IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 December 2023, accepted 29 January 2024, date of publication 14 February 2024, date of current version 22 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3366280

RESEARCH ARTICLE

RHRA-DRL: RSU-Assisted Hybrid Road-Aware Routing Using Distributed Reinforcement Learning in Internet of Vehicles

JOO-HYUNG PARK, QIN YANG, (Graduate Student Member, IEEE), AND SANG-JO YOO, (Member, IEEE)
Department of Electrical and Computer Engineering, Inha University, Incheon 402-751, South Korea
Corresponding author: Sang-Jo Yoo (sjyoo@inha.ac.kr)

This work was supported by the Inha University Research Grant.

ABSTRACT In this paper, we propose a novel RSU-assisted hybrid road-aware routing algorithm, RHRA-DRL, designed for urban vehicular networks to optimize real-time data delivery considering dynamic road conditions. The algorithm minimizes broadcast overhead and efficiently determines optimal routing paths by incorporating two key components. Firstly, a multihop road-segment reward-based ad-hoc (RRAH) routing algorithm is introduced to adaptively respond to changing vehicle topologies within road segments. Rewards are calculated based on performance metrics, and the segment reward integrates into RSU-to-RSU (R2R) routing. Secondly, a distributed Q-learning-based road-aware (DQRA) routing algorithm determines RSUs traversed during data transmission using a decentralized agent reinforcement learning approach. The combination of these algorithms in RHRA-DRL ensures effective and consistent path establishment with a unified reward system. Simulation results demonstrate the superiority of RHRA-DRL over AODV in Internet of Vehicles (IoV) networks, showcasing enhanced communication, prolonged link lifetime, rapid establishment and repair of routing paths, and reduced overhead.

INDEX TERMS Data routing, distributed reinforcement learning, intelligent transportation system, Internet of Vehicles, vehicular ad hoc networks.

I. INTRODUCTION

The accelerated urbanization and the surging proliferation of vehicles on roadways necessitate a critical enhancement of transportation systems. Addressing this challenge extends beyond the purview of mechanical engineering, now encompassing the expertise of computer professionals. Consequently, a spectrum of methods and solutions is materializing within the domain of the Internet of Vehicles (IoV). IoV, an outgrowth of the Internet of Things (IoT), constitutes a dynamic network infrastructure establishing connectivity among vehicles, users, and an array of intelligent devices through the Internet. The number of vehicles seamlessly integrated into IoV systems is progressively on the rise, with each vehicle assuming the role of a network node [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Wen Chen.

Vehicular ad-hoc networks (VANETs) represent a specialized subset of IoV crucial for augmenting road safety, operational efficiency, and user convenience. The distinctive characteristic of VANETs lies in the intricate nature of routing decisions based on node locations, owing to the inherent high mobility of these networks [2], [3]. Furthermore, VANETs operate without centralized physical control, and their expansive scope is marked by the perpetual variability in vehicle speeds, resulting in frequent changes in vehicle positions and network topology. Additionally, the transitory nature of VANETs necessitates frequent exchange of information between vehicles and roadside units (RSUs) [4].

Within the realm of VANETs, a diverse range of communication models is deployed, including vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and infrastructure-to-infrastructure (I2I) communications, all of which facilitate seamless communication among vehicles [5]. The ad-hoc

on-demand distance vector (AODV) emerges as a widely employed approach that considers the hop count for V2V communication [6]. Another prominent routing strategy, Dijkstra's algorithm, is frequently applied to determine the shortest paths [7]. The optimized link-state routing (OLSR) algorithm represents an optimization of existing link-state algorithms tailored to meet the requirements of mobile wireless LANs [8].

Nevertheless, urban VANETs confront a multitude of challenges. Traditional cost-based proactive routing and on-demand reactive routing methods encounter challenges in promptly adapting to the dynamic environmental conditions inherent in IoV [9]. Scaling issues arise during the reconstruction of the entire network or reconfiguration of routes due to minor adjustments in the road network. Frequent route reconstructions become necessary to accommodate environmental fluctuations, leading to significant signal transmission and overhead. Consequently, recent studies have witnessed a shift towards the adoption of reinforcement learning (RL) solutions for IoV routing. However, challenges such as slow convergence, a limited distributed system learning structure, and reduced learning efficiency in short-range routing between vehicles due to frequent topology changes must be effectively addressed.

The primary contributions of this study are succinctly summarized as follows:

- Introduction of road-segment reward-based ad-hoc (RRAH) routing: We propose the RRAH routing method for multi-hop vehicular route discovery and transmission within a road segment. The Q-route discovery request-and-reply packets are meticulously designed to update vehicle communication information during broadcasting. This strategic approach effectively reduces the broadcast scope and expedites the repair process.
- Development of a comprehensive reward function: We have devised a reward function to evaluate each vehicle communication path, taking into account factors such as vehicle movements and channel conditions. This assessment considers total delay, total hops, minimum link lifetime, and minimum link quality, aiding in the determination of the optimal next node for relaying in V2V ad-hoc routing. Moreover, this reward function is hybridized in RSU-to-RSU (R2R) routing to ascertain the next road segment.
- Introduction of distributed reinforcement learning (DRL) routing scheme: We propose a DRL routing scheme that adeptly responds to frequent changes in topology and dynamic environments. Distributed Q-learning-based road-aware (DQRA) routing is implemented among RSUs, with multiple RSUs serving as agents at intersections to facilitate R2R routing. This enhances awareness of road traffic and ensures timely responses to situational changes.
- Presentation of RSU-assisted hybrid road-aware distributed reinforcement learning (RHRA-DRL) routing algorithm: We introduce the RHRA-DRL routing algorithm for the IoV, executing a two-step routing process involving RRAH and DQRA. This innovative approach establishes real-time data routing paths comprising a sequential set of vehicles and RSUs, thereby minimizing overheads, enhancing environmental awareness, and dynamically adapting to network changes.

The remainder of this paper is structured as follows: Section II presents related work. Section III elaborates on the proposed system model and the distributed reinforcement learning-based RSU-assisted hybrid road-aware routing algorithm. Section IV presents the simulation results and performance comparisons with existing approaches. Finally, Section V concludes the paper.

II. RELATED WORK

This section provides an in-depth exploration of data routing research, covering both traditional methods and intelligent approaches to IoV. The greedy perimeter stateless routing (GPSR) algorithm makes routing decisions based on the local information regarding neighboring routers in a network topology [10]. Proactive AODV (Pro-AODV)-based approaches use AODV routing table information to minimize congestion in VANETs while maintaining other performance indicators at acceptable levels [11]. Ant colony optimization (ACO) and dynamic source routing (DSR) protocol-based routing methods have been proposed for reactive path discovery protocols that consider speed and fading conditions [12]. In a dynamic urban environment with constantly changing road conditions, focusing solely on V2V routing for data delivery is challenging.

Research has also been conducted on V2V and R2R routing with RSU support in real-time changing road environments. GyTAR is a geographic routing protocol based on intersections in urban environments, where intersections are dynamically selected based on changes in vehicle traffic and distance. Vehicles select the nearest vehicle as the next node towards the destination intersection [13]. Direction geographic source routing (DGSR) protocol combines directional delivery strategy with geographic source routing (GSR) in urban environments. The source node obtains the location information of the destination node and calculates the shortest path to the destination using Dijkstra's algorithm [14]. A new infrastructure-based connection-aware routing protocol called iCAR-II was proposed to enable multihop vehicle applications [15]. Another reliable traffic-aware routing protocol was proposed in [16], which selects the next hop based on the road structure, neighbor position prediction, received signal strength, and mobility information from the neighbors.

Recently, artificial intelligence (AI) technologies have seen active advancements in routing path optimization using deep learning. In [17], a bat algorithm served as a vital factor, providing inputs that enabled the deep neural network (DNN) routing algorithm to make optimal decisions, and thus alleviate network traffic congestion. DNN efficiently processes routing decisions at a faster pace and offer network

solutions for setting optimal paths, thereby reducing network congestion more swiftly. Simple modifications to the stochastic gradient descent (SGD) framework provide dynamic and expectation-maximization (EM) routing behaviors in convolutional neural networks (CNN) [18]. Neural networks offer advantages in terms of performance enhancement. However, it is worth noting that the use of large neural networks may introduce increased access delays, additional resources, and memory usage.

RL has garnered increasing interest in recent research on data routing. A Q-learning algorithm is used to infer network state information and ensure real-time route availability through unicast control packets in the Q-learning AODV (QLAODV) routing protocol [19]. The RSU-supported Q-learning-based traffic-aware routing (QTAR) employs a Q-greedy geographical forwarding (QGGF) strategy to achieve distributed V2V Q-learning, reduce delivery delays, and mitigate the impact of fast vehicle movement on route sensitivity [20]. RL-based routing with infrastructure node data dissemination in a vehicle network, called RRIN, uses two Q-routing methods: road model segment selection and intermediate vehicle selection. R2R routing considers the shortest distance and a higher connectivity distribution for efficient road-segment determination, whereas V2V routing considers factors such as vehicle speed differences and number of data packets to find efficient next-hop deliverers [21]. In [22], an adaptive intersection-based distributed routing (IDR) method was proposed to enhance adaptability to dynamically changing networks by considering the impact of vehicle movement, traffic signals, and road traffic conditions on the routing performance in urban VANETs. An intersection-based vehicle-to-everything (V2X) routing protocol was proposed, leveraging Q-learning to learn routing strategies based on past traffic flows and monitoring real-time network conditions [23]. RL enhances the efficiency and adaptation of data routing. However, it is crucial to note that employing RL for both R2R and V2V routing can lead to convergence issues. Because the convergence speed of RL is relatively slow compared to short-term inter-vehicle communication and changes in the driving environment, routing path establishment using RL for V2V communication within a road segment has problems.

III. RSU-ASSISTED HYBRID ROAD-AWARE ROUTING WITH DISTRIBUTED Q-LEARNING (RHRA-DRL) IN VEHICULAR NETWORKS

A. SYSTEM MODEL

Given the dynamic nature of vehicular topology in expansive urban areas, existing routing strategies encounter numerous challenges. While fully distributed AODV-based reactive methods offer optimization, their notable control overhead poses concerns. To address this, this study introduces a constrained broadcast process, specifically targeting a single road segment. This approach utilizes a reward-based Q-route discovery request-and-reply mechanism to optimize communication.

Conventional centralized routing schemes, while effective, introduce frequent communication exchanges, potentially resulting in access delays and resource wastage. These schemes may struggle to promptly capture local condition changes. In response, this paper advocates for a distributed reinforcement learning scheme tailored to adapt to the dynamic and nonstationary nature of the environment. Notably, the focus is on the independent learning of intersection-based routing path decisions.

Within the DRL framework, RSUs function as multi-agents, leveraging their capabilities to expedite learning convergence and enhance decision-making processes. Distributed Q-learning incorporates within-road-segment rewards, fostering road awareness at the between-road segment level. This integrated approach promotes adaptive routing, enhancing the system's ability to respond to changing conditions. The decentralized and responsive nature of this system enhances its efficiency in traffic management, offering a more adaptive and nuanced approach to the challenges posed by dynamic urban environments.

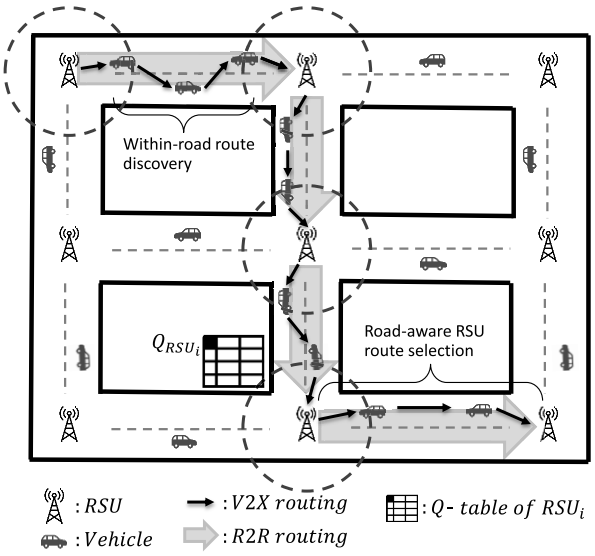

FIGURE 1. Network scenario.

The overall network scenario is depicted in Fig. 1, where each RSU strategically occupies positions at intersections. Vehicles navigate freely along road segments connecting neighboring RSUs, demonstrating random speeds and directions. Our proposed routing model comprises two key components:

i) RSU-to-RSU routing path decision process: This component involves determining the optimal next RSU for data transfer from the source RSU to the destination RSU. It employs RSU-assisted distributed Q-learning to facilitate the decision-making process. This method ensures efficient and informed routing decisions in the transfer of data between RSUs.

ii) Inter-vehicle data transfer within a road segment: The second aspect pertains to data transfer occurring within a road segment between two adjacent RSUs. This process encompasses constrained RSU broadcasting and reward-based intra-segment path decisions. The constrained broadcast approach optimizes communication within the specific road segment, enhancing the efficiency of data transfer between vehicles in the designated area.

In summary, our routing model encompasses both inter-RSU and intra-segment data transfer mechanisms, utilizing RSU-assisted distributed Q-learning and reward-based decision processes. This dual approach aims to optimize routing efficiency within the vehicular network, addressing the challenges posed by random vehicle speeds and directions in dynamic urban environments.

Through the integration of both RSU-to-RSU data transfer paths using distributed Q-learning and intra-segment inter-vehicle data transfer with constrained RSU broadcasting and reward-based decisions, we have developed a hybrid routing scheme aimed at achieving road awareness and deriving optimal routing paths. We define the set of RSUs as $RSU \triangleq \{RSU_1, \cdots, RSU_i, \cdots, RSU_I\}$, where I is the total number of RSUs. Each RSU maintains an independent knowledge table called Q-table, and the set of these Q-tables is denoted as $Q_{RSU} \triangleq \{Q_{RSU_1}, \cdots, Q_{RSU_i}, \cdots, Q_{RSU_I}\}$, where Q_{RSU_i} represents the Q-table of RSU_i.

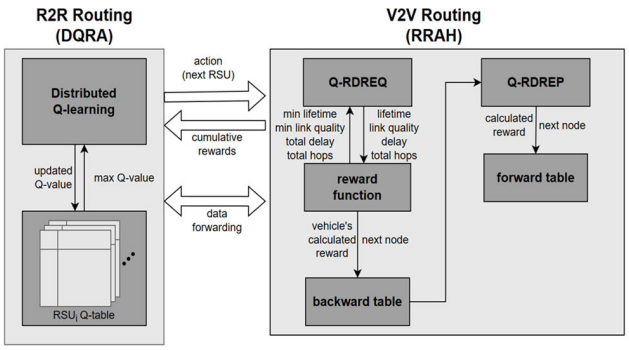

FIGURE 2. Block diagram.

Fig. 2 illustrates a block diagram of the proposed method, where the RRAH is employed in V2V routing for communication establishment and evaluation. The Q-route discovery requests (Q-RDREQs) are strategically designed to update vehicle communication information and construct a backward table during broadcasting. A reward function is implemented to assess various factors, including total delay, total hops, minimum link lifetime, and minimum link quality, in the evaluation of vehicle communication information. Subsequently, Q-route discovery replies, denoted as Q-RDREPs, are utilized to construct the forward table for vehicles. For R2R routing, a distributed Q-learning scheme named DQRA is deployed. Each RSU maintains a Q-table for iterative learning and updates the Q-value by receiving rewards from V2V routing. These Q-values accurately depict an RSU's experience in selecting the next RSU to forward the data packets, contributing to the efficient establishment of routing paths in both V2V and R2R scenarios.

B. ROAD-SEGMENT REWARD-BASED AH HOC (RRAH) ROUTING

In this subsection, we provide a detailed overview of the RRAH approach for discovering and transmitting multi-hop vehicle routes within a road segment, as illustrated in Fig. 3. In the proposed RRAH routing mechanism, an RSU initiates a broadcast of the Q-RDREQ packet, which is received by vehicles within its transmission range. These vehicles subsequently rebroadcast the packet until it reaches the next RSU, serving as the neighboring RSU in the road segment. Upon reception of the Q-RDREQ packet, the neighboring RSU transmits the Q-RDREP packet backward through the vehicles, ultimately reaching the RSU that initially broadcasted the Q-RDREQ packet. This bidirectional transmission ensures the efficient narrowing of the broadcast scope, expediting the route repair process, and facilitating the determination of the optimal next node for relaying in V2V ad-hoc routing.

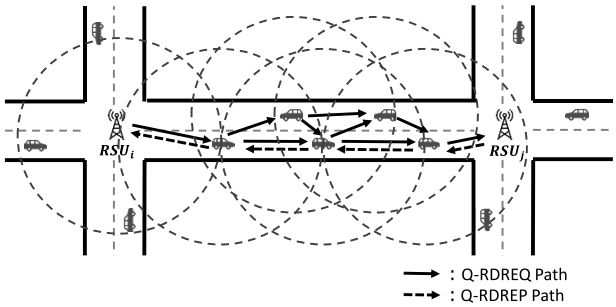

FIGURE 3. RRAH routing mechanism.

1) Q-ROUTE DISCOVERY REQUEST AND REWARD FUNCTION

An RSU selects the neighboring RSU based on its Q-table when data packets arrive and broadcasts the Q-RDREQ packet to nodes in the direction of the selected neighboring RSU if it lacks a fresh route to the selected RSU, with freshness determined by the sequence number (SEQ) of existing route information. The structure of a Q-RDREQ packet is illustrated in Fig. 4(a), where the sending node includes information about the forwarding vehicle, such as its ID, location, and speed. The sending RSU represents the RSU ID and location of the RSU sending the Q-RDREQ packet. The receiving RSU is the next RSU where the Q-RDREQ packet should finally arrive, which is determined by the sending RSU. The destination RSU is the ID and location of the RSU to which the data must arrive finally. $SEQ_{sendingRSU}$ is the sequence number of the RSU generating the Q-RDREQ. $SEQ_{receivingRSU}$ is the sequence number that the sending RSU knows and sends to the receiving RSU. If the sending RSU does not have any sequence information

about the receiving RSU, $SEQ_{receivingRSU}$ is set to zero. The timestamp stores the initial time at which the packet was sent. The remaining four items are evaluations of the route: *TotalDelay*, *TotalHops*, $\min LT$, and $\min LQ$.

TotalDelay represents the cumulative time taken for the Q-RDREQ packet to be transmitted from the sending RSU to the packet receiving vehicle or RSU, denoted as $delay_{RSU_i \to v}$ and derived as in (1).

$$delay_{RSU_i \to v} = delay_{RSU_i \to v_p} + delay_{v_p \to v} \tag{1}$$

TotalHops indicate the number of intermediate vehicles traversed by the Q-RDREQ packet from sending the RSU to the current vehicle or the next RSU. It increases by 1 each time a vehicle receives a Q-RDREQ packet. The number of hops $hop_{RSU_i \to v}$ from RSU_i to the current vehicle v is defined by the sum of $hop_{RSU_i \to v_p}$ and 1 as $hop_{RSU_i \to v} = hop_{RSU_i \to v_p} + 1$.

The minimum link lifetime ($\min LT$) represents the estimated minimum expiration time of communication links among all the links on a path on which vehicles receive the Q-RDREQ packet. In general, when two vehicles travel at similar speeds and in the same direction, they can maintain stable communication, resulting in a long link lifetime. If the distance between the two vehicles is greater than the maximum communication distance (d^{tr}), the communication link between them will be disconnected. The link disconnection condition is expressed as in (2)

$$d_{v_p \to v}\left(t + LT_{v_p \to v}\right) \geq d^{tr} \tag{2}$$

where $d_{v_p \to v}\left(t + LT_{v_p \to v}\right)$ represents the geometric distance between vehicle v_p and v at time $t + LT_{v_p \to v}$. $LT_{v_p \to v}$ is the link lifetime between vehicles v_p and v. The lifetime $LT_{v_p \to v}$ is derived by solving (3).

$$\begin{aligned} & d_{v_p \to v}\left(t + LT_{v_p \to v}\right)^2 \\ & = \left[\left(x_v + V_{x_v} LT_{v_p \to v}\right) - \left(x_{v_p} + V_{x_{v_p}} LT_{v_p \to v}\right)\right]^2 \\ & \quad + \left[\left(y_v + V_{y_v} LT_{v_p \to v}\right) - \left(y_{v_p} + V_{y_{v_p}} LT_{v_p \to v}\right)\right]^2 = d^{tr} \end{aligned} \tag{3}$$

where V_{x_v} and $V_{x_{v_p}}$ denote the velocities along the x-axis for vehicles v and v_p, respectively. V_{y_v} and $V_{y_{v_p}}$ represent the velocities along the y-axis for vehicles v and v_p, respectively. x_v and x_{v_p} correspond to the x-axis coordinates of vehicles v and v_p, respectively. y_v and y_{v_p} correspond to the y-axis coordinates of vehicles v and v_p, respectively. The maximum communication distance d^{tr}, is computed using the path loss model incorporating log-normal shadowing, and is expressed as

$$P^{rcv}_{v_p \to v} = P_{tx} + 10log_{10}K - 10\beta log_{10}\left(\frac{d^{tr}}{d_0}\right) + X_\sigma \tag{4}$$

with

$$K = \frac{G_t G_r \lambda^2}{(4\pi)^2}$$

where $P^{rcv}_{v_p \to v}$ is the received power of vehicle v from previous node v_p, P_{tx} is the transmission power, β is the path loss exponent, d_0 is the reference distance (equal to 1 m), G_t and G_r are antenna gains for transmitter and receiver, respectively, and λ is the wavelength. X_σ represents a zero-mean Gaussian random variable with a variance of σ^2, commonly referred to as the shadowing variance. We obtain the minimum link lifetime $minLT_{RSU_i \to v}$ from RSU_i to vehicle v which is defined as in

$$minLT_{RSU_i \to v} = \min[minLT_{RSU_i \to v_p}, LT_{v_p \to v}] \tag{5}$$

where $minLT_{RSU_i \to v_p}$ the minimum link lifetime from RSU_i to the previous vehicle v_p, which is included in the Q-REREQ packet sent from the vehicle v_p.

The minimum link quality ($\min LQ$) represents the minimum receive strength among all the links on a path when vehicles receive the Q-RDREQ packet. In general, when two vehicles are closer together, a higher link quality is observed during communication. Link quality between v_p and v ($LQ_{v_p \to v}$) is estimated at vehicle v. The minimum link quality $minLQ_{RSU_i \to v}$ from RSU RSU_i to vehicle v is defined as in

$$minLQ_{RSU_i \to v} = \min\left[minLQ_{RSU_i \to v_p}, LQ_{v_p \to v}\right] \tag{6}$$

where $minLQ_{RSU_i \to v_p}$ is the minimum link quality from RSU_i to the previous vehicle v_p, which is included in the Q-REREQ packet sent from the vehicle v_p.

When the vehicle v received the Q-RDREQ packet from the previous vehicle, it can derive the route reward $r_{RSU_i \to v}$ from RSU_i to the vehicle v using the computed total delay, total hops, the minimum link lifetime, and the minimum link quality as expressed in

$$\begin{aligned} r_{RSU_i \to v} = & -\omega_1 \frac{delay_{RSU_i \to v}}{delay^{norm}} - \omega_2 \frac{hop_{RSU_i \to v}}{hop^{norm}} \\ & + \omega_3 \frac{minLT_{RSU_i \to v}}{minLT^{norm}} + \omega_4 \frac{minLQ_{RSU_i \to v}}{minLQ^{norm}} \end{aligned} \tag{7}$$

with

$$\sum_{n=1}^{4} \omega_n = 1$$

where ω_1, ω_2, ω_3, and ω_4 are the weight parameters of each evaluation; $delay^{norm}$, hop^{norm}, $minLT^{norm}$, *and* $minLQ^{norm}$ represent the predefined normalization values of total delay, total hops, the minimum link lifetime, and the minimum link quality, respectively.

Whenever vehicle v receives a Q-RDREQ packet, it calculates the reward value $r_{RSU_i \to v}$ using (7) and stores the previous vehicle v_p and the calculated reward value. If vehicle v receives multiple Q-RDREQ packets from the same RSU and sends identical RSU SEQ numbers to multiple neighboring vehicles, it saves the vehicle with the highest reward value. After calculating the reward, vehicle v constructs a backward table for the reverse path to RSU_i to which it broadcasts the packet, as shown in Fig. 4(b). The backward table consists of the source and destination of the

reverse path for the Q-RDREQ packet, sequence number of the sending RSU that sends the Q-RDREQ, route reward from the sending RSU to the vehicle, and next node to deliver the packet backward, which is the previous vehicle.

When Q-RDREQ reaches the next RSU_j, the R2R route reward between RSU_i and RSU_j is defined as

$$r_{RSU_i \to RSU_j} = \max_{\forall v \in SNV\_RSU_j} [r_{RSU_i \to RSU_j}(v)] \quad (8)$$

where SNV_RSU_j is the set of neighboring vehicles that transmit the Q-RDREQ packets received by RSU_j. $r_{RSU_i \to RSU_j}(v)$ is the computed reward based on the Q-RDREQ packet transmitted by vehicle v. $r_{RSU_i \to RSU_j}$ represents the maximum reward through all the possible paths from RSU_i to RSU_j. This value is utilized as a reward value in DRL for R2R routing to simultaneously provide road traffic and in-time situational awareness.

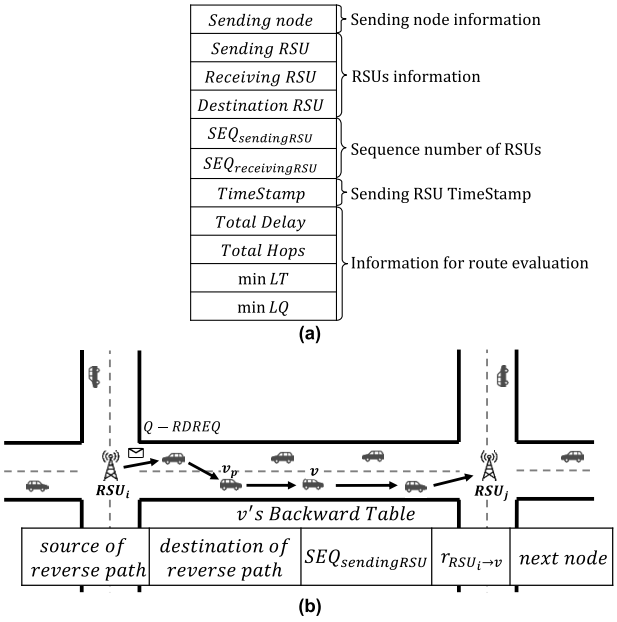

FIGURE 4. (a) Q-RDREQ structure, and (b) backward table update process.

2) Q-ROUTE DISCOVERY REPLY

When the next RSU receives a Q-RDREQ packet, it generates a Q-RDREP packet in the reverse direction of the discovered path. The proposed Q-RDREP packet structure is illustrated in Fig. 5(a). The sending node field contains information regarding the node that transmits the Q-RDREP packet, including its ID, location, and speed. The receiving node field includes information regarding the node that receives the Q-RDREP packet, such as its ID, location, and speed. The sending and receiving RSU fields provide information about the RSUs that send and receive Q-RDREP packets. $SEQ_{sendingRSU}$ is the sequence number of the RSU that sends the Q-RDREP. $SEQ_{receivingRSU}$ is the sequence number that sends the RSU or the sending node maintained for the receiving RSU. The destination RSU indicates the destination of the data packets. The total route reward is obtained

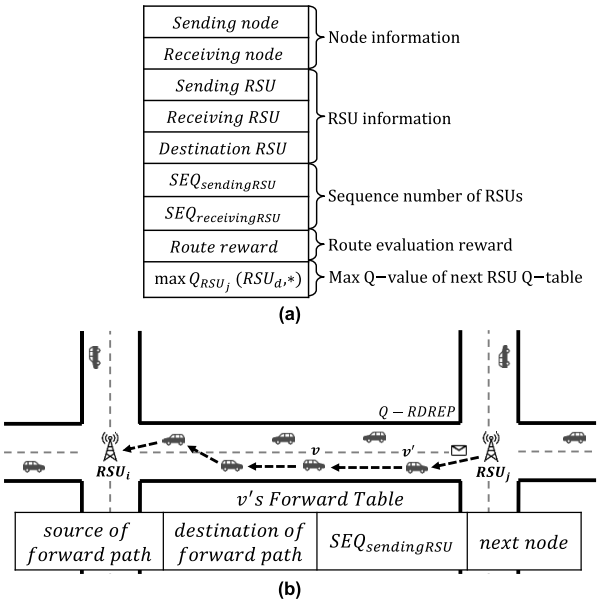

FIGURE 5. (a) Q-RDREP structure, and (b) forward table update process.

from the Q-RDREQ packet. $maxQ_{RSU_j}(RSU_d, *)$ provides information regarding the maximum Q-value in the Q-table of RSU_j corresponding to destination RSU RSU_d. Whenever the vehicles receive a Q-RDREP packet, they construct a forward table to designate the forward path to RSU_j. In Fig. 5(b), the forward table of a vehicle stores the source and destinations of the forward path for the Q-RDREP packet, the sequence number of the RSU that sends the Q-RDREP, and the next node to deliver the packet in a forward manner.

C. DISTRIBUTED Q-LEARNING-BASED RSU-ASSISTED HYBRID ROAD-AWARE (RHRA-DRL) ROUTING ALGORITHM

We propose a distributed Q-learning-based road-aware RSU-to-RSU routing scheme, referred to as DQRA, which strategically positions multiple RSUs as agents at intersections to enhance the R2R routing efficiency and increase the awareness of road traffic and conditions. The RSU that receives the data packet determines the destination RSU_d, where the final destination vehicle is located on the road segment, using the location service. To deliver the data packet to RSU_d, the RSU selects the optimal RSU from the surrounding road segments through distributed Q-learning. Once the road segment on which the next RSU is located has been determined, data packets are delivered through the forward tables of vehicles on the road between the two RSUs. When the next RSU receives a data packet, it returns an ACK packet to the data-transmitting RSU to update its route knowledge to the destination RSU. The proposed RRAH and DQRA integrate the RHRA-DRL routing algorithm. At the end of this subsection, example scenarios are presented to illustrate route discovery and Q-table update mechanisms.

1) Q-LEARNING SCHEME

Q-learning is a model-free RL algorithm that learns without the use of a predefined environmental model. In Q-learning, the state space comprises descriptions of the current situation or configuration of the environment, whereas the action space represents the possible decisions or moves available in a given state. A Q-table is a structured representation of these Q-values, where each row corresponds to a state, each column corresponds to an action, and the entries indicate the associated Q-values. A Q-table serves as a data structure that conveys the value of a particular action in a specific state. The Q-learning algorithm employs temporal difference learning to update Q-values. Through this iterative process, the agent refines its understanding of the environment by adjusting the Q-values based on the observed rewards and estimated future rewards. The Q-table update is expressed as follows:

$$Q(s_t, a_t) \leftarrow (1-\alpha)\, Q(s_t, a_t) + \alpha(r + \gamma \max_{\forall a_{t+1}} Q(s_{t+1}, a_{t+1})) \tag{9}$$

where $Q(s_t, a_t)$ represents the Q-value for taking action a_t in state s_t at time t, and r denotes the reward obtained when the agent takes action a_t. The parameter α signifies the learning rate (ranging from 0 to 1), determining the weight of the new information compared to the existing Q-value, and γ serves as the discount factor (ranging from 0 to 1), balancing immediate and future rewards. The term $\max_{\forall a_{t+1}} Q(s_{t+1}, a_{t+1})$ represents the maximum Q-value for the next state at time $t+1$ among all possible actions at $t+1$.

During the Q-learning process, an agent explores the environment, performs actions, and receives rewards. Through this iterative process, the Q-table gradually converges to values that approximate the optimal Q-values. This convergence leads to a policy that guides the agent to make optimal decisions in different states. When selecting the next action, the Q-learning algorithm employs a decaying epsilon-greedy strategy to balance the exploration and exploitation. The parameter ε (ranging from 0 to 1), gradually decreases with increasing episodes as in

$$\varepsilon = \varepsilon - \varepsilon_{decay} \tag{10}$$

where ε starts as a hyperparameter with an initial value (ε_{max}) and decreases by the specified value (ε_{decay}) for each episode. This reduction continues until ε reaches the predefined minimal value (ε_{min}), rather than reducing it all the way to 0. The decision to avoid reducing ε to 0 is deliberate to prevent purely greedy choices from persisting, which could lead to consistently selecting the initially specified path. In real-world traffic scenarios, while initially favoring a specific route may seem beneficial, the dynamic movement of vehicles on the road may later make that path less optimal. Therefore, to encourage exploration with a certain probability, ε is decreased until it reaches ε_{min}.

2) DISTRIBUTED Q-LEARNING-BASED RSU-ASSISTED (DQRA) ROUTING

In distributed Q-learning, the fundamental concepts of Q-tables and learning remain similar to those of traditional Q-learning; however, the key difference lies in the distribution of the Q-table and learning process across multiple agents. The goal of distributed Q-learning is for each agent to learn an optimal policy for its local environment, while leveraging communication to benefit from the collective knowledge of the entire distributed system. This is particularly useful when a centralized approach is impractical owing to scalability, communication constraints, or the distributed nature of the environment.

In the proposed method, each RSU acts as an individual agent with a dedicated Q-table. In the proposed distributed RL, the set of states for agent RSU_i is defined by all destination RSUs as $S_i \triangleq \{RSU_1, \cdots, RSU_I\}$, where I is the total number of RSUs. Because each RSU can be a destination for a data packet, the set of states includes all the RSUs installed at intersections. The set of actions in the Q-table of the RSU agent is defined as all neighboring RSUs, denoted as $A_i \triangleq \left\{RSU_i^1, \cdots, RSU_i^K\right\}$, where K is the total number of neighboring RSUs of RSU_i.

The reward function for DQRA is defined using the segment reward, as in the RRAH routing process. The R2R reward is defined as follows:

$$r_{R2R} = \begin{cases} r_{RSU_i \rightarrow RSU_j}, & if\ RSU_j \neq RSU_d \\ r_{RSU_i \rightarrow RSU_j} + r_d, & if\ RSU_j = RSU_d \end{cases} \tag{11}$$

where the reward value is divided into two parts, $r_{RSU_i \rightarrow RSU_j}$ represents the reward obtained between RSU_i and the next RSU_j. If the next RSU_j is not the destination RSU_d, the reward $r_{RSU_i \rightarrow RSU_j}$ is assigned. However, if the next RSU_j is the ultimate destination RSU_d, a predefined terminal reward r_d is given. The Q-table update in (12) is illustrated in Fig. 6.

$$\begin{aligned} &Q_{RSU_i}\left(RSU_d, RSU_j\right) \\ &= (1-\alpha)\, Q_{RSU_i}\left(RSU_d, RSU_j\right) \\ &\quad + \alpha[r_{R2R} + \gamma \max_{\forall RSU_j^k \epsilon SNR\_RSU_j} Q_{RSU_j}(RSU_d, RSU_j^k)] \end{aligned} \tag{12}$$

where $Q_{RSU_i}\left(RSU_d, RSU_j\right)$ represent the state-action Q-values for state RSU_d and action RSU_j of the RSU_i Q-table, respectively. SNR_RSU_j is the set of neighboring RSUs of RSU_j. This value is then scaled by the learning rate α and added to the product of the discount factorγ and the maximum Q-value from the neighboring RSU's Q-table, denoted as $\max_{\forall RSU_j^k \epsilon SNR\_RSU_j} Q_{RSU_j}(RSU_d, RSU_j^k)]$. The result is added to the previous Q-value $Q_{RSU_i}\left(RSU_d, RSU_j^i\right)$ after applying a factor $(1-\alpha)$.

When each RSU receives a data packet, it determines the optimum next RSU to reach the destination by using the current Q-table. With $(1-\varepsilon)$ probability, it selects the action (i.e., the next RSU) that has the largest Q-value for the given

destination RSU, and with ε probability it selects a random action. Once the next RSU is determined, the vehicle route on the road segment is determined using the proposed RRAH process. For each vehicle, data packets are forwarded using the forward table obtained, whereas ACK packets are sent back using the backward table.

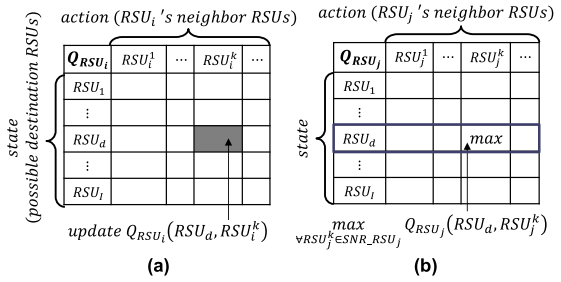

FIGURE 6. Q-value update with (a) the current Q-table, and (b) the next Q-table.

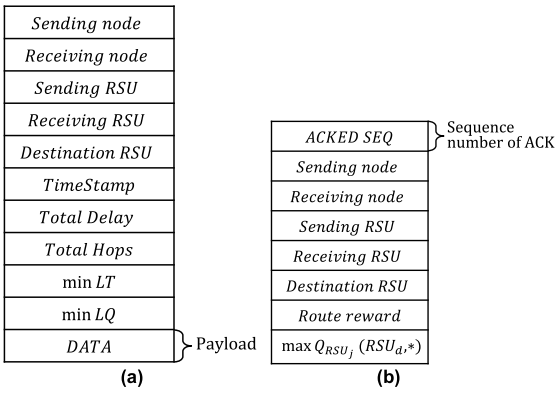

FIGURE 7. Structures of (a) data packet, and (b) ACK packet.

The structure of a data packet is illustrated in Fig. 7(a). The sending node field contains information regarding the node that sends the data packet, including its ID, location, and speed. The receiving node field represents information regarding the node that receives the data packet. This information is related to the details of the next node, which are listed in the forward table. The sending RSU field represents the ID and location of the RSU sending the data packet. The receiving RSU was the RSU selected from the sending RSU Q-table. The destination RSU is the RSU ID, which is the final data destination. A timestamp stores the time at which a data packet is sent from the sending RSU. Four additional metric fields (*Total Delay*, *Total Hops*, min LT and min LQ) are updated for each vehicle on the path using (1)–(6). The remainder of the packet is the payload. Finally, when the data packet from RSU_i is delivered to the next RSU intended (RSU_j), RSU_j computes the route reward $r_{RSU_i \to RSU_j}$ using four metric fields in the data packet using (7). The ACK packet structure is shown in Fig. 7(b). The ACKED SEQ stores the ACK sequence numbers. The receiving node is identified using a backward table to receive the RSU. The sending and receiving RSU indicate the ACK packet-initiated RSU and the destination RSU of the ACK packet, respectively. The route reward is set by the ACK packet, which sends the RSU to the computed route reward for the data packet. The max $Q_{RSU_j}(RSU_d, *)$ includes the maximum Q-value among all actions of the RSU that sends the ACK packet for state RSU_d. This value is used to update the Q-table of the RSU that receives the ACK packet, as shown in (12).

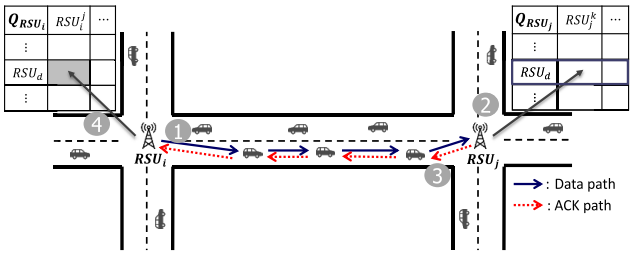

FIGURE 8. DQRA routing process.

In Fig. 8, we illustrate the data routing process for DQRA. When RSU_i receives a Q-RDREP packet from RSU_j, RSU_i sends a data packet to RSU_j as in step 1. Upon receiving the data packet, RSU_j checks its Q-table for the maximum Q-value with the corresponding state and computes the route reward for the data, as in step 2. Then, the ACK packets are sent back to RSU_i in the reverse path, as in step 3. Once RSU_i receives the ACK packet, it updates its Q-table, as in step 4.

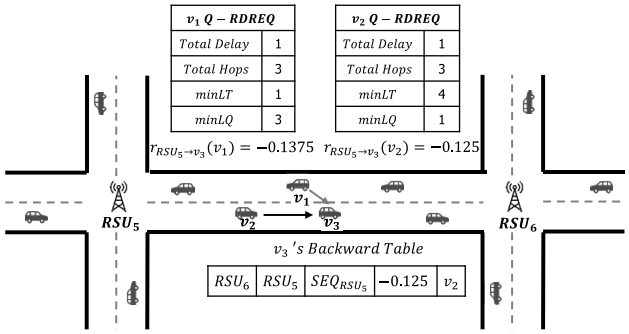

FIGURE 9. Example of backward table update.

3) EXAMPLE SCENARIO

Fig. 9 provides an example illustrating the update of the backward table when broadcasting Q-RDREQ packets to neighboring nodes. Vehicles v_1 and v_2 deliver the Q-RDREQ packet with the same timestamp as vehicle v_3. In this scenario, v_3 calculates rewards using the total delay, total hops, *minLT*, and *minLQ* among v_1 and v_2 in the Q-RDREQ packets. Assuming that the weight parameters are 0.25, $delay^{norm}$, hop^{norm}, $minLT^{norn}$, *and* $minLQ^{norn}$ are 4, 6, 20, and 20, respectively. The reward from RSU_5 to v_3 through v_1 was -0.1375 and the reward from RSU_5 to v_3 through v_2 was -0.125. Because the path through v_2 had a larger reward,v_2 was selected as the next node in the backward table.

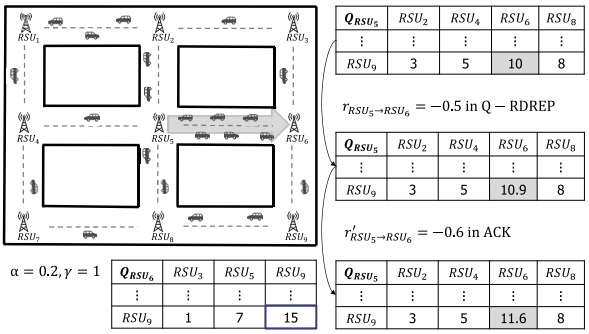

FIGURE 10. Example of Q-table update.

In this scenario, the Q-table updates for the proposed method are shown in Fig. 10. If RSU_5 receives a data packet with RSU_9 as the final destination RSU, it refers to its Q-table (Q_{RSU_5}) for state RSU_9. Among the state-action Q-values, the entry $Q_{RSU_5}(RSU_9, RSU_6)$ exhibit the maximum value, which is 10, RSU_6 is chosen as the next RSU in the routing path. Subsequently, RSU_5 broadcasts the Q-RDREQ to RSU_6. Upon receiving Q-RDREQ, RSU_6 processes the information, noting a route reward of -0.5 and consulting its Q-table to find the maximum value for state RSU_9. In this example, it is $Q_{RSU_6}(RSU_9, RSU_9) = 15$. This information is conveyed in the Q-RDREP until it reaches RSU_5. Assuming a learning rate of 0.2, discount factor of 1, and utilizing (12), the Q-values $Q_{RSU_5}(RSU_9, RSU_6)$ are updated to 10.9. Subsequently, the data packet is successfully delivered using the forward tables of the relay vehicles and an ACK is sent back to the sending RSU using the backward tables. The ACK includes a new route reward of -0.6 and the latest maximum Q-value at the next RSU. The Q-values of $Q_{RSU_5}(RSU_9, RSU_6)$ were updated again, resulting in a new value of 11.6. The Q-table updates exclusively upon receiving Q-RREP or ACK packets from the next RSU. The number of vehicles within a road segment has no impact on the update count and maintenance of the Q-table. However, as the demand increases with more vehicles, the Q-table update count rises, contributing to faster convergence.

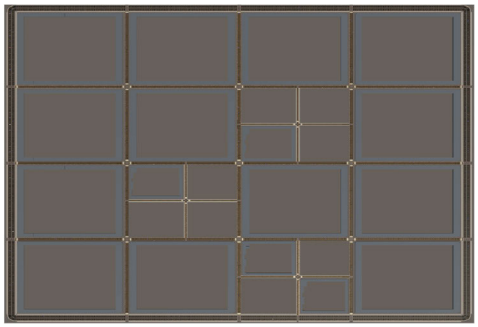

FIGURE 11. Simulation topology in unity 3D environment.

IV. SIMULATION RESULTS

In this study, the simulation experiments are conducted in a Unity 3D environment. Unity is a powerful and widely used game development engine that allows developers to create two-dimensional (2D) and three-dimensional (3D) games across multiple platforms. It provides a user-friendly interface and vast asset stores, and supports the building of a real-world road and traffic environment. The simulation topology for this environment is shown in Fig. 11. We deployed 25 RSUs at each intersection and systematically positioned RSU1 at the top-left, RSU5 at the top-right, RSU21 at the bottom-left, and RSU25 at the bottom-right. Each road-segment length was set to 500 m, and the number of lanes was four for the interior roads and eight for the border roads to increase the capacity of the topology. A total of 900 vehicles were randomly moved on roads at speeds of 30, 40, and 50 km/h. Once each vehicle passes an intersection, its speed is reset using three speed options. The experimental parameters used for the simulation are listed in Table 1. We set the starting RSU randomly and the destination RSUs as RSU21, RSU22, RSU23, RSU24, and RSU25.

TABLE 1. Simulation experimental parameters.

Parameter	Value	Parameter	Value
Number of vehicles	900	Epsilon min and max value($\varepsilon_{max}, \varepsilon_{min}$)	0.5, 0.1
Number of RSUs	25	Epsilon decay value (ε_{decay})	0.01
Vehicular speed	30 km/h, 40 km/h, 50 km/h	Minimum decodable power	-92 dBm
Learning rate (α)	0.2	Transmission power (P_{tx})	15 dBm
Discount factor (γ)	0.9	Antenna gains (G_t, G_r)	1
Terminal reward (r_d)	200	Frequency	5.9 GHz
Weight parameters ($\omega_1, \omega_2, \omega_3, \omega_4$)	0.25	Path loss exponent (β)	3
Data packet size	512 bytes	Shadowing variance (σ)	2

In our simulation experiments, we conducted learning using three learning rates (0.1, 0.2, and 0.3) to identify the most suitable rate for our proposed approach. The results are shown in Fig. 12, where the x-axis represents the episodes and the y-axis represents the maximum Q-value at RSU1. An episode is considered complete when the data packets successfully reach the destination RSUs 21, 22, 23, 24, and 25 sequentially from the source RSU. From the perspective of the source RSU, a higher Q-value is obtained when the number of intermediate RSUs decreases from the source RSU to the destination RSU. As shown in Fig. 12, a higher learning rate results in a faster convergence speed. Specifically, when the learning rate was 0.1, convergence occurred at around 120 episodes; for a learning rate of 0.2, convergence occurred at around 60 episodes; and for a learning rate of 0.3, convergence occurred at around 40 episodes. However, when the learning rate increased to 0.3, larger fluctuations in the Q-values were observed compared to the other cases. A learning rate of 0.2 is applied to the following experiments.

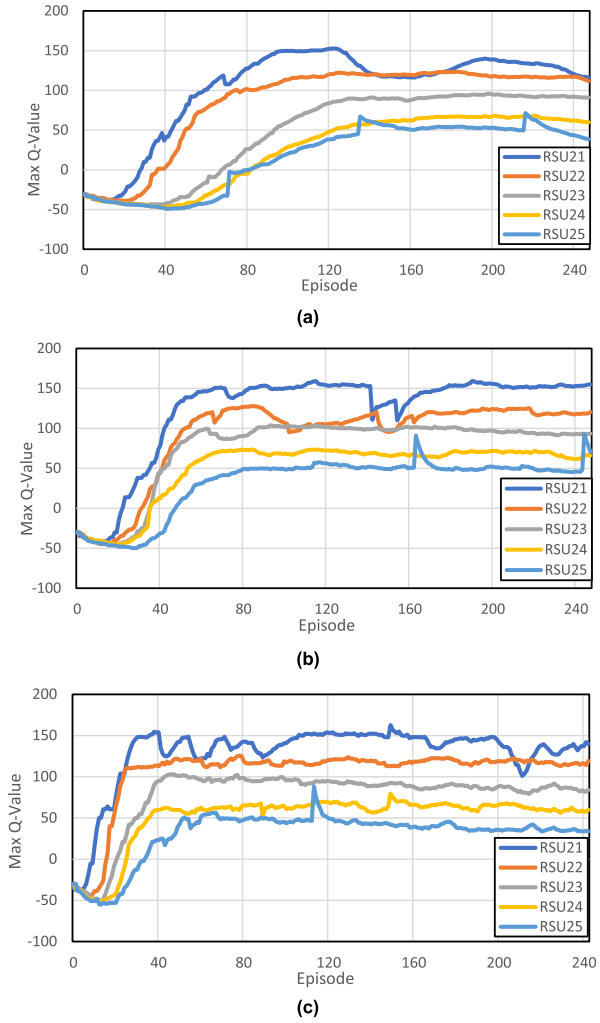

FIGURE 12. Learning rates comparison of (a) 0.1, (b) 0.2, and (c) 0.3.

We examined the convergence characteristics of different performance metrics, as shown in Fig. 13. In Fig. 13(a), the horizontal axis represents the episodes, whereas the vertical axis depicts the sum of the packet delays from the source RSU (RSU1) to the destination RSUs. Initially, with limited learning progress, a notable increase in delay values was observed. Nevertheless, as learning advanced, the delay values gradually decreased, ultimately showing a significant reduction. As shown in Figs. 13(b) and 13(c), as the number of episodes increased, the required hops gradually decreased, and the cumulative minimum link quality from the source RSU to the destination RSUs increased. Fig. 13(d) shows the cumulative rewards. As is evident from the figure, the cumulative reward value increased rapidly in the early episodes but exhibited relatively large fluctuations. However, after approximately 60 episodes, the reward value stabilized and consistently converged.

In Fig. 14, a comparison is made between the proposed method and AODV-based approach regarding the average minimum link lifetime, which is calculated for the established routes from the source RSU and five destination RSUs.

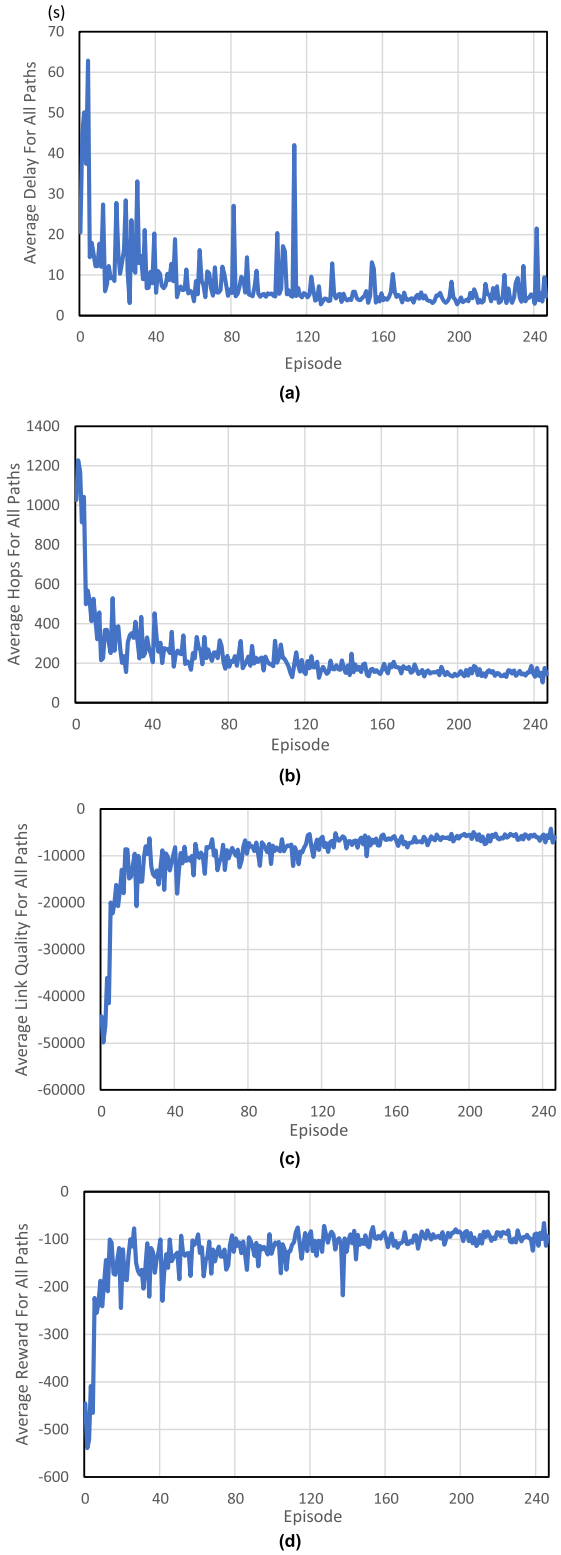

FIGURE 13. Learning convergences (a) average packet delay, (b) average packet hops, (c) average minimum link quality, and (d) average reward for all paths from source to destinations.

As the episode count increased, the proposed method initially exhibited a relatively good link lifetime owing to the uniform random distribution of vehicles. However, as vehicles move

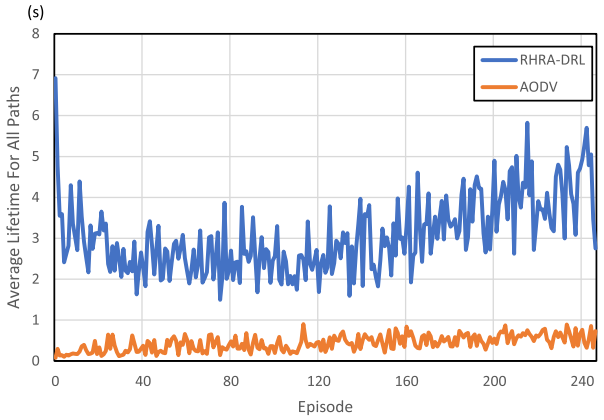

FIGURE 14. Average minimum link lifetime comparison.

on the road in different directions and at different speeds, the arrangement of vehicles gradually becomes uneven, leading to a decrease in the lifetime of the paths compared with the initial stages. However, as the episodes progressed and the Q-table converged, the link lifetime of the proposed method increased. In contrast, the AODV-based approach requires frequent path disconnections and re-establishments because of the end-to-end path setup through broadcasting across the entire IoV network, rather than a distributed path setup at the road-segment level. Therefore, the average link lifetime of AODV was consistently lower than that of the proposed RHRA-DRL.

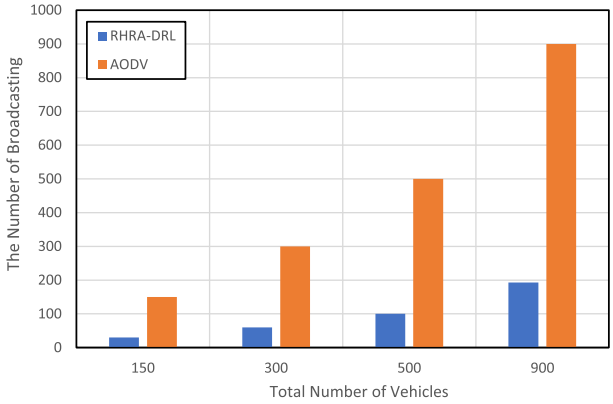

FIGURE 15. Number of broadcasts comparison.

Frequent broadcasts carry the risk of consuming a substantial portion of the bandwidth, leading to increased collision risks and, consequently, deteriorating the overall network performance. Fig. 15 depicts the correlation between the number of vehicles and total number of broadcasts. The AODV approach exhibited a sharp increase in the number of broadcasts as the number of vehicles in the network increased. Conversely, the proposed method employs segment-specific broadcasting, which restricts broadcasts to a specific segment when the lifetime of the path expires. As a result, AODV incurs a significantly higher broadcasting overhead than the proposed method.

V. CONCLUSION

To achieve fast and reliable data delivery in urban vehicular networks, it is essential to adapt quickly to dynamic road conditions and continuously derive optimal routes in real-time by considering various performance metrics. To address these challenges, this study proposes an RSU-assisted hybrid road-aware routing algorithm, RHRA-DRL, aimed at minimizing the broadcast overhead while efficiently determining the optimal routing paths. In this study, we propose a multihop road-segment reward-based adhoc (RRAH) routing algorithm to adaptively respond to rapidly changing vehicle topologies within the road segment between two RSUs installed at an intersection. To establish the optimal V2V path within the segment, rewards are calculated by considering various performance metrics, and the final computed segment reward is directly incorporated into the R2R routing. To determine the RSUs traversed during data transmission from the source RSU to the final destination RSU, we propose a distributed Q-learning-based road-aware (DQRA) routing using a decentralized agent RL approach. Ultimately, by combining the two routings, RHRA-DRL performs more effectively and consistently in path establishment, utilizing a consistent reward system, despite employing different approaches for intra-segment and inter-segment routing. The simulation results demonstrate that the proposed method enhances communication with a longer link lifetime and rapidly establishes and repairs routing paths while reducing overhead when compared to AODV in the context of IoV networks.

ACKNOWLEDGMENT

(Joo-Hyung Park and Qin Yang are co-first authors.)

REFERENCES

[1] S. M. Karim, A. Habbal, S. A. Chaudhry, and A. Irshad, "Architecture, protocols, and security in IoV: Taxonomy, analysis, challenges, and solutions," *Secur. Commun. Netw.*, vol. 2022, pp. 1–19, Oct. 2022, doi: 10.1155/2022/1131479.

[2] S. Bhuvaneshwari, G. Divya, K. B. Kirithika, and S. Nithya, "A survey on vehicular ad-hoc network," *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.*, vol. 2, no. 10, pp. 4993–5000, 2013.

[3] B. Paul, M. Ibrahim, and M. A. N. Bikas, "VANET routing protocols: Pros and cons," *Int. J. Comput. Appl.*, vol. 20, no. 3, pp. 28–34, Apr. 2011.

[4] A. D. Devangavi and R. Gupta, "Routing protocols in VANET—A survey," in *Proc. Int. Conf. Smart Technol. Smart Nation (SmartTechCon)*, Bengaluru, India, Aug. 2017, pp. 163–167, doi: 10.1109/SmartTechCon.2017.8358362.

[5] G. S. Aujla, R. Chaudhary, N. Kumar, J. J. Rodrigues, and A. Vinel, "Data offloading in 5G-enabled software-defined vehicular networks: A Stackelberg-game-based approach," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 100–108, Aug. 2017, doi: 10.1109/MCOM.2017.1601224.

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

[7] J.-C. Chen, "Dijkstra's shortest path algorithm," *J. Formalized Mathematics*, vol. 15, no. 9, pp. 237–247, 2003.

[8] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, and P. Minet, *Optimized Link State Routing Protocol (OLSR)*, document ⟨inria-00471712⟩, 2003.

[9] T. Kayarga and S. A. Kumar, "A study on various technologies to solve the routing problem in Internet of Vehicles (IoV)," *Wireless Pers. Commun.*, vol. 119, pp. 459–487, Jul. 2021, doi: 10.1007/s11277-021-08220-w.

[10] 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.

[11] T. Kabir, N. Nurain, and Md. H. Kabir, "Pro-AODV (proactive AODV): Simple modifications to AODV for proactively minimizing congestion in VANETs," in *Proc. Int. Conf. Netw. Syst. Secur. (NSysS)*, Dhaka, Bangladesh, Jan. 2015, pp. 1–6, doi: 10.1109/NSysS.2015.7043521.
[12] R. M. Kumar and S. K. Routray, "Ant colony based dynamic source routing for VANET," in *Proc. 2nd Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT)*, Jul. 2016, pp. 279–282.
[13] M. Jerbi, S.-M. Senouci, R. Meraihi, and Y. Ghamri-Doudane, "An improved vehicular ad hoc routing protocol for city environments," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2007, pp. 3972–3979, doi: 10.1109/ICC.2007.654.
[14] S. M. Bilal, A. U. R. Khan, and S. Ali, "Review and performance analysis of position based routing in VANETs," *Wireless Pers. Commun.*, vol. 94, no. 3, pp. 559–578, Jun. 2017, doi: 10.1007/s11277-016-3637-6.
[15] 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, May 2017.
[16] 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.
[17] S. Kannan, G. Dhiman, Y. Natarajan, A. Sharma, S. N. Mohanty, M. Soni, U. Easwaran, H. Ghorbani, A. Asheralieva, and M. Gheisari, "Ubiquitous vehicular ad-hoc network computing using deep neural network with IoT-based bat agents for traffic management," *Electronics*, vol. 10, no. 7, p. 785, Mar. 2021, doi: 10.3390/electronics10070785.
[18] 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.
[19] C. Wu, K. Kumekawa, and T. Kato, "Distributed reinforcement learning approach for vehicular ad hoc networks," *IEICE Trans. Commun.*, vol. E93-B, no. 6, pp. 1431–1442, 2010.
[20] 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, doi: 10.1109/ACCESS.2020.2963850.
[21] A. Lolai, X. Wang, A. Hawbani, F. A. Dharejo, T. Qureshi, M. U. Farooq, M. Mujahid, and A. H. Babar, "Reinforcement learning based on routing with infrastructure nodes for data dissemination in vehicular networks (RRIN)," *Wireless Netw.*, vol. 28, no. 5, pp. 2169–2184, Jul. 2022, doi: 10.1007/s11276-022-02926-w.
[22] G. Sun, Y. Zhang, H. Yu, X. Du, and M. Guizani, "Intersection fog-based distributed routing for V2V communication in urban vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2409–2426, Jun. 2020, doi: 10.1109/TITS.2019.2918255.
[23] 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, doi: 10.1109/TITS.2021.3053958.

JOO-HYUNG PARK received the B.S. degree from the Department of Information and Communication Engineering, Hanshin University, Gyeonggi, South Korea. He is currently pursuing the M.S. degree in electrical and computer engineering with the Multimedia Network Laboratory, Inha University, Incheon, South Korea. His research interests include machine learning, reinforcement learning, vehicular networks, and the Internet of Things.

QIN YANG (Graduate Student Member, IEEE) received the B.E. degree in communication and information engineering from 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 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, where he was involved in communication protocol conformance testing and network design. From 1994 to 1995 and from 2007 to 2008, he was the 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 Department of Electrical and Computer Engineering. His current research interests include cognitive radio network protocols, VANET and FANET protocol designs, AI-based networking algorithms, and the IoT applications.