ \underset{\forall R_n^i \in R_{nbg}^i}{\text{argmax}}\ Q_i^L(R_d, R_n^i), & \text{if } R_i \in G_d \end{cases} \quad (16)$$

where R_{nbg}^i is the set of neighboring RSUs of R_i.

TABLE II
PROPOSED STRUCTURES OF DATA PACKET, AND ACK PACKET

Data Packet Structure			
Data header	Vehicle position	Vehicle velocity	Vehicle direction
Delivery delay	Hop count	Minimum link quality	Minimum expected lifetime
Data			
ACK Packet Structure			
Data path reward ($r_{R_i-R_j}^{data}$)		Maximum Q-value ($maxQ_j(G_d\_R_d)$)	
Acknowledgment for data			

To forward the data packet to the next RSU, the intermediate vehicles on the road segment determine the next vehicle for V2V routing. This decision is based on the link table that contains the next vehicle address for each direction on the segment. The data packet piggybacks and uses the same control fields as the QVRS message to calculate the reward. Upon delivery to the next RSU, R_j, the segment path reward for the data packet, denoted as $r_{R_i-R_j}^{data}$, is computed. It should be noted that the computation of the data packet reward differs from that of the QVRS reward $r_{R_i-R_j}$, whereas the QVRS reward is determined by broadcasting QVRS messages and selecting the largest value among the received messages, the data packet reward $r_{R_i-R_j}^{data}$, is solely based on the path traveled by the data packet along the segment. As depicted in Fig. 5, the subsequent RSU R_j sends an acknowledgment (ACK) packet back to the previous RSU, R_i. The ACK packet includes the computed reward $r_{R_i-R_j}^{data}$ and the maximum Q-value, denoted as $max\,Q_j\,(G_d\_R_d)$ using R_j's current Q-tables. $max\,Q_j\,(G_d\_R_d)$ is derived as in (17). Table II depicts the proposed data and ACK packet structures. In our simulations, the size of the QVRS message ranges from approximately 100 to 200 bytes when utilizing different groups. The data packet contains 512 bytes of payload and an additional 64 bytes for vehicle movement and V2X metrics. The size of an ACK packet is approximately 16 bytes.

$$max\,Q_j\,(G_d\_R_d) = \begin{cases} \max\limits_{\forall R_n^j \in R_{nbg}^j} \left(Q_j^G(G_d, R_n^j)\right), & \text{if} R_j \notin G_d \\ \max\limits_{\forall R_n^j \in R_{nbg}^j} \left(Q_j^L(R_d, R_n^j)\right), & \text{if } R_j \in G_d \end{cases} \quad (17)$$

Whenever R_i receives an ACK, it applies the reward function $\mathcal{R}$ in (18) for Q-learning.

$$\mathcal{R} \triangleq \begin{cases} \mathcal{R}_T + r_{R_i-R_j}^{data}, & \text{if } R_j = R_d \\ r_{R_i-R_j}^{data}, & \text{if } R_j \neq R_d \end{cases} \quad (18)$$

where $\mathcal{R}_T$ is the terminal reward that is given only when the data packet reaches the destination RSU R_d. Then it updates its group or local Q-tables using $\mathcal{R}$ and $max\,Q_j\,(G_d\_R_d)$.If the groups of R_i, R_j, and R_d are the same (i.e., $G_{(i)} = G_{(j)} = G_d$), then R_i updates its local Q-table, as in (19).

$$Q_i^L(R_d, R_j) = (1-\alpha_L) \times Q_i^L\left(R_d, R_j\right) + \alpha_L\left\{\mathcal{R} + \gamma max\,Q_j\,(G_d\_R_d)\right\} \quad (19)$$

Otherwise (i.e., $G_{(i)} \neq G_{(j)}$ or $G_{(i)} = G_{(j)} \neq G_d$), R_i updates its group Q-table as in (20).

$$Q_i^G(G_d, R_j) = (1-\alpha_G) \times Q_i^G(G_d, R_j) + \alpha_G \left\{\mathcal{R} + \gamma max Q_j(G_d\_R_d)\right\} \quad (20)$$

where α_L represents the learning rate of the local Q-table. We employ different learning rates between the group and local Q-tables update, which allows the agent to be more receptive to updating its Q-values based on new experiences.

Over time, the Q-tables can be continuously updated with additional information regarding the expected returns for each action in each state. As the Q-table becomes more refined and accurate, the RSU gains greater confidence in selecting actions that maximize its expected long-term rewards. To balance exploration and exploitation, an $\varepsilon-$ greedy strategy [46] is employed. An initial value ε_{ini} is set for exploration and decreases exponentially over time until it reaches ε_{end}. Algorithm 2 outlines the proposed HQGR for routing decisions.

Fig. 6 depicts an example scenario for the proposed method. We assumed three groups and eight RSUs in the example topology. At each intersection, a Q-vector was formed based on its current group Q-table and local Q-table using (3) and (4), respectively. As depicted in Fig. 6(a), R_1^1's Q-vector values of G_2 and G_3 captured the maximum Q-values in the group Q-table at G_2, G_3 states, whereas R_1^1's Q-vector value of G_1, where R_1^1 belongs was calculated by summing the maximum Q-values at all states in the local Q-table and then averaging them. The Q-vector was included within the QVRS message and broadcasted to the neighboring RSUs. As depicted in Fig. 6(b), R_1^1 broadcasted a QVRS message through multi-hop vehicles to R_2^1 by recording the V2X metric. The hop count and delivery delay were accumulated through vehicles, but the link quality and expected lifetime obtained the minimum values among all the links in a path. At the same time, each vehicle calculated the path reward using (13) when weight factors were 0.25, maximum normalization values were 10. The path reward was stored in vehicle's link table to make V2X routing decisions. In the QVRS, the path reward was updated during broadcasting until it reached R_2^1. R_2^1 selected the maximum path reward $r_{R_1^1-R_2^1}$ by comparing multiple income QVRS messages using (14). In Fig. 6(c), R_2^1's group Q-table update was displayed based on R_1^1's Q-vector and $r_{R_1^1-R_2^1}$ corresponding to R_1^1 action using (15) when the learning rate was 0.1 and the discount factor was 0.9. In Fig. 6(d), when a data packet arrived at R_1^1 with destination information G_2, it is evident that $Q_1^G\left(G_2, R_2^1\right)$ stored the maximum Q-value at state G_2. The data packet was forwarded by the vehicle based on its link table to R_2^1. If the data packet arrived at R_2^1, R_2^1's maximum Q-value at state G_2 and the data path reward $r_{R_1^1-R_2^1}^{data}$ were included in the ACK to send these back to R_1^1. Finally, R_1^1 updated the Q-value of $Q_1^G\left(G_2, R_2^1\right)$ using (20).

Algorithm 2 Proposed group-based hierarchical Q-learning mechanism

Input: v_s, V_d, Q-tables, V2X link table, hyperparameters
Output: Q-tables, routing path

if v_s has data packet **then**
  Send data packet to source RSU R_s throught neighbor vehicle v_m based on v_s's link table;
end if
if an RSU R_i receives data packet **then**
  if $R_i = R_s$ **then**
    Identify destination RSU R_d based on V_d and destination group G_d based on R_d;
    if R_i is outside of G_d **then**
      Select the next RSU with max $Q_i^G(G_d, *)$ in the group Q-table;
    else if // R_i is inside G_d **then**
      Select the next RSU with max $Q_i^L(R_d, *)$ in the local Q-table;
    end if
    Send data packet to next RSU through v_m based on link table;
  else if $R_i \neq R_s \neq R_d$ **then**
    Calculate data path reward $r_{R_i-R_j}^{data}$ using (5)-(13);
    Send backward ACK packet with $r_{R_i-R_j}^{data}$ and its max Q-value;
    if R_i is outside of G_d **then**
      Select the next RSU with max $Q_i^G(G_d, *)$ in the group Q-table;
    else if // R_i is inside G_d **then**
      Select the next RSU with max $Q_i^L(R_d, *)$ in the local Q-table;
    end if
    Send data packet to next RSU through v_m based on link table;
  else
    // $R_i = R_d$
  end if
  Calculate data path reward $r_{R_i-R_j}^{data}$ using (5)-(13);
  Compute reward with $\mathcal{R} = \mathcal{R}_T + r_{R_i-R_j}^{data}$;
  Send backward ACK packet with $\mathcal{R}$;
  Send data packet to v_m through v_m based on link table;
else if an RSU receives ACK packet **then**
  if current RSU is inside G_d **then then**
    Update local Q-value using (19);
  else if current RSU is outside of G_d **then**
    Update group Q-value using (20);
  end if
end if
if a vehicle receives data packet **then**
  if current vehicle is v_d **then**
    Return
  else
    Calculate V2X data path reward using (5)-(13);
    Update vehicle parameters and path reward in data packet;
    Deliver data packet to next neighbor node based on link table;
  end if
else if a vehicle receives ACK packet **then then**
  Deliver ACK to the next backward neighbor node based on link table;
end if

V. Simulation Results

In this section, we detail the simulations which were conducted using SUMO, OMNeT++, and Veins. SUMO is a road traffic simulator that provides vehicular movements in the real world, OMNeT++ is an extensible network simulator, and the Veins framework provides an interface between SUMO and OMNeT++. We demonstrated extensive simulation results using the parameters listed in Table II. The 900 m × 900 m urban area was covered with 24 road segments. Each road segment was 300 m long and comprised two lanes. To ensure a comprehensive coverage of the network, 16 RSUs were

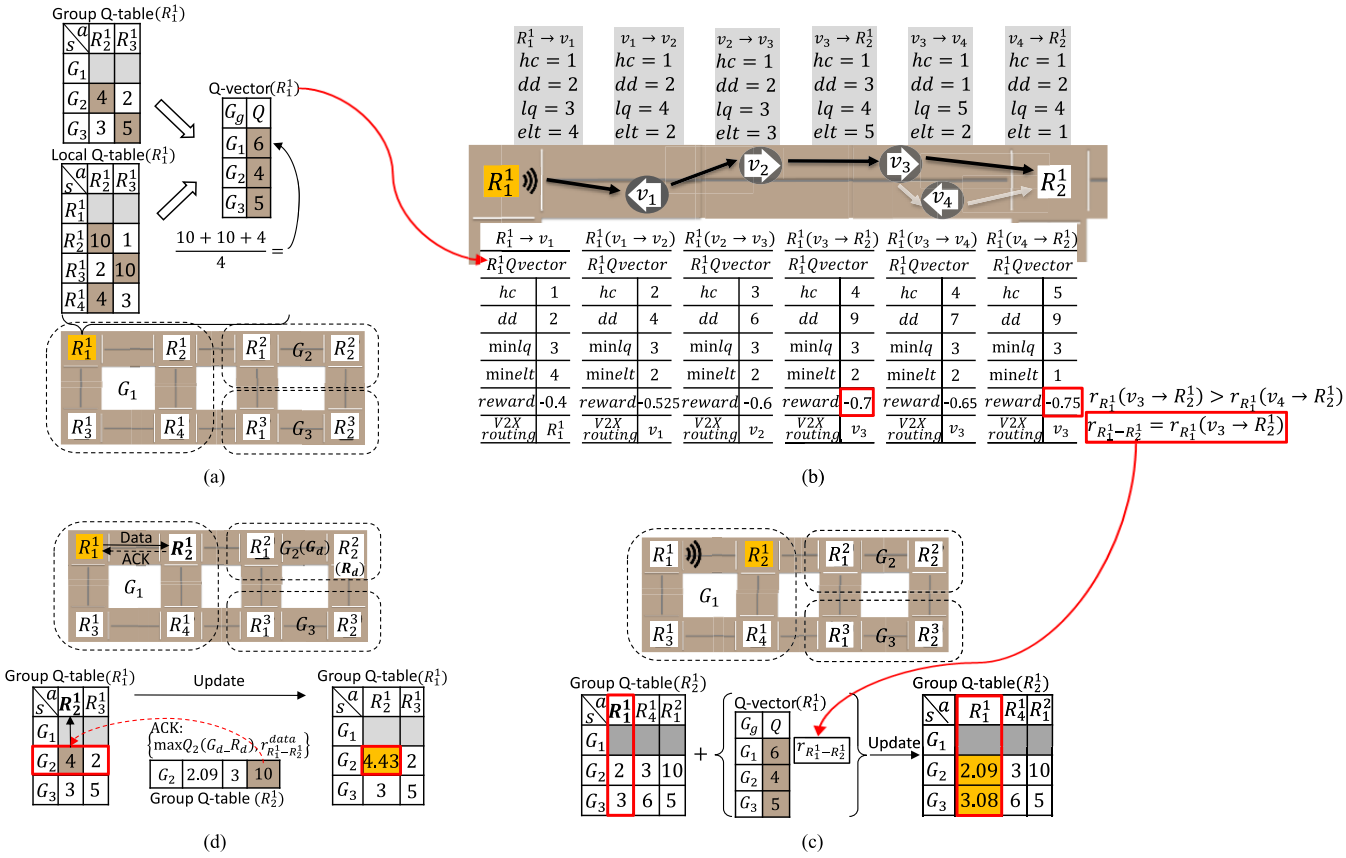

Fig. 6. Example scenario of the proposed method with (a) Q-vector formation, (b) V2X metric calculation during broadcast, (c) Q-table update with Q-vector and path reward, and (d) RSU selection and Q-table update in data routing.

deployed, with one at each intersection. The velocity of each vehicle was randomly determined to be within the predefined maximum allowable velocity. In addition, at each time four different source-destination pairs were randomly selected throughout the network. In this study, we assessed the network performances of three routing protocols: AODV [9], IV2XQ [42], and our proposed HQGR, to determine their effectiveness in the given scenarios. AODV is a well-known on-demand routing protocol for vehicular networks. IV2XQ is an intersection-based routing using RL. Furthermore, we evaluated the learning performance of IV2XQ and HQGR to analyze the effectiveness of these protocols to adapt and improve over time.

A. Network Performance Evaluation

In our approach, we strategically deployed RSUs at each intersection. This RSUs placement allows us to efficiently manage and control the vehicle network while minimizing broadcasting costs. The road topology shown in Fig. 1 is applied in Fig. 7 to Fig. 12. In this simulation, each RSU group consists of an equal number of RSUs. A network performance comparison was conducted in terms of packet delivery ratio, average end-to-end delay, number of broadcasts, and path lifetime.

1) Packet Delivery Ratio (PDR): The PDR refers to the percentage of data packets successfully delivered from a source to a destination in a network. It measures the proportion of packets arriving at their intended destinations compared to the total number of packets transmitted. The PDR was evaluated under four different conditions: the number of vehicles (default 200), maximum allowable velocity (default 10 m/s), broadcast interval (default 3 s), and data packet sending rate (default 1 pkt/s).

Fig. 7(a) depicts the PDR comparison for different numbers of vehicles. When the number of vehicles increased from 120 to 240, the PDR showed an upward trend for all the methods. Fig. 7(b) depicts a PDR comparison for different maximum allowable velocities of the vehicles. As the velocity increased from 5 to 20 m/s, the PDR of all methods decreased. Having more vehicles with lower velocities can lead to a higher PDR by extending the lifetime of the links. Fig. 7(c) depicts the PDR comparison with different broadcast intervals, and Fig. 7(d) depicts the PDR comparison with different data packet sending rates. By increasing the broadcast interval, the number of control messages transmitted per unit time decreased, further leading to fewer opportunities for packet collisions. However, when the data packet sending rate is 1 pkt/s, the movement of vehicles results in frequent changes in the network topology, and vehicles miss opportunities to forward messages to other vehicles that can help in message propagation, ultimately resulting in a lower PDR. Increasing the data sending rate from 1 to 4 pkt/s resulted in increased network congestion and packet loss. Overall, the proposed

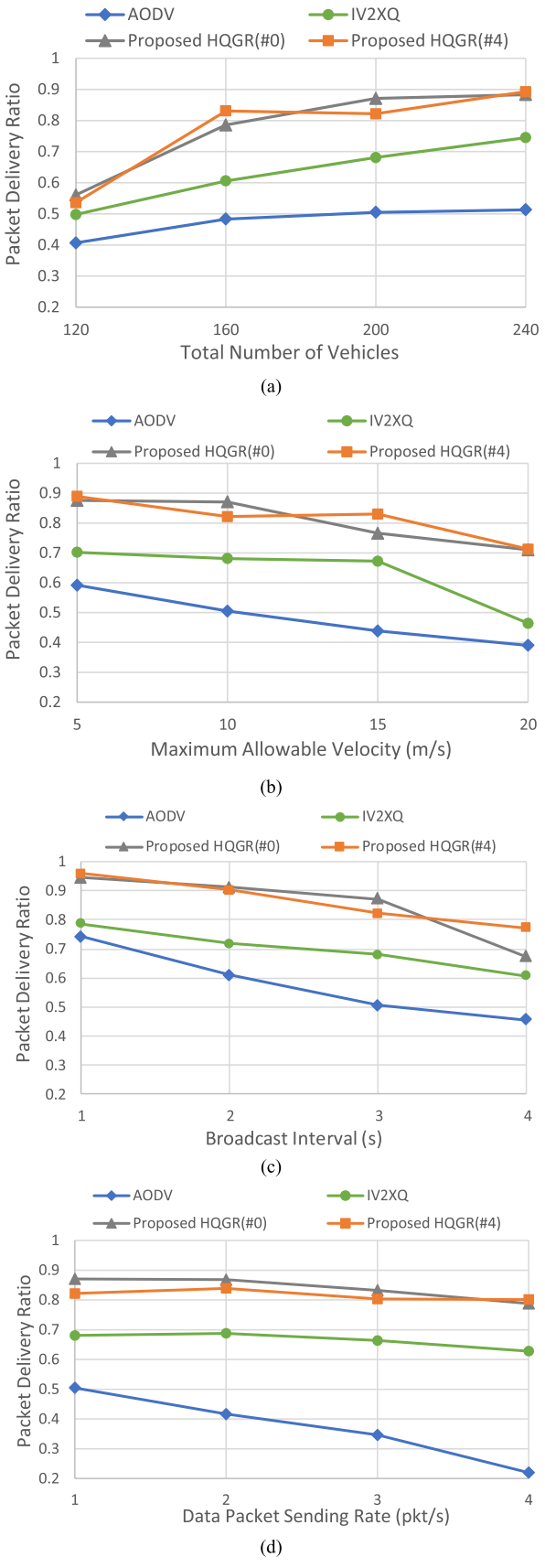

Fig. 7. PDR performance with (a) different numbers of vehicles, (b) different maximum allowable velocities, (c) different broadcast intervals, and (d) different data packet sending rates.

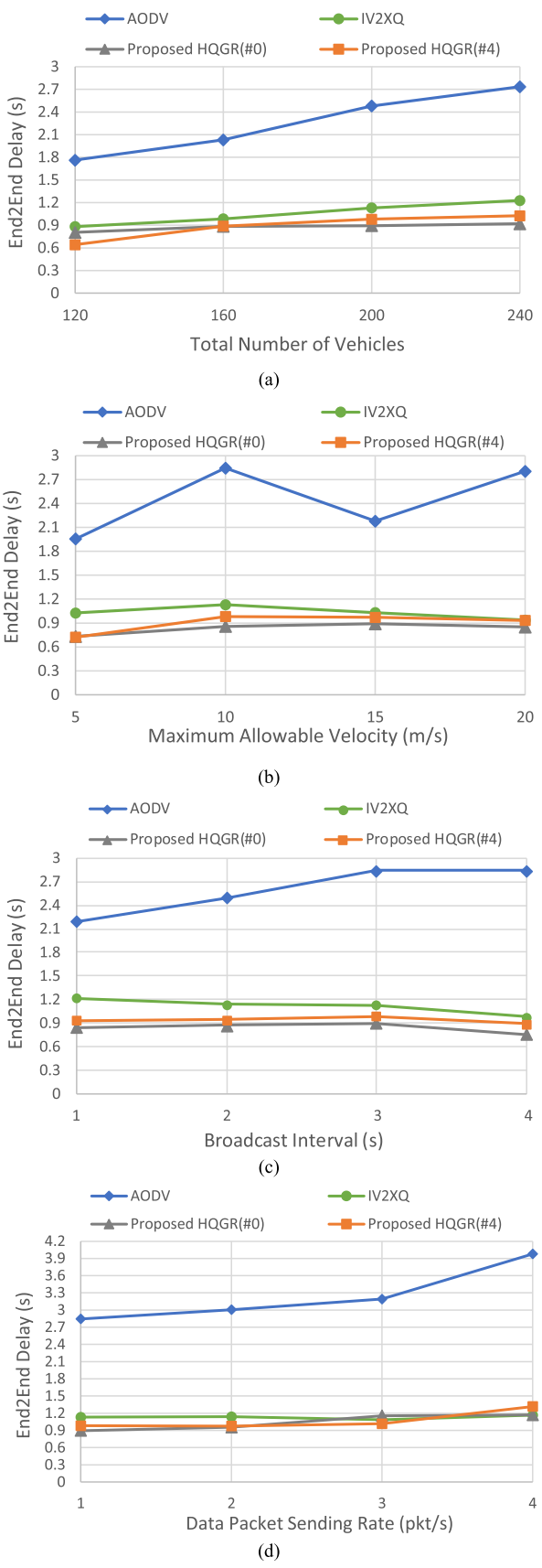

Fig. 8. Average E2E performance with (a) different numbers of vehicles, (b) different maximum allowable velocities, (c) different broadcast intervals, and (d) different data packet sending rates.

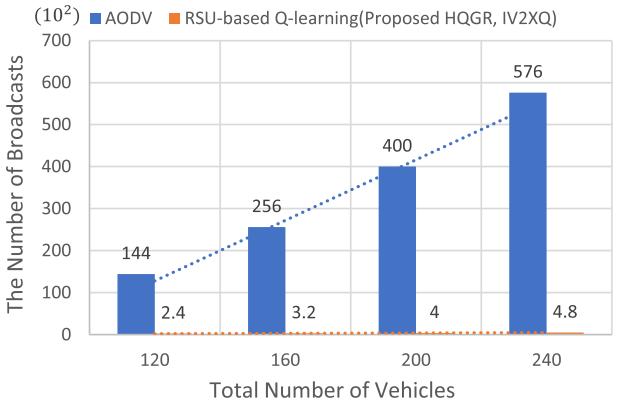

Fig. 9. Total number of broadcasts comparison.

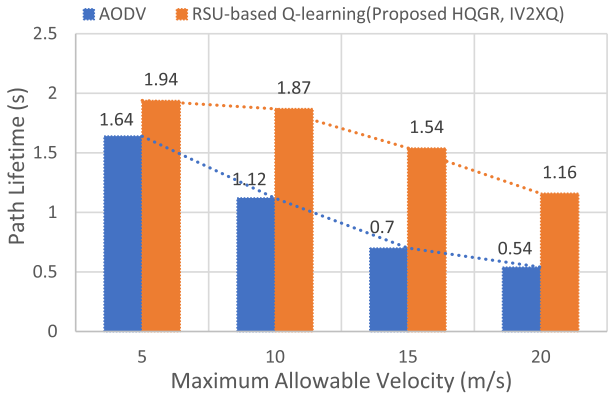

Fig. 10. Path lifetime comparison.

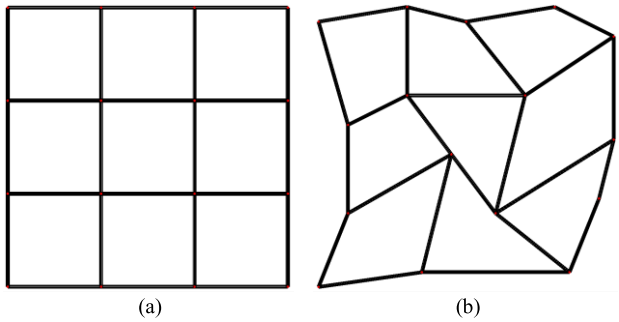

Fig. 11. Experimental topologies of (a) map 1, and (b) map 2.

HQGR, which includes two variants (with no group and four groups), achieved the highest PDR in all scenarios. In the proposed HQGR method, no group and four groups exhibited similar PDR performance results. IV2XQ had a relatively lower PDR when compared to that of the proposed method, whereas AODV performed the least. Note that the broadcast interval in AODV represents the route refresh time.

2) Average End-to-End (E2E) Delay: The E2E delay (also known as latency) is the amount of time it takes for the data to travel from the source to destination in a network. This is the time elapsed between the transmission of the first bit of a packet by the sender and the arrival of the last bit of the same packet at the receiver. The average E2E delay was also evaluated for four different conditions: number of vehicles, maximum allowable velocity, broadcast interval, and data packet sending rate.

Fig. 8(a) depicts a comparison of the E2E delay for different numbers of vehicles. When the number of vehicles increased, the E2E delay for the proposed HQGR and IV2XQ increased slightly, whereas AODV increased by almost twice because of the greater number of broadcasts. Fig. 8(b) depicts a comparison of the E2E delay for different maximum allowable velocities of the vehicles. A higher velocity implies a shorter link lifetime and faster movement, which affects the E2E delay. By comparing the E2E delay of the methods, we observed that the proposed method outperformed both IV2XQ and AODV. Fig. 8(c) and Fig. 8(d) depict an E2E delay comparison with different broadcast intervals and data packet sending rates, respectively. Channel congestion can be observed owing to the different sending rates. The proposed HQGR and IV2XQ achieved similar E2E delay with a small amount of variation, whereas AODV had the worst results.

3) The Number of Broadcasts: Broadcast messages require resources to be transmitted and processed by every node in the network, which can result in network bottlenecks and delays. In addition, excessive broadcasts can consume a large amount of bandwidth and increase the likelihood of collisions, which can further degrade network performance. As depicted in Fig. 9, we compared the numbers of broadcasts for different numbers of vehicles. In our study, we observed that AODV broadcasts spanned the entire network, further reaching all nodes within the network. However, the RSU-based broadcasts were limited to specific road segments. By adopting RSU-based broadcasts with a reduced propagation range, the overall network traffic could be better managed effectively, thus leading to improved efficiency and reduced congestion compared to the widespread broadcasts of AODV.

4) Path Lifetime: Fig. 10 depicts the comparison of the path lifetime. The path lifetime refers to the length of time for which a valid routing path in a network. It represents the minimum expected lifetime of any link among all links in a path, and is an important metric in network routing protocols that ensures that packets are routed through the most reliable path. As the maximum allowable velocity of vehicles increased, the path lifetime decreased. AODV's path lifetime decreased at a faster pace than that of RSU-based Q-learning methods and dropped below 1 s when the velocities were larger than 15 m/s. In contrast, the path lifetime of the RSU-based Q-learning methods decreased slowly and was maintained above 1 s even when the velocities reached 20 m/s.

5) Different Road Topologies: To facilitate simulation experiments under various environmental conditions, we introduced an additional experimental environment, as depicted in Fig. 11(b). In Fig. 12, we conducted a network performance comparison in terms of PDR and E2E delay for the two provided topology maps. We conducted 10 simulations for each given experimental condition (with parameters set to: number of vehicles = 200, maximum allowable velocity = 10 m/s, broadcast interval = 3 s, and data packet sending

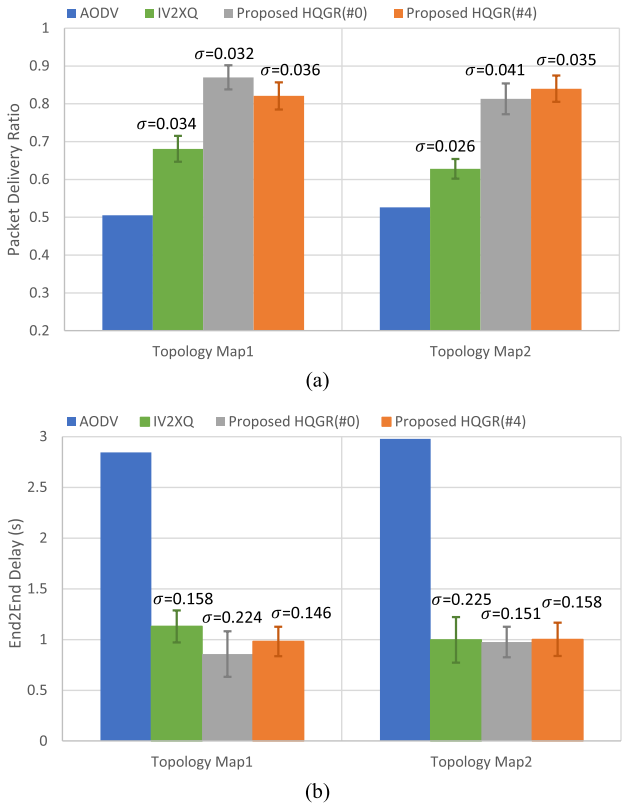

Fig. 12. Network performance comparison with standard deviations on (a) packet delivery ratio, and (b) end-to-end delay.

rate = 1 pkt/s) and calculated both the averages and standard deviations of the simulation results.

As depicted in Fig. 12, the RL-based approaches demonstrate superior performance compared to the traditional AODV. Specifically, the proposed HQGR outperforms the IV2XQ method for both Map 1 and Map 2. The PDR performance of the proposed method was approximately 20% to 34% higher than that of the IV2XQ method. For Map 1, the proposed method exhibited a reduction in delay performance of approximately 13% to 25% compared to IV2XQ. In the case of Map 2, they showed similar performance. It's worth noting that the proposed method had a slightly higher standard deviation compared to IV2XQ method, but both methods maintained very low standard deviation values.

B. Learning Performance Evaluation

1) Average Maximum Q-Value: By analyzing the average maximum Q-values, we assessed the ability of IV2XQ and HQGR to learn and determine the most promising routes towards the destination RSUs. The hyperparameters of the IV2XQ are outlined in [42]. Fig. 13(a) depicts the convergence characteristics as the number of episodes increased, in which the average maximum Q-values were normalized with the converged maximum value. The HQGR with four groups demonstrated the faster convergence speed, and exhibited a rapid convergence towards the optimal solutions with minimal fluctuations. IV2XQ, on the other hand, exhibited a slower convergence speed when compared to the HQGR with four groups. This suggests that IV2XQ can gradually converge towards optimal solutions, albeit at a slightly slower pace. To gain further insight and verify the impact of different groups on the learning process, we compared the average maximum Q-values of groups 0, 2, 4, and 8, as depicted in Fig. 13(b). The results demonstrated that the HQGR with eight groups exhibited the fastest convergence speed among all groups; however, its maximum Q-value indicated a slight decrease after fast convergence. In comparison, the HQGR with no group and two groups converged relatively slower and exhibited larger fluctuations when compared to the HQGR with four groups and eight groups. This indicates that the grouping strategy based on four distinct groups contributed to the efficiency and stability of the convergence process. Choosing the appropriate number of groups according to the topology is also important for achieving both fast convergence and stable performance.

2) Overhead for Q-Table and Q-Vector Maintenance: To assess the scalability of the proposed group-based approach, we increased the number of RSUs. This increase in the number of RSUs corresponds to an expansion in the intersections of the road topology used in the experiment. For example, 16, 32, and 64 RSUs represent topologies covering 24, 56, and 112 road segments, respectively. In Table IV, we provide a comparison of the Q-table size overhead associated with IV2XQ and proposed methods with different

TABLE III
SIMULATION PARAMETERS

Parameters	Value	Parameters	Value
Number of RSUs (NR)	16	Number of data packets from sources	800
Number of vehicles (NV)	120, 160, 200, 240	Transmission rate between nodes (μ_{nm})	1 Mbps
Broadcast interval	1, 2, 3, 4 s	Length of a message (l_{msg})	512 bytes
Data packet sending rate	1, 2, 3, 4 pkt/s	Transmission power for vehicle (P_{tx}^{v})	10 dBm
Maximum allowable velocity	5, 10, 15, 20 m/s	Transmission power for RSU (P_{tx}^{RSU})	15 dBm
Discount factor (γ)	0.9	Minimum decodable power (P_{min}^{rcv})	-92 dBm
Epsilon decaying parameters ($\varepsilon_{ini}, \varepsilon_{end}$)	0.5, 0.1	Antenna gains for transmitter and receiver (G_t, G_r)	2, 2 dBi
Learning rates (α_G, α_L)	0.3, 0.1	Operating frequency (f)	5.9 GHz
Terminal reward ($\mathcal{R}_T$)	4	Path loss exponent (β)	3
Weight parameters ($\omega_1, \omega_2, \omega_3, \omega_4$)	0.25	Shadowing variance (σ)	2

TABLE IV
Q-TABLE SIZE OVERHEAD COMPARISON

Average Q-Tables Size (Bytes)						
Methods \ Total RSUs	16	32	64	128	256	514
IV2XQ	512	1024	2048	4096	8192	16448
HQGR #0	512	1024	2048	4096	8192	16448
HQGR #2	320	576	1088	2112	4160	8256
HQGR #4	256	384	640	1152	2176	4224
HQGR #8	320	384	512	768	1280	2304

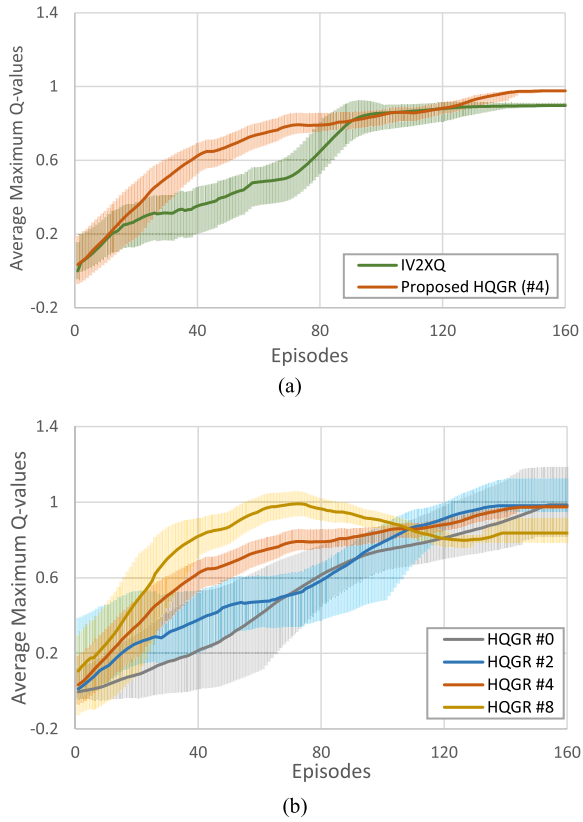

Fig. 13. Normalized average maximum Q-values comparison on (a) IV2XQ and propose HQGR with four groups, and (b) different group configurations.

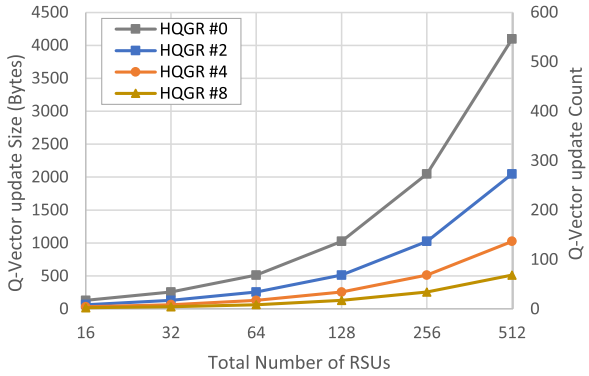

Fig. 14. Q-vector size and update count comparison for different group sizes.

group numbers as increasing the number of RSUs. Q-values were stored using a double data type. When no grouping was implemented, we observed a significant increase in the Q-table size as the topology (number of RSUs) increased, which indicates that the memory requirement for storing the Q-values becomes substantially larger as the urban environment becomes more extensive. The use of groups allows for more efficient representation and utilization of memory, further resulting in reduced overhead and optimized storage of Q-tables.

Fig. 14 depicts the comparison of the Q-vector size and number of update counts for different numbers of RSUs and group sizes. As mentioned in the previous section, the Q-vector was included in the QVRS message and broadcasted to the network, thus making its size a critical factor that affects communication efficiency and RSU local processing. When no grouping was applied, we observed that an increase in the number of RSUs leads to greater communication and more updating overhead for the Q-vectors. However, when multiple groups were utilized, the overhead of Q-vector maintenance could be dramatically reduced, particularly for large-scale topologies.

TABLE V
LEARNING COMPLEXITY AND PATH MAINTENANCE COMPARISON

Learning complexity (at each RSU R_i)		
Methods	**Overhead for updating and maintaining Q-tables**	
HQGR	Within a group: O((number of group member RSUs) × (number of neighboring RSUs)) $= O(NG \times NNR^i)$ Between groups: O((number of RSU groups) × (number of neighboring RSUs)) $= O(NR_G \times NNR^i)$	
IV2XQ	O((total number of RSUs) × (number of neighboring RSUs)) $= O(NR \times NNR^i)$	
Path maintenance overhead		
Methods	**Number of broadcasts at each interval**	**Path information at each vehicle**
HQGR	O(total number of vehicles) $= O(NV)$	Road segment RSU pair $= O(1)$
AODV	O((number of source vehicles) × (total number of vehicles)) $= O(NV^2)$	All source to destination vehicle pairs $= O(NV^2)$

3) Learning and Path Maintenance Complexity: In Table V, we conducted an analysis of the computational overhead for convergence in Q-table learning and the overhead related to table memory management for the proposed HQGR and the compared method IV2XQ. We also have compared the path maintenance overhead in terms of the total number of broadcast packets sent to the network at each broadcast interval and the required path information at each vehicle with AODV. In the proposed method, the local Q-table update and memory management overhead at each RSU R_i are proportional to the product of the number of group member RSUs (NR_G) and the number of neighboring RSUs (NNR^i). For group Q-table update, they are proportional to the product of the total number of groups (NG) and the number of neighboring RSUs. On the other hand, in IV2XQ, the overhead is proportional to the product of the total number of RSUs (NR) and the number of neighboring RSUs. It should be noted that generally $(NR_G \times NNR^i) \ll (NR \times NNR^i)$ and $(NG \times NNR^i) \ll (NR \times NNR^i)$, making the complexity of the proposed methods much smaller than that of IV2XQ. In the proposed method, the total number of broadcast messages for network control is proportional to the total number of vehicles (NV) because broadcast occurs on each road segment. In AODV, since the route request packets from all source vehicles are broadcast to all vehicles in the network, it is proportional toNV^2. Furthermore, in the proposed method, for path management, each vehicle only manages path information to the RSUs at both ends of the segment where the vehicle is located, resulting in constant complexity $O(1)$. In contrast, AODV needs to manage path information for all (source, destination) pairs in its forward and backward tables, which is proportional to NV^2.

VI. Conclusion

This paper presents HQGR, a hierarchical Q-learning-based routing algorithm that integrates grouped RSU for urban VANETs. The HQGR utilizes grouped RSUs to provide a comprehensive view of the area and local perspectives within specific groups at intersections. The algorithm utilizes multi-agent hierarchical Q-learning to train group and local Q-tables. The group Q-table estimates the reward for reaching the destination group, whereas the local Q-table focuses on reaching the destination RSU inside a group. This hierarchical approach enables a more targeted and efficient routing decision-making process. To update the V2X evaluation metric, QVRS control messages carrying a Q-vector with group information are broadcasted periodically on each segment. This Q-vector serves as a representation of the Q-table and facilitates evaluation of the integrated road-segment V2X metric. This metric considers factors such as the delivery delay, hop count, link quality, and expected lifetime. It is used for V2V routing decisions, and serves as the reward function in Q-learning for R2R routing decisions. The simulation results demonstrated the effectiveness of HQGR in reducing network overhead, extending path lifetime, and achieving a high PDR and low average E2E delay. In addition, our method contributes to accelerating the learning procedure and enables more efficient communication in VANETs. As part of our future research agenda, we intend to collaborate with local authorities and national research support agencies to test and validate the proposed method in real-world environments under diverse traffic conditions and dynamic environmental changes.

References

[1] S. Harrabi, I. B. Jaafar, and K. Ghedira, "Survey on IoV routing protocols," *Wireless Pers. Commun.*, vol. 128, no. 2, pp. 791–811, 2023.

[2] Q. Yang and S.-J. Yoo, "Optimal UAV path planning: Sensing data acquisition over IoT sensor networks using multi-objective bio-inspired algorithms," *IEEE Access*, vol. 6, pp. 13671–13684, 2018.

[3] A. Beishenalieva and S.-J. Yoo, "Multiobjective 3-D UAV movement planning in wireless sensor networks using bioinspired swarm intelligence," *IEEE Internet Things J.*, vol. 10, no. 9, pp. 8096–8110, May 2023.

[4] I. Wahid, A. A. Ikram, M. Ahmad, S. Ali, and A. Ali, "State of the art routing protocols in VANETs: A review," *Proc. Comput. Sci.*, vol. 130, pp. 689–694, Jan. 2018.

[5] T. Chatterjee, R. Karmakar, G. Kaddoum, S. Chattopadhyay, and S. Chakraborty, "A survey of VANET/V2X routing from the perspective of non-learning- and learning-based approaches," *IEEE Access*, vol. 10, pp. 23022–23050, 2022.

[6] Q. Yang, S. J. Jang, and S. J. Yoo, "Q-learning-based fuzzy logic for multi-objective routing algorithm in flying ad hoc networks," *Wireless Pers. Commun.*, vol. 113, pp. 115–138, Jan. 2020.

[7] W.-K. Yun and S.-J. Yoo, "Q-learning-based data-aggregation-aware energy-efficient routing protocol for wireless sensor networks," *IEEE Access*, vol. 9, pp. 10737–10750, 2021.

[8] B. Karp and H.-T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. 6th Annu. Int. Conf. Mobile Comput. Netw.*, 2000, pp. 243–254.

[9] C. Perkins, E. Belding-Royer, and S. Das, *Ad Hoc On-Demand Distance Vector (AODV) Routing*, document RFC3561, 2003.

[10] T. Karunathilake and A. Förster, "A survey on mobile roadside units in VANETs," *Vehicles*, vol. 4, no. 2, pp. 482–500, 2022.

[11] A. Mchergui, T. Moulahi, and S. Zeadally, "Survey on artificial intelligence (AI) techniques for vehicular ad-hoc networks (VANETs)," *Veh. Commun.*, vol. 34, Apr. 2022, Art. no. 100403.

[12] A. R. Abdellah, A. Muthanna, and A. Koucheryavy, "Energy estimation for VANET performance based robust neural networks learning," in *Proc. 22nd Int. Conf. Distrib. Comput. Commun. Netw. (DCCN)*, Moscow, Russia. Cham, Switzerland: Springer, Sep. 2019, pp. 127–138, 2019.

[13] T. Liu, S. Azarm, and N. Chopra, "Integrating optimal vehicle routing and control with load-dependent vehicle dynamics using a confidence bounds for trees-based approach," *J. Dyn. Syst., Meas., Control*, vol. 142, no. 4, Apr. 2020, Art. no. 041006.

[14] Y. Azzoug and A. Boukra, "Bio-inspired VANET routing optimization: An overview: A taxonomy of notable VANET routing problems, overview, advancement state, and future perspective under the bio-inspired optimization approaches," *Artif. Intell. Rev.*, vol. 54, pp. 1005–1062, Feb. 2021.

[15] M. Elhoseny and K. Shankar, "Energy efficient optimal routing for communication in VANETs via clustering model," in *Emerging Technologies for Connected Internet of Vehicles and Intelligent Transportation System Networks: Emerging Technologies for Connected and Smart Vehicles*. Cham, Switzerland: Springer, 2020, pp. 1–14.

[16] S. Padakandla, "A survey of reinforcement learning algorithms for dynamically varying environments," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–25, 2021.

[17] O. Jafarzadeh, M. Dehghan, H. Sargolzaey, and M. M. Esnaashari, "A model-based reinforcement learning protocol for routing in vehicular ad hoc network," *Wireless Pers. Commun.*, vol. 123, no. 1, pp. 975–1001, 2022.

[18] Q. Yang and S.-J. Yoo, "Grouped intersection-based routing using reinforcement learning for urban VANETs," in *Proc. 13th Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2022, pp. 1855–1858.

[19] M. Al-Rabayah and R. Malaney, "A new scalable hybrid routing protocol for VANETs," *IEEE Trans. Veh. Technol.*, vol. 61, no. 6, pp. 2625–2635, Jul. 2012.

[20] J. Bernsen and D. Manivannan, "RIVER: A reliable inter-vehicular routing protocol for vehicular ad hoc networks," *Comput. Netw.*, vol. 56, no. 17, pp. 3795–3807, 2012.

[21] C. Tripp-Barba, A. Zaldívar-Colado, L. Urquiza-Aguiar, and J. A. Aguilar-Calderón, "Survey on routing protocols for vehicular ad hoc networks based on multimetrics," *Electronics*, vol. 8, no. 10, p. 1177, 2019.

[22] S. Kassir, G. de Veciana, N. Wang, X. Wang, and P. Palacharla, "Enhancing cellular performance via vehicular-based opportunistic relaying and load balancing," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2019, pp. 91–99.

[23] K. N. Qureshi, F. Bashir, and A. H. Abdullah, "Distance and signal quality aware next hop selection routing protocol for vehicular ad hoc networks," *Neural Comput. Appl.*, vol. 32, pp. 2351–2364, Jun. 2019.

[24] K. Liu, J. K. Y. Ng, V. C. S. Lee, S. H. Son, and I. Stojmenovic, "Cooperative data scheduling in hybrid vehicular ad hoc networks: VANET as a software defined network," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1759–1773, Jun. 2016.

[25] M. Jerbi, R. Meraihi, S.-M. Senouci, and Y. Ghamri-Doudane, "GyTAR: Improved greedy traffic aware routing protocol for vehicular ad hoc networks in city environments," in *Proc. 3rd Int. Workshop Veh. Ad Hoc Netw.*, Sep. 2006, pp. 88–89.

[26] N. Alsharif and X. Shen, "*i*CAR-II: Infrastructure-based connectivity aware routing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 4231–4244, Aug. 2016.

[27] T. S. J. Darwish, K. A. Bakar, and K. Haseeb, "Reliable intersection-based traffic aware routing protocol for urban areas vehicular ad hoc networks," *IEEE Intell. Transp. Syst. Mag.*, vol. 10, no. 1, pp. 60–73, Spring. 2018.

[28] Z. Ahmed, S. Naz, and J. Ahmed, "Minimizing transmission delays in vehicular ad hoc networks by optimized placement of road-side unit," *Wireless Netw.*, vol. 26, pp. 2905–2914, Jan. 2020.

[29] N. N. Srinidhi, C. S. Sagar, S. D. Chethan, J. Shreyas, and S. M. D. Kumar, "An improved PRoPHET—Random forest based optimized multi-copy routing for opportunistic IoT networks," *Internet Things*, vol. 11, Sep. 2020, Art. no. 100203.

[30] J. Nadarajan and J. Kaliyaperumal, "QOS aware and secured routing algorithm using machine intelligence in next generation VANET," *Int. J. Syst. Assurance Eng. Manage.*, vol. 2021, pp. 1–12, Mar. 2021.

[31] Y. Sun, S. Ravi, and V. Singh, "Adaptive activation thresholding: Dynamic routing type behavior for interpretability in convolutional neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4937–4946.

[32] J. Wu, M. Fang, and X. Li, "Reinforcement learning based mobility adaptive routing for vehicular ad-hoc networks," *Wireless Pers. Commun.*, vol. 101, pp. 2143–2171, May 2018.
[33] Y.-R. Chen, A. Rezapour, W.-G. Tzeng, and S.-C. Tsai, "RL-routing: An SDN routing algorithm based on deep reinforcement learning," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 3185–3199, Oct. 2020.
[34] L. H. Teixeira and Á. Huszák, "Reinforcement learning in path lifetime routing algorithm for VANETs," *J. Inf. Sci. Eng.*, vol. 39, pp. 129–147, Jan. 2023.
[35] J. Lansky, A. M. Rahmani, and M. Hosseinzadeh, "Reinforcement learning-based routing protocols in vehicular ad hoc networks for intelligent transport system (ITS): A survey," *Mathematics*, vol. 10, no. 24, p. 4673, 2022.
[36] F. Li, X. Song, H. Chen, X. Li, and Y. Wang, "Hierarchical routing for vehicular ad hoc networks via reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1852–1865, Feb. 2019.
[37] D. Zhang, T. Zhang, and X. Liu, "Novel self-adaptive routing service algorithm for application in VANET," *Appl. Intell.*, vol. 49, pp. 1866–1879, May 2019.
[38] X. Yang, W. Zhang, H. Lu, and L. Zhao, "V2V routing in VANET based on heuristic Q-learning," *Int. J. Comput. Commun. Control*, vol. 15, no. 5, p. 3328, Jul. 2020.
[39] M. Yuan, "Research on VANET routing algorithm based on reinforcement learning," Xi'an Univ. Electron. Sci. Technol., Xi'an, China, Tech. Rep., 2017.
[40] N. Geng et al., "A reinforcement learning framework for vehicular network routing under peak and average constraints," *IEEE Trans. Veh. Technol.*, vol. 72, no. 5, pp. 6753–6764, Jan. 2023.
[41] J. Wu, M. Fang, H. Li, and X. Li, "RSU-assisted traffic-aware routing based on reinforcement learning for urban vanets," *IEEE Access*, vol. 8, pp. 5733–5748, 2020.
[42] L. Luo, L. Sheng, H. Yu, and G. Sun, "Intersection-based V2X routing via reinforcement learning in vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5446–5459, Jun. 2022.
[43] M. U. Khan, M. Hosseinzadeh, and A. Mosavi, "An intersection-based routing scheme using Q-learning in vehicular ad hoc networks for traffic management in the intelligent transportation system," *Mathematics*, vol. 10, no. 20, p. 3731, 2022.
[44] A. M. Rahmani et al., "A Q-learning and fuzzy logic-based hierarchical routing scheme in the intelligent transportation system for smart cities," *Mathematics*, vol. 10, no. 22, p. 4192, 2022.
[45] A. Lolai et al., "Reinforcement learning based on routing with infrastructure nodes for data dissemination in vehicular networks (RRIN)," *Wireless Netw.*, vol. 28, no. 5, pp. 2169–2184, 2022.
[46] S.-J. Yoo and S.-H. Choi, "Indoor AR navigation and emergency evacuation system based on machine learning and IoT technologies," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 20853–20868, Nov. 2022.

Qin Yang (Graduate Student Member, IEEE) received the B.E. degree in communication and information engineering from the Chongqing University of Posts and Telecommunications, Chongqing, China, in 2016, and the M.S. degree in electrical and computer engineering from Inha University, Incheon, South Korea, in 2018, where she is currently pursuing the Ph.D. degree with the Multimedia Network Laboratory. Her research interests include machine learning, reinforcement learning, wireless sensor networks, vehicular networks, and the Internet of Things.

Sang-Jo Yoo (Member, IEEE) received the B.S. degree in electronic communication engineering from Hanyang University, Seoul, South Korea, in 1988, and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology in 1990 and 2000, respectively. From 1990 to 2001, he was a member of the Technical Staff with the KT Research and Development Group. From 1994 to 1995 and from 2007 to 2008, he was a Guest Researcher with the National Institute of Standards and Technology, USA. Since 2001, he has been with Inha University, where he is currently a Professor with the Electrical and Computer Engineering Department. His current research interests include machine learning, cognitive radio networks, vehicular networks, wireless sensor networks, and the Internet of Things.