IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received January 10, 2021, accepted February 7, 2021, date of publication February 10, 2021, date of current version February 18, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3058388*

# Reinforcement Learning-Based Routing Protocols for Vehicular Ad Hoc Networks: A Comparative Survey

**REZOAN AHMED NAZIB AND SANGMAN MOH, (Member, IEEE)**
Department of Computer Engineering, Chosun University, Gwangju 61452, South Korea

Corresponding author: Sangman Moh (smmoh@chosun.ac.kr)

This work was supported in part by the Chosun University, 2020, under Grant K202160030.

**ABSTRACT** Vehicular-ad hoc networks (VANETs) hold great importance because of their potentials in road safety improvement, traffic monitoring, and in-vehicle infotainment services. Due to high mobility, sparse connectivity, road-side obstacles, and shortage of roadside units, the links between the vehicles are subject to frequent disconnections; consequently, routing is crucial. Recently, to achieve more efficient routing, reinforcement learning (RL)-based routing algorithms have been investigated. RL represents a class of artificial intelligence that implements a learning procedure based on previous experiences and provides a better solution for future operations. RL algorithms are more favorable than other optimization techniques owing to their modest usage of memory and computational resources. Because a VANET deals with passenger safety, any kind of flaw is intolerable in VANET routing. Fortunately, RL-based algorithms have the potentials to optimize the different quality-of-service parameters of VANET routing such as bandwidth, end-to-end delay, throughput, control overhead, and packet delivery ratio. However, to the best of the authors' knowledge, surveys on RL-based routing protocols for VANETs have not been conducted. To fulfill this gap in the literature and to provide future research directions, it is necessary to aggregate the scattered works on this topic. This study presents a comparative investigation of RL-based routing protocols, by considering their working procedure, advantages, disadvantages, and applications. They are qualitatively compared in terms of key features, characteristics, optimization criteria, performance evaluation techniques, and implemented RL techniques. Lastly, open issues and research challenges are discussed to make RL-based VANET routing protocols more efficient in the future.

**INDEX TERMS** Vehicular ad hoc network, routing protocol, reinforcement learning, Q-learning, intelligent algorithm, quality-of-service routing, intelligent transportation system.

## I. INTRODUCTION

Vehicular ad hoc networks (VANETs) are among the most investigated topics in the field of mobile ad hoc networks (MANETs). In VANETs, vehicles transmit information in a multihop fashion to deliver data from the source to a destination [1]. VANETs can be used to improve passenger safety, in-vehicle infotainment, blind-spot prevention, traffic maintenance, emergency message propagation, and autonomous driving. Over the last decade, many researchers have attempted to optimize the performance of routing protocols for VANETs [2]. Despite their usefulness, VANETs have limitations and challenges [3], [4]. Routing in a VANET is a challenging task due to high vehicle mobility and dynamic link connectivity.

The connection between vehicles is adversely affected by their fragile link condition. Roads do not follow a common paradigm [5] for urban areas, rural areas, and highway road conditions. Moreover, roadside obstacles create a non-line-of-sight (NLOS) situation, which increases the complexity of routing [6]. Consequently, numerous VANET routing algorithms have been reported in the literature. Popular MANET routing protocols that have been tested for VANET include ad-hoc on-demand distance vector (AODV) routing [7], dynamic source routing (DSR) [8], destination-sequence distance vector (DSDV) routing [9], greedy perimeter stateless routing (GPSR) [10], and link-state routing protocol [11]. These major routing protocols have been further modified

The associate editor coordinating the review of this manuscript and approving it for publication was Asad Waqar Malik.

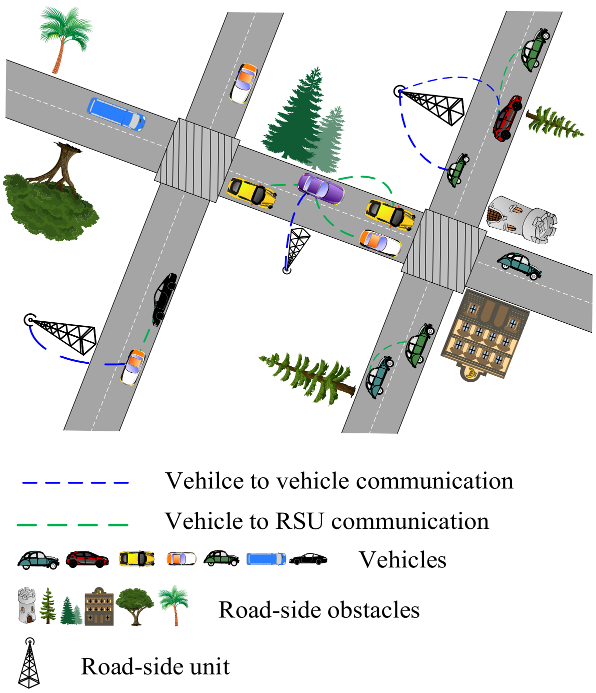
Vehilce to vehicle communication
Vehicle to RSU communication
Vehicles
Road-side obstacles
Road-side unit

**FIGURE 1.** A simplified example of a VANET configuration.

and implemented to improve performance in a VANET environment [12]. Fig. 1 illustrates the basic operation and communication paradigm of VANET architecture. Machines are more capable and efficient than humans in terms of solving problems in a controlled environment.

Machine learning (ML) algorithms can be divided into supervised, unsupervised, and reinforcement learning (RL) categories [13]. These subfields of ML are also used to optimize the different features of the VANET architecture. The prediction of traffic conditions, network traffic estimation, control of traffic lights, vehicle speed suggestions, control of network congestion, assisting in navigation, increasing VANET security, and resource allocation [14]–[16] are examples of these features. ML algorithms are used to improve the performance of routing protocols for VANETs [17]. These algorithms are designed to optimize the various quality-of-service (QoS) parameters of VANET routing algorithms under different circumstances [18].

The RL algorithm is applied to improve the routing algorithm for different ad-hoc network architectures, such as wireless sensor networks [19], VANET [20], flying ad hoc networks [21], and drone ad hoc networks [22]. Due to the constrained environment and current limitations of VANETs, RL algorithms are utilized to improve the routing performance of the VANET architecture. RL algorithms are primarily used to optimize the different QoS parameters such as the end-to-end delay (EED), throughput, packet delivery ratio (PDR), the number of hops (NoH), routing overhead, and security [23], [24].

VANET is the key technology to enable intelligent transportation service in smart cities. Besides entertainment services, VANETs also deal with road safety services. As a result, an error-prone routing protocol for VANETs will raise a serious safety concern, and the aim of VANETs will go in vain. The autonomous vehicles take various decisions based on the information disseminated by other vehicles. In such a case, errorless routing of vehicles' data is a must. Despite putting in a good amount of effort, the routing protocols in VANETs are still far from perfection. RL-based algorithms work based on experiences and only get better with time. These algorithms have the potentials of improving the routing experiences of the VANET environment. However, more study is needed to embed the RL concept successfully into the routing mechanism in VANETs. In this circumstance, a comprehensive review paper can play as a good kick-starter for researchers interested in designing RL-based VANET routing protocols. Nevertheless, to the best of our knowledge, no survey has been conducted on this topic till date. Apart from addressing the research gap in the literature, a survey on RL-based VANET routing is needed to motivate researchers to focus more on intelligent VANET routing protocols.

This research presents the results of a survey on RL-based routing protocols for the VANET architecture. To select the existing protocols, at first, we have focused on whether the research work is an RL-aided VANET routing protocol or not. We have emphasized the protocols, which include all the aspects of a routing protocol such as route discovery, data dissemination, route maintenance, and topology control mechanism. All the added protocols have their unique properties, which are worth investigating for working with RL-based VANET routing algorithms. The papers written on a single point of view such as broadcasting mechanism [25], [26] and aggregation mechanism[27] are excluded. However, the protocols that are extended from other protocols and enhanced with the RL algorithms are included. As there is no other survey done on the topic of "RL-based VANET routing algorithms," we have not restricted the publication time of the researches.

The searching methodology of the existing works includes two phases. First, we have searched public domain search sites and academic databases such as IEEE Xplore, Elsevier, Springer, Sage, Wiley-Blackwell, and so on to find out relevant research works. We have listed the papers with their abstract to ensure the exclusion of duplication. We rigorously searched the web result with relevant keywords, in order to ensure the inclusion of all the RL-based VANET routing algorithms.

The novelty of this research lies in the title of the work. To the best of the authors' knowledge, there exists no survey that focuses on the RL-based VANET routing algorithms. We have repeatedly searched the literature but could not find any other survey paper, which shares the idea of this paper. The qualitative comparison given in this literature is mainly focused on the implementation of RL-techniques in VANET routing, which is also not witnessed in the literature so far. The key aspects of this survey are as follows:

- In total, 26 RL-based VANET routing protocols are surveyed in this report. The investigated routing protocols are divided into hybrid, zone-based, geographic, topology-based, hierarchical-based, and security-based protocols. A taxonomy is included to illustrate the categorization, as shown in Fig. 3.
- Critical analysis of the RL-based VANET routing protocols is presented by emphasizing their working procedure, advantages, disadvantages, and applications.
- A comparison of the routing protocols is performed based on their key features, optimization criteria and techniques, performance evaluation techniques and parameters, and performance metrics and analysis. Given that the only way to evaluate the reliability of the proposed theory is to examine the performance, this report presents an in-depth review of the performance evaluation techniques used in the literature. A thorough discussion and the authors' opinions are also presented in addition to tabular comparisons.
- The composition of the RL algorithms proposed in all the reviewed protocols is compared in tabular format. In-depth analysis and suitable application scenarios for the learning techniques are discussed as well.
- Open research issues and challenges are presented with detailed descriptions, which serve as guidelines for future researchers. Each of the given research issues is discussed concerning the lessons learned, existing issues, and brief recommendations.

The remainder of this report is organized as follows. Section II describes the RL procedure. Section III elaborates on the reviewed protocols with their advantages, disadvantages, and applications. The taxonomy of the routing protocols is also presented. In Section IV, the comparisons of the reviewed protocols are discussed based on optimization criteria, innovative ideas, and performance measurement techniques. In Section V, the implemented RL variants are analyzed and recommendations are addressed. In Section VI, open research issues and challenges are summarized and discussed. Finally, the main conclusions are presented in Section VII.

## II. PRELIMINARIES ON REINFORCEMENT LEARNING

RL is a subclass of ML algorithms, in which an agent perceives knowledge from the surrounding environment and attempts to maximize a reward to reach a goal. RL is applicable to moderately complex and perplexing environments. The agent receives a reward or penalty for every action based on its impact on the environment. The agent learns which action should be performed to maximize the rewards and to avoid penalties. Fig. 2 shows the basic working procedure of the RL mechanism [28]. RL is modeled as a Markov decision process (MDP) problem [29]. An MDP includes a set of environments, states, actions, a probability distribution table of actions, the reward function, and some constraints. The probability of transition can be written as the following equation:

$$P_a\left(s, s'\right) = P_R\left(s_{t+1} = s' | s_t = s, a_t = a\right), \tag{1}$$

where $P_a$ is the probability of transition from state $s$ to $s'$, $P_R$ represents the probability distribution, $s_t$ denotes the state at time $t$, $s_{t+1}$ denotes the state at time $t+1$, and $a_t$ is the action performed at time $t$. A state is the current situation of the environment upon which the agent acts. A single state $s$ belongs to a set of states $S = s_1, s_2, s_3, s_4, \ldots s_n$. Actions belong to the set of actions denoted by $A = \{a_1, a_2, a_3, a_4, \ldots a_n$. The reward can be denoted as $r_a\left(s, s'\right)$. $r_a$ is the reward for performing action $a$ at time $t$. An agent gets the reward as an immediate return for performing an action on a state. The prior condition for applying the RL algorithms is that the environment must be stochastic. RL algorithms are more applicable to environments in which no prior information on the environment is available, and the environment can be simulated either via computer simulations or using test-bed simulations. The only way to accumulate data about the environment is to associate the environment [28]. There are two potential approaches for the RL mechanism that can be pursued by an agent. One approach is to find the value of the state, and the other is to find the value of the action. The policy of the RL algorithms states the outcome of a state and a particular action [30]. Based on a policy, the agent determines the action to be taken for a given state. The policy of a state can be expressed as follows:

$$\pi\left(a, s\right) = P_R\left(a_t = a | s_t = s\right), \tag{2}$$

where $\pi$ is the policy and gives the probability of performing an action $a$ in the state $s$. The state value represents the long-term return for a state after considering the discount factor. This can be expressed using the following formula:

$$R = \sum_{t=0}^{\infty} \gamma^t r_t \tag{3}$$

where $R$ is the long-term return; $\gamma^t$ is the discount factor at time $t$ and indicates the impact of subsequent state values on the computation of the current state value. The value of $\gamma$ differs within the range of 0 to 1 [28]. $r_t$ is the reward at time $t$. The state value function represents the reward for being in a state. The function is expressed as follows:

$$V_\pi\left(s\right) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s\right], \tag{4}$$

where the expected return is denoted by $E$, and $V_\pi$ is the state value function. $s_0$ indicates the initial state. An RL algorithm converges when it finds the optimal policy from all available policies for a given state [28]. The optimality of the RL algorithm can be denoted by the following formula:

$$V^*\left(s\right) = \max_\pi V^\pi\left(s\right) \tag{5}$$

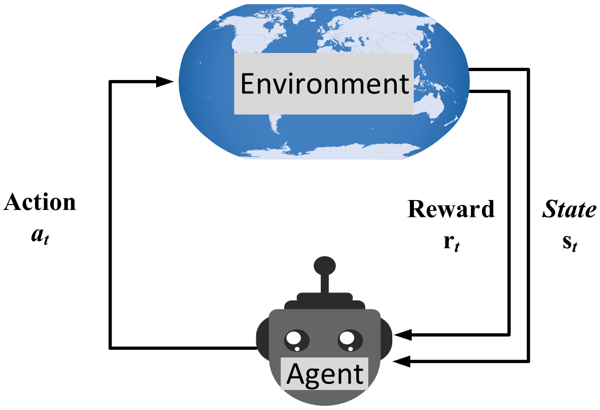

**FIGURE 2.** Basic working procedure of the RL mechanism.

where $V^*$ is the optimal value function for state $s$, which is achieved by applying the optimal policy. $\pi^*$ is called the optimal policy, which is defined by the corresponding state $s$ and the action that returns the highest reward. $V^\pi$ is the state value function for a given policy $\pi$. The RL algorithms can be designed based on the policy iteration function or the value iteration function. Policy iteration methodologies include Monte Carlo [31] and temporal differencing mechanisms [32]. In the Monte Carlo mechanism, the rewards depend on the sampling of the states as much as possible, whereas temporal difference utilizes the value returned by the immediate state only. The Q-learning technique is the easiest and most practiced technique, which falls under the category of value iteration functions.

The RL algorithm can be classified based on the given information. Model-based [33] algorithms use a probability distribution table for every legal action in the environment. These algorithms are not practical because of the increasing number of states and actions. In contrast, model-free [34] algorithms do not have such a distribution table. Instead, these algorithms depend solely on the learning policy. Model-free algorithms adopt a trial-and-error process to improve the quality of the action performed on a certain state.

In total, there are four types of actions that an agent can perform: random action [35], greedy action [36], epsilon greedy action [37], and softmax action [38]. The exploration and exploitation percentage depends on the type of action an agent performs. In greedy action, no exploration is performed, whereas in random action, all the actions are based on exploration. The epsilon greedy method chooses between exploration and exploitation based on a fixed value called epsilon. SoftMax functions reduce the number of explorations with time and increase exploitation. A task can be categorized as an episodic or a continuing task. An episodic task has a terminal state, but a continuing task does not have a terminal state. To fit an RL algorithm, continuing tasks are mostly converted into episodic tasks.

Depending on the policy, RL algorithms can be divided into on-policy [39] and off-policy algorithms [40]. In an on-policy algorithm, the agent learns based on the action, whereas in the off-policy algorithm, the action is taken from another policy that returns the obtained maximum value. Such a policy resembles a greedy action policy.

The quality of a policy is evaluated using a policy evaluation technique in which the state value of the policy is evaluated based on the value of the greedy policy. However, policy improvement enhances and updates the policy, which returns the maximum state value. Some major RL techniques include trust region policy optimization (TRPO) [41], proximal policy optimization (PPO) [42], Q-learning or value iteration method [43], state-action-reward-state-action (SARSA) [44], and deep Q network (DQN) [45] algorithm. However, Q-learning is the most popular RL algorithm in use and practice. There are mainly four types of RL algorithms' variants used in the investigated RL-based VANET routing protocols. They are the Q-learning algorithm, policy hill climbing, SARSA($\lambda$) and deep RL (DRL) algorithm. Further discussion is given in the following subsections, describing the working methodology of the two algorithms.

### A. Q LEARNING

Any process that can be modeled as an MDP model can be solved using the Q learning approach. Q learning is a model-free approach that can act in a stochastic environment [46]. This algorithm interacts with the environment and attempts to maximize the reward from the current state to the goal state. Q learning utilizes a table called the Q table to store the Q values of a state corresponding to an action. Thus, the Q table stores only one value per pair of states and actions. This table can be visualized as a two-dimensional array, wherein the columns can represent the action and the rows can represent the states. Initially, the cells in the tables are filled with 0s [47]. This means that for a particular state and action, a pair has not been explored. The Q value computation is performed using the Bellman equation, which can be expressed as follows:

$$\begin{aligned} &Q_{t+1}\left(s_t, a_t\right) \\ &= Q_t\left(s_t, a_t\right) + \propto \left(r_{t+1} + \gamma \max_a Q_t\left(s_{t+1}, a_t\right) - Q_t\left(s_t, a_t\right)\right), \end{aligned} \tag{6}$$

where $\propto$ is the learning factor, $Q_t$ is the $Q$ value of an action $a$ at time $t$, and $Q_{t+1}$ is the $Q$ value at time $t+1$. $a_t$ is the action $a$ performed at time $t$, $s_t$ denotes the state at time $t$, $S_{t+1}$ represents the state at time $t+1$, and $r_{t+1}$ is the reward at time $t+1$ . The parameter $\propto$ varies between 0 and 1. The higher the value $\propto$, the lesser the time required for the algorithm to converge. However, the likelihood of premature convergence increases. The lower the parameter $\propto$, the more the time required by the algorithm for convergence [47].

### B. SARSA ($\lambda$)

SARSA is an RL algorithm and it stands for state, action, reward, state, and action. This is represented with $(S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1})$. It is an on-policy algorithm. Q-learning can learn only one step at a time whereas the

SARSA can relate its own experience with other state experience, by following the same policy. Q-value update is done in SARSA based on the following equation

$$\begin{aligned} &Q_{t+1}\left(s_t, a_t\right) \\ &\quad = Q_t\left(s_t, a_t\right) + \propto \left(r_{t+1} + \gamma Q_t\left(s_{t+1}, a_{t+1}\right) - Q_t\left(s_t, a_t\right)\right). \end{aligned} \tag{7}$$

In Q-learning, when we compute the value of an action, we don't compute the value of the next states. We only take the maximum available value on the next state and update the Q-value of the current state. SARSA is considered as less optimistic in comparison to the Q-learning procedure, as SARSA does not always take the best available value. For $n$-step look-ahead policy, the SARSA($\lambda$) method is used. SARSA($\lambda$) algorithm uses an additional data structure called the eligibility trace. The eligibility trace is similar to the Q-table data structure. The eligibility trace records the whole path to the destination or looks ahead limitation. For the most recent state, the eligibility is assumed as 1. In every time step, the eligibility reduces $\lambda$ amount. SARSA($\lambda$) can be shown as the connection between the Monte-Carlo method and the temporal difference mechanism. SARSA($\lambda$) is simply the eligibility trace enabled version of the SARSA algorithm. The traces of all state-action pairs are stored inside the eligibility trace. Traces can be of three types and they are: accumulating, replace, or dutch. The main drawback of the SARSA($\lambda$) algorithm is known as the temporal credit assignment problem. This is a problem that indicates the reward assigning issue when multiple states are being considered.

### C. DEEP REINFORCEMENT LEARNING (DRL)

Deep reinforcement learning (DRL) is an improved version of RL which showed great achievements in different research works [48] by combining RL and deep learning [49]. Q-learning has some limitations in maintaining Q-table values. When the state and action spaces become large, the Q-table becomes intractably large. As the agent has to traverse all the possible states, the algorithm may not reach convergence. In DRL, a class of deep neural networks is used known as deep Q network (DQN) for approximating the Q values [50]. DRL takes the advantage of deep learning for taking the raw sensory data as input from the observed environment, and then return output based on the approximation. Unlike RL, DRL uses a replay memory to store the results. In the replay memory, all the experiences of the DRL agent are kept as a tuple in the form of $\{s_t, a_t, r_t, s_{t+1}\}$. Here, $a_t$ is the action taken in state $s_t$ at time $t$ and $r_t$ is the reward DRL agent received upon the action then passed to the next state $s_{t+1}$. From the replay memory, a mini-batch is randomly chosen to train the DQN. The size of the mini-batch has an impact on the performance of the algorithm, which needs to be chosen carefully [51]. The weights of the DQN is updated in every iteration. In order to stabilize the learning process of DQN, an additional neural network called the target network can be used. In that case, the DQN can update the weights after several time periods which reduces the correlations between the target and estimated outputs. This type of DQN is called double DQN and the approach is known as double DRL.

### D. POLICY HILL CLIMBING (PHC)

In the action value-based RL-procedure, at first, the optimal action-value pair is derived. Then, from the optimal action-value pair, the optimal policy is determined. The action value-based procedure follows a tabular mechanism, where the values are stored against an action. The highest value for an action from a state is the optimal value. However, for a small state space, the tabular mechanism works fine but, with the increasing number of state spaces, the memory problems begin [52]. This problem can be easily solved by implementing a policy-based solution. Rather than learning action value, a policy-based method learns the optimal policy directly. The most straightforward policy-based algorithm is the policy hill-climbing algorithm. In the hill-climbing algorithm, the optimal weights of a policy can be found. The agent tends to improve the weight of the policy over time by interacting with the environment. The weights are evaluated based on their return. Some initial guesses are taken at first. Later on, the weights are updated by interacting with the environment. The weights obtained from an episode are degraded with some added noise, in order to get newer weights. For every iteration, the best weight is taken to search for a new policy where newer best weights will be found.

## III. RL-BASED ROUTING PROTOCOLS FOR VANETS

In this section, RL-based VANET routing algorithms are discussed and analyzed in terms of their working procedure, advantages, disadvantages, and best-suited applications. These routing algorithms are categorized into hybrid, position-based, topology-based, hierarchical, and security categories. Fig. 3 shows the taxonomy of the investigated routing protocols.

### A. HYBRID ROUTING PROTOCOLS

In the hybrid routing protocols, traits are inherited from reactive and proactive routing protocols. Some of the hybrid routing protocols analyze the traffic and mobility conditions. Based on the results, the protocols switch their type of operation. Other types of hybrid routing algorithms define zones or clusters. These protocols maintain tables differently for in-zone members and out-zone members. They are mostly designed to be proactive in the case of zone members and reactive for in-case transmission of packets to other zones or cluster members [53].

#### 1) RL-BASED HYBRID ROUTING ALGORITHM (RHR)

Ji *et al.* proposed an RHR routing protocol [54] for the VANET paradigm, which updates the freshest path information using an RL technique. The authors noted that the blind path problem occurs frequently in traditional VANET routing algorithms. Due to the high mobility, a valid path from a source to a destination can be broken before the path expires.

This situation is described as the blind path problem. Due to this problem, the number of successful packet deliveries decreases, and packet loss increases. Rather than depending on a single path, RHR explores multiple paths. Based on a packet-carry-on feedback system, RHR assigns rewards to certain paths and also penalizes certain paths. Fig. 4 illustrates the blind path problem in the VANET. In the figure, at $t_0$, the destination $D_v$ is within the communication range of the source $S_v$. However, at $t_1$, due to high mobility, $D_v$ goes out of the range of $N_1$. Therefore, the path from $S_v$ to $D_v$ becomes a blind path. The application of RHR always updates the freshest path, and the data route will be continued through $S_v - N2 - N5 - D_v$ and $S_v - N3 - N4 - N_5 - D_v$. The routing algorithm penalizes a certain path wherein the number of control packets is relatively high and packet drops occur frequently. In addition, it assigns rewards to those paths that can improve the packet forwarding mechanism. Based on the data mined from the packet received, the routing algorithm updates its forwarding table and chooses the best forwarding path from the table. This calculation occurs when there is no path to forward the data packet towards the destination. To minimize the routing overhead, a conditional routing technique is utilized in RHR. Depending on the neighboring states, an agent may need to evaluate many states, which increases the routing overhead of the RHR algorithm. To minimize the overhead of the RHR, it selects only a fixed number of states. The vehicles only save information about the fixed number of neighboring states. A vehicle also considers the number of neighboring nodes before rebroadcasting.

Advantages: To mitigate a broadcast storm, the protocol uses an adaptive broadcasting technique that predicts the future position of vehicles. The authors have previously stated that to predict the correct movement of the vehicle, the time interval of the broadcast must increase. The time to live (TTL) value of a broadcast packet is also kept at a minimum compared to the TTL value of the data packet.

Disadvantage: The protocol does not state how it selects a fixed number of neighbors among the available neighboring protocols. Moreover, the receipt of a broadcast control packet from a specific path does not necessarily mean that the path is bad compared to other paths that contain only data packets.

Application: The protocol does not require assistance from the RSU; therefore, RHR is also applicable to the rural scenario. In addition, the protocol does not state any mechanism about the recovery policy, which may return a bad result in the sparse network condition.

### 2) Q-LEARNING AND GRID-BASED ROUTING PROTOCOL FOR VEHICULAR AD HOC NETWORKS (QGRID)

Li *et al.* proposed a Q-learning-based VANET routing protocol QGRID [55]. This routing protocol considers the routing decision from two viewpoints. One is macroscopic, and the other is microscopic. The total geographic region is divided into grids to conceptualize the learning environment of the routing protocol. The macroscopic decision process is responsible for choosing the best next routing grid, whereas the microscopic decision-making process is responsible for choosing the exact vehicle in the chosen next-hop grid. For a given destination, Q-values are calculated for the grids based

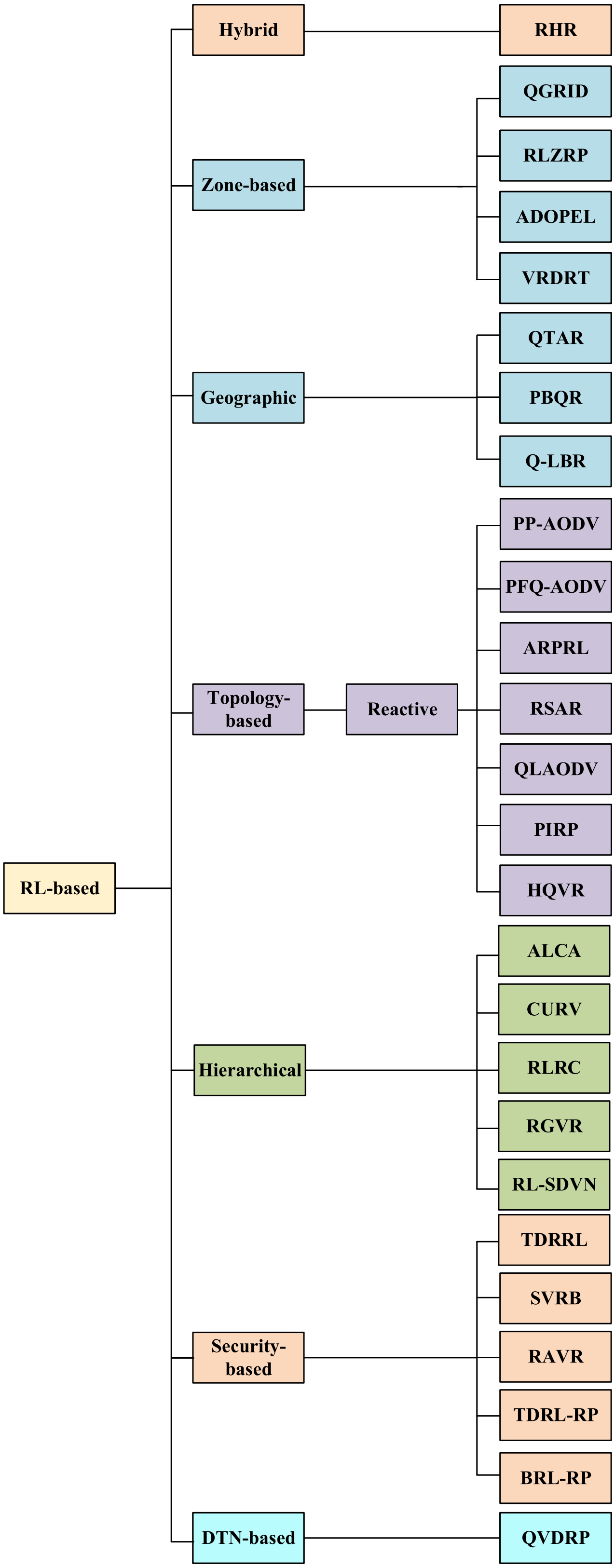

FIGURE 3. Taxonomy of RL-based routing protocols for VANETs.

on the movement paradigm of the vehicles. The selection of the next hop for the data forward mechanism is performed in two ways. In the first process, the vehicles greedily select the next hop that is geographically closer to the destination. In the second process, the sender vehicles forward their data to the vehicle with the highest probability of moving to the calculated next best zone based on the second-order Markov chain. The authors stated that in the VANET scenario, the reward cannot be calculated until a message is delivered to the destination. As a result, the environment model in QGRID is assumed to be a modeless environment. The QGRID routing protocol aims to select the grid with the highest vehicle density, to reach a destination. Fig. 5 shows the grid system used in the QGRID routing protocol. The source vehicle is in grid $S_4$, and the destination vehicle is in grid $S_3$. All the arrows and the corresponding values inside the rectangular box represent the Q-value for exiting or entering a corresponding grid. The simulation is performed based on a true dataset from taxies in Shanghai. The dataset includes the directions, time signatures, longitudes, latitudes, and unique IDs of taxies. The dataset shows a specific pattern in the movement of the taxies. Based on this pattern, the authors calculated the data in an offline manner. The grids were assumed to be square-shaped geographical areas.

Advantages: The Q-learning algorithm is run on historical data from the city of Shanghai. This decreases the possibility of a broadcast storm. By following this procedure, the convergence speed also decreased.

Disadvantages: The routing protocol works only in an offline manner. The dataset is also created based on the collected data from the taxies, which can vary due to irregular situations or accidents. The routing protocol will not be functional or can be erroneous in such cases. It should be noted that the primary function of VANET is to provide emergency information to vehicles to enhance security.

Application: This routing algorithm requires historical data to operate. Therefore, the prerequisite is to gather the vehicle movements and establish a centralized data collection scheme. An application without such a facility will yield an erroneous result.

### 3) RL ASSISTED ZONE BASED VANET ROUTING PROTOCOL (RLZRP)

Tamsui *et al.* proposed RLZRP, which implements the RL technique to train the routing table [56]. The routing table is trained to identify a suitable hop to deliver packets to the destined zone. This mechanism enables the protocol to increase the link stability of the discovered link to the destination node, and it also reduces the number of instances of packet elimination and path recalculation. This algorithm attempts to adopt the functionality of switching. In a switch, the packets are forwarded based on the MAC address and the specific port number. Inside a switch, there is a table that stores the MAC address of a device and its corresponding port number. Thus, the switch can forward the correct packet to the appropriate user. In RLZRP, this mechanism is mimicked and implemented for the delivery of the data packet to the destination address. RLZRP stores a pair of values that represent the ''Junction's'' and ''Vehicle's'' IDs. A hello packet also contains the same information while receiving this data. The vehicle updates its routing table with the corresponding freshest value obtained from the received packet. This information is taken into consideration to route a packet to its destination. In some common routing protocols, when the desired next hop is not within the limit of the current hop, the packet is discarded. However, RLZRP delivers the packet greedily, which increases the chance of successful transmission.

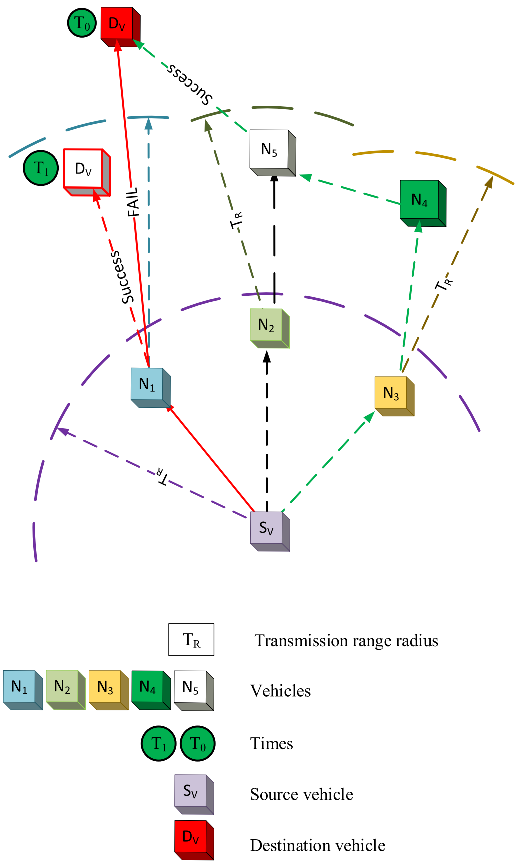

**FIGURE 4.** Visualization of blind path communication between nodes.

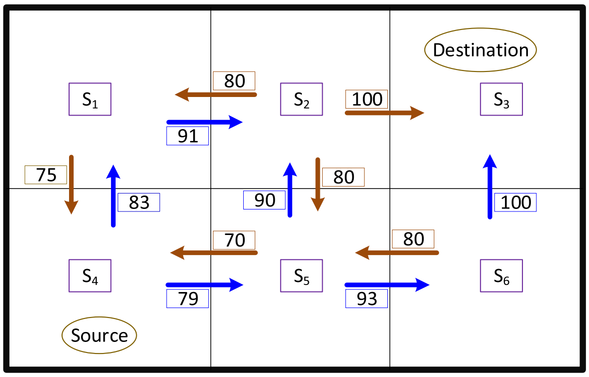

**FIGURE 5.** Illustration of grid-based Q-value system implemented in QGRID.

Advantages: In the case of unavailability of the next hop of the routing table, the packet is greedily forwarded to the next hop, which reduces delay, avoids unnecessary searching, and increases the PDR.

Disadvantages: Even though there is a chance that forwarding a packet greedily may reduce the delay, the actual result might vary. Greedy transmission might fail because of the unavailability of the next hop. However, due to the unavailability of the required information, the source node delays the search for a new route. Thus, there is an increase in the delay as well as the network congestion because of the greedy forwarding technique. Subsequent packets could also use the discovered route and bypass the delay.

Application: This is a general routing algorithm for VANETs and applies to sparse or dense areas.

#### 4) ADAPTIVE DATA COLLECTION PROTOCOL USING RL (ADOPEL)

Soua *et al.* proposed ADOPEL and utilized a distributed Q-learning algorithm to establish a routing protocol that is more adaptive to the vehicle's mobility and change in topology [57]. The Q-learning technique is applied based on the delay of the links and the number of aggregable data packets. ADOPEL uses information from the global positioning system (GPS) location services. This routing protocol controls the type of control messages it utilizes. ADOPEL only uses two types of messages. One is beacon messages, and the other is the event-driven messages that carry information based on the different information in the routing process. The beacon file includes vehicle-specific information such as velocity, position, and direction. The beacon packets are transmitted regularly, whereas the event-driven message is transmitted only when there is a need to gather traffic information. This routing protocol assumes a new kind of infrastructure that is similar to RSU and is called the traffic control center (TCC). According to the description, vehicles collect traffic data and transmit them to the TCC. Thus, the TCC obtains a global vision of the traffic throughout the network. To enable the distributed learning mechanism, the vehicles interact with each other and exchange information. The aggregation process is modeled as the MDP process with the objective of utilizing the RL algorithm to address the routing issue. ADOPEL has been used to apply the Q-learning algorithm to all the RL algorithms by utilizing its model-free nature. The reward function is designed based on a vehicle's neighboring nodes and the propagation delay with data aggregation. The utilized reward function can be described by the following equation:

$$r = \begin{cases} \beta * \left(1 - \dfrac{1}{neighbor_{number(i)}}\right) + (1-\beta) * \left(\dfrac{adv\ (ij)}{adv\ (i)_{avg}}\right) \\ \tau_1 \qquad if\ next\ hop\ is\ the\ destination \\ -\tau_1 \qquad if\ the\ node\ doesn't\ have\ any\ neighbors \end{cases} \tag{8}$$

where the benefits of a node $i$ to node $j$ are denoted as *adv(ij)*, and $adv\ (i)_{avg}$ refers to the advantages of node $i$ to the destination vehicle $D$. $\beta$ is the normalized factor that balances the weight between the two parameters. $\tau_1$ is a positive reward. $neighbor_{number(i)}$ represents the number of neighbors of node $i$. To handle the link stability, ADOPEL has adopted the variable discount factor. When a node receives a relay request, it first collects information from its neighborhood. The priority of the neighboring node is given based on the node degree and the distance from the destination. The second classification is performed by choosing the relaying node according to the Q-value stored in the Q-table.

Advantages: Information collection is limited by introducing a parameter, $d_{collect}$. This parameter represents the intra-vehicular distance that can initiate the information-gathering process. This technique is similar to the zone concept that is widely used in VANETs. This algorithm also adopts a strategy to address the void problem. If neighbor vehicles are not available, the reward is negative and avoided as an intermediary node.

Disadvantages: The existence of the TCC is a strong assumption, and the highways do not have such infrastructure in reality. However, the assumed functionalities can be given inside the RSU, and only then will ADOPEL achieve feasibility for practical implementation. The biggest disadvantage lies in the simulation setup of the ADOPEL. A single grid simulation area in which all the vehicles start at the same time creates a generic simulation scenario and is not aligned with real-life road conditions.

Application: ADOPEL is applicable in the highway environment, and the description presented in this report also supports the analogy.

#### 5) VANET ROUTING USING DEEP RL TECHNIQUE (VRDRT)

Saravanan *et al.* proposed VRDRT [58], in which they used deep reinforcement learning (DRL) algorithms to predict the movements of vehicles in a road segment. The authors argued that to reduce the store carry forward (SCF) mechanism, a routing algorithm should predict the densest road segment. Due to the high-mobility, the traffic conditions of the roads change frequently. Thus, the authors proposed a VRDRT routing algorithm that uses the DRL technique to predict the traffic conditions of a road segment at a given time. According to VRDRT, every RSU collects and maintains the vehicle's information on the road segment and runs the DRL algorithm to predict the traffic condition. Along with the traffic condition, the DRL technique is also used to calculate the transmission delay and the destination position, which yields a significant improvement in the performance of VRDRT. In this routing algorithm, the roads are segmented into multiple clusters based on the density of the vehicles. The density is calculated for a specific time in a given region by comparison of the total number of vehicles in the topology. Based on the transmission probability, VRDRT utilizes one transmission matrix to determine the best available route. The authors in [58]argued that due to the high speed, obtaining the exact GPS value of a vehicle is difficult in the VANET scenario. The working procedure of VRDRT is divided into two phases: the route selection phase (RSP) and the route establishment phase (REP). REP is responsible for finding the route, whereas RSP is responsible for searching for the optimal route based on the discovered routes. Both phases use the DRL technique to achieve the desired outcome. In the REP phase, the vehicles broadcast hello messages to inform their neighbors about their current situation, which includes the density, distance from the RSU, position, and delay for a particular area. When a packet arrives at a destination vehicle, the reward is a constant value; otherwise, the reward for the intermediary nodes is selected by the transition function. Based on the distance and density level, a vehicle node accepts or rejects data packets from a specific route. At the end of the REP phase, the RSP phase starts with the aim of selecting the optimal next-hop neighboring vehicles. The RSP operation is divided into equal time divisions. The optimal path selection is performed by the DRL agent based on the previous experience; thus, the approach can be regarded as supervised learning.

Advantages: VRDRT applies the learning technique on top of the traditional routing algorithm, which ensures a better routing performance compared to traditional routing.

Disadvantages: The process of calculating the vehicle density in the case of VRDRT only reveals the relative density of the vehicles on a road segment. The result shows which road is denser but does not reveal whether the density is sufficient for the propagation of data. It may be that other road segments are also capable of routing the data successfully, but VRDRT does not consider these road segments.

Application: It is previously indicated in this report that the protocol is tested in an urban area. Moreover, VRDRT depends heavily on the RSU functionalities. Therefore, a good infrastructure environment is necessary, which is often unavailable in highways or urban areas.

## B. GEOGRAPHIC ROUTING PROTOCOLS

The geographic routing protocols are aided by geographic information from the location that provides the services. Based on this information, the vehicles take the routing decision. Geographic position-based routing algorithms use GPS values to locate the destination and suitable intermediary nodes [59].

### 1) Q-LEARNING BASED TRAFFIC-AWARE ROUTING PROTOCOL (QTAR)

WU *et al.* proposed a Q-learning-based traffic-aware routing protocol called QTAR [60]. The algorithm takes advantage of the benefits of the geographic routing paradigm and also successfully utilizes the RSU to deliver the routing packet to the destination. The Q-learning algorithm is implemented in QTAR for the vehicle-to-vehicle (V2V) and RSU-to-RSU (R2R) data transmissions. For V2V routing in QTAR, the packets are assumed to be the agents, and the vehicles are assumed to be the states. For R2R routing, the hello packets are also considered as the agents, and the neighboring RSUs are considered as the states. Fig. 6 describes the fields of the hello packets used in the QTAR routing algorithm. Two different types of hello packets are used for V2V communication and R2R communication. In the V2V communication, the hello packets include the RSU's or vehicle's unique ID, timestamp of the packet, x- and y-axes value, velocity of the node, and the entering and upcoming intersection addresses. $Q_{MAX}$ represents the maximum Q value of the RSU required to reach the next hop, and NH simply denotes the next hop. In the R2R hello packet, the ID of the RSU is included with the timestamp, $Q_{MAX}$ values, and the $Q_{MAX}$ values' count. A $Q_{MAX}$ field contains the destination RSU ID as $RSU_{Dest}$, corresponding Q value, and the id of the next RSU as $RSU_{Next}$.

To determine the Q value of a state, high connection reliability, and minimization of the EED are considered. The protocol assumes that most of the road segments are occupied by one RSU, which can partially communicate with the adjacent road segment. The algorithm used in this investigation shows that the vehicles use an SCF mechanism in the event of unavailability of the next hop to transmit the data. The algorithm utilizes specially formatted hello packets to determine the Q-value of a state. QTAR is a traffic-aware urban routing protocol that considers road intersections.

Advantages: The implementation of Q-learning for the selection of the next hop increases the throughput and PDR.

Disadvantages: QTAR does not estimate the vehicle's direction, which will impair the performance of the protocol in real life.

Application: The protocol assumes the existence of an RSU in every road segment. This assumption renders the protocol applicable to only urban areas.

### 2) POSITION-BASED Q-LEARNING ROUTING (PBQR)

Sun *et al.* proposed an RL-assisted position-based routing technique for the VANET paradigm called PbQR [61] The reliability and stability of the link serve as selection parameters to choose the next-hop node for transmitting data to the destination. PbQR considers the vehicles as states in the formulation of the RL algorithm. The combination of all the vehicles in the networks constitutes the state space of the RL algorithm. Periodic Hello messages are used in PbQR to exchange information about neighboring nodes. According to the Q-learning algorithm used in PbQR, the agent always performs a greedy action. Greedy action in Q-learning means that the agent always performs the best available action in the Q-table. PbQR calculates the stability factor and the continuity factor to evaluate the link quality for the selection of the next-hop node. The links with short periods tend to fail more often compared to links with long periods. The stability factor of the PbQR algorithm can be evaluated as follows:

$$SF_t(c,x) = \begin{cases} 1-\dfrac{|D_t(c,x)-D_{t-1}(c,x)|}{T_R} & |D_t(c,x)-D_{t-1}(c,x)| \le T_R \\ 0 & otherwise \end{cases} \tag{9}$$

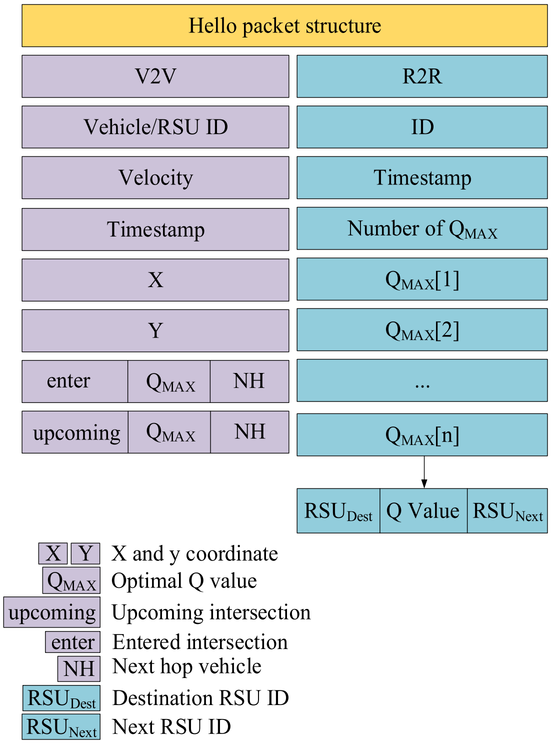

**FIGURE 6.** V2V and R2R hello packet structure used in QTAR.

where the stability factor for nodes $x$ and $c$ at time $t$ is $SF_t(c, x)$. $D_t$ and $D_{t-1}$ represent the Euclidian distances between two nodes at time $t$ and $t-1$, respectively. $T_R$ represents the transmission range of the vehicles. The value of $SF$ varies between 0 and 1. The greater the value of $SF$, the better the link quality between the examining nodes. PbQR considers another important factor, that is, the node degree. If this factor is not considered, the transmitting nodes select the next hop in situations when nodes are not available. The continuity factor used in PbQR indicates the node degree of the neighboring nodes. The continuity factor can be calculated based on the following equation:

$$CF(c, x) = \frac{NUM_x}{NUM_{max}} \tag{10}$$

where $NUM_X$ indicates the node degree of node $X$. $CF$ is the continuity factor of node $c$. $NUM_{max}$ is the maximum node degree based on an examination of node $c$. The reward function is the summation of the continuity factor and the stability factor of a node. The distance factor is used to determine the distance relationship between the source and destination nodes. The discount factor needed for Q-learning is implemented using this distant factor.

Advantages: PbQR considers $SF$ as one of the deciding factors for the selection of the next-hop node. The routing algorithm also applies a mechanism to avoid the bias of the parameter by adding weighting factors for two consecutive times. The bias can be caused by the relative distance as a result of acceleration or deceleration. The same mechanism is also applied for the continuity factor.

Disadvantages: The Q-learning algorithm adopts a greedy approach for the selection of Q values from the Q-table.

Application: The routing algorithm is applicable in general-purpose situations. The absence of a recovery model, RSU dependency, and traffic light considerations render the algorithm suitable for only dense regions.

### 3) Q-LEARNING BASED LOAD BALANCING ROUTING (Q-LBR)

Roh *et al.* has proposed a load balancing routing protocol for VANET called Q-LBR [62]. This routing protocol is assisted with UAV to enable NLOS communication for the ground vehicles. The load balancing mechanism in Q-LBR is established in three main ways. First, the authors proposed an overhead optimized ground vehicles' load estimation technique with the help of the UAV. In this technique, based on the broadcast messages, the UAV gets to know the queue size of the ground vehicles. This is executable because the UAV has the ability to create an NLOS communication with the vehicles. Second, the Q-learning technique is applied for establishing the load balancing data communication by defining the UAV routing policy area (URPA). Finally, a reward function is specially designed for quicker convergence. Q-LBR defines three types of packets. They are urgent service messages, real-time service, and connection-oriented protocol which have the highest, medium, and low priority, respectively.

The working procedure of Q-LBR is divided into two phases. In the first phase, the UAV collects the ground vehicles' congestion conditions by hearing the broadcast messages, and then detect the congestion level. The information about the URPA is broadcasted in the second phase. The broadcasting information contains the ground nodes' congestion information and also the UAV's congestion information if the UAV is used as the relay node. The path discovery process in Q-LBR is similar to the reactive routing protocols such as AODV and DSR. The RREP packet is sent back in all the paths which include the optimal and near-optimal solutions. The replied packet has all the paths which also have paths including the UAV. When the best route is unavailable, other routes can be chosen. The queuing load can be calculated based on the following equation:

$$q_{ground_i(t)} = \frac{AQL_i(t)}{MQL_i} \tag{11}$$

where $MQL_i$ and $AQL_i$ are the maximum and average queue length for a vehicle $i$ at time $t$, respectively. The objective of the learning procedure is to find a suitable URPA that will keep the congestion level as close as to the threshold limit. Q-LBR adopts a quick convergence technique, which ensures a better outcome as the environment is dynamic.

Advantages: Q-LBR has multipath support, which ensures less route discovery packets to be transmitted. The learning process is triggered only when both the ground nodes' congestion threshold and the UAVs' threshold are not met. This procedure also reduces the number of broadcast messages.

Disadvantages: The addition of UAV is a bottleneck of the proposed Q-LBR. A UAV-aided routing algorithm raises questions such as optimal deployment, height optimization, and the number of UAVs. None of the above scenarios is taken into assumption.

Application: This routing protocol takes assistance from UAV. The UAV is an easily deployable and replaceable unit. Q-LBR will be especially acceptable in the areas that the amount of generated data is large as in the urban area. However, in the disastrous area, Q-LBR will also be operable because of the easy deployment of UAV.

### C. REACTIVE ROUTING PROTOCOLS

Reactive routing protocols determine the route when a node needs to transmit data. This routing protocol conserves the bandwidth of the network and is applicable to the high-speed mobility scenario. However, the delay is higher compared to proactive routing [63].

#### 1) POINT TO POINT AD-HOC ON-DEMAND VECTOR (PP-AODV)

Valantina *et al.* proposed a fuzzy constraint Q-learning-based routing algorithm called point-to-point ad-hoc on-demand vector (PP-AODV) [64]. The routing algorithm is a modified version of the well-known AODV routing algorithm with integrated intelligence based on the implementation of learning techniques. The original algorithm is modified so that it more suitable for the VANET environment. PP-AODV considers multiple parameters for the optimization process. These parameters include the bandwidth of the link, delay performance of a link, and the probability of packet collision. The protocol is assisted by the mobility pattern of the neighboring vehicle, even when positional information is unavailable. The rest of the routing mechanisms are kept similar to the original AODV routing protocol. To start a transmission, a route request (RREQ) packet is generated by a source to find the destination. The Q-values in the Q-tables is maintained using the RREQ message. To send the route reply (RREP) message to the source, the destination node utilizes its Q-tables. The best node is selected according to the Q-values as the next hop for sending the RREP message to the source node.

Advantages: The protocol can estimate the movement of other vehicles without the need for any positioning technology such as GPS.

Disadvantages: The protocol does not state any mechanism in the case of data failure or unavailability of the neighboring node. In the case of a sparse network, this routing algorithm does not perform according to expectations.

Application: Without any recovery policy, the algorithm performance is similar to that of the original AODV, and no performance improvement is observed. As a result, the applicability remains the same as that of the original AODV in urban or highway areas.

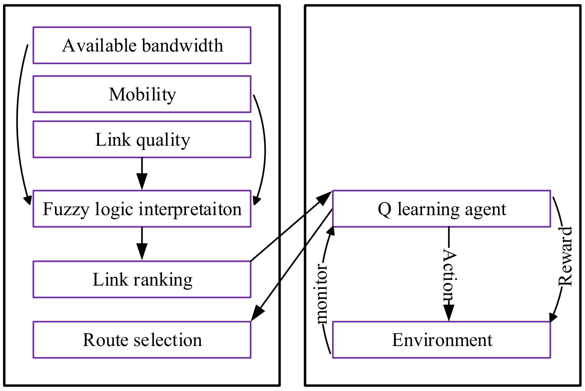

**FIGURE 7. Fuzzy logic-assisted route-selection mechanism in PFQ-AODV.**

#### 2) PORTABLE FUZZY CONSTRAINTS Q- LEARNING AODV (PFQ-AODV)

Wu *et al.* proposed a modified version of the AODV routing protocol called PFQ-AODV [65]. In this modified approach, VANET learns to transmit data packets through the optimal route using a fuzzy constraint and Q-learning algorithm. The direct transmission link is evaluated using fuzzy logic, whereas the multi-hop links are evaluated based on the Q-learning algorithm. The routing algorithm attempts to determine the optimal route in terms of the bandwidth of the link, the present link quality, and the change in the vehicle's direction and speed. Fig. 7 describes the route selection procedure based on fuzzy logic constraints and Q-learning mechanisms. The RREQ packet was used to evaluate the parameters of the links. Based on the received hello packets from the neighboring nodes, the vehicles predict their future position. First, the protocol broadcasts the RREQ message to the neighbors, who in turn rebroadcast the RREQ message. When a destination node receives the same RREQ packet, it compares the old path with the new path. It should be mentioned that PFQ-AODV maintains two-hop neighbor information inside its neighbor table. The mobility of a node is calculated based on the relative position change information of the vehicles or by evaluating the stored information from the two-hop neighbor table. Each vehicle in the network has a Q table wherein the Q-values are stored and range from 0 to 1. A vehicle stores three types of Q values. The first Q value is stored for the one-hop neighbor, the second Q value is stored for the second hop neighbor, and the third Q value is stored for the source node that generates the traffic. The Q-value of a vehicle is broadcasted using hello messages among the neighbors. Thus, the size of the hello message is dependent on the cardinality of the neighboring vehicles. In PFQ-AODV, the RREP packet mechanism works in the same way as in the original AODV. In the case of choosing the best next hop, the current vehicle chooses the best node with the highest Q-value from its table.

Advantages: The performance of the PFQ-AODV is evaluated and tested in a real-life environment, which validates the protocol's performance. This protocol has no dependency on the lower stack of the networking layers. PFQ-AODV can also calculate the mobility factor without the need for position information.

Disadvantages: No broadcast mitigation technique is adopted. With the increase in the number of neighboring nodes, the broadcast storm might be a regular incident in the PFQ-AODV. The routing delay is a major issue in this routing protocol in an obstacle area.

Application Scenario: In the highway scenario, the routing protocol is better aligned. There is a possibility of a broadcast storm in the network, and no recovery policy is stated. Therefore, a limited but reasonable number of vehicles is required to effectively run this routing protocol.

### 3) ADAPTIVE ROUTING PROTOCOL BASED ON RL (ARPRL)

Wu *et al.* proposed an RL-based mobility-aware VANET routing algorithm called ARPRL [66]. Using periodic hello packets, ARPRL only maintains the freshest valid path in its routing table. The routing table applies a distributed Q-learning technique to learn about the freshest link for multihop communication. While updating the Q-values of the neighboring vehicles, the host vehicle also transmits its mobility information. Thus, a vehicle can learn about the sender vehicle's mobility model. This protocol uses a feedback mechanism for the packet loss information from the MAC layer, which causes the Q-learning technique to be better adapted to the VANET environment. To learn about the nodes' mobility model, ARPRL utilizes a vehicle's positional information such as the current position, the direction of travel, and the speed. Due to the high-mobility, the control packet is frequently exchanged to keep the Q-table updated. To learn about the breakage of the link, data packets are used in ARPRL. Both temporal difference and the Monte Carlo technique are used to obtain the optimal value function. ARPRL maintains two distinct tables: a Q-table or a routing table and a neighbor table. Using the hello timer, the arrival or exit event of a neighbor node is detected. Along with the vehicle's position, speed, and direction, additional information extracted from the Q table is also added inside the hello packet. To route a packet towards a destination, the vehicles first examine their Q-table. If no suitable next hop is found, the vehicle initiates a route probe request that is similar to the RREQ packet of the AODV routing protocol. To facilitate faster convergence, the algorithm initially implements a proactive learning procedure. Route looping is reduced using a modified version of the hello packets. The position of a vehicle, timestamp, and Q-values are inside the hello packet. The information from the Q table is broadcasted to the neighboring vehicles to keep the Q tables updated after every hello packet interval. In addition to LPREQ, LPREP also contributes to the update of the Q table.

Advantages: The cost of feedback from the MAC layer is negligible compared to that of updating the Q table.

Disadvantages: The size of the hello packets increases significantly. Given that the packets carry the maximum Q-values from the nodes, the size increases significantly with the increment in the number of intermediate hops. Another major disadvantage of the proposed routing protocol is that it does not consider the movement direction and velocity of the nodes.

Application: To maintain the good performance of APRL, a large number of vehicles is needed. The protocol did not consider the SCF mechanism or the presence of the RSU; therefore, packet drop has a normal ratio in the case of a sparse network.

### 4) RELIABLE SELF-ADAPTIVE ROUTING ALGORITHM (RSAR)

RSAR identifies and analyzes the various reasons for link disconnection between vehicles to provide a QoS-optimized routing experience [2]. This routing utilizes link lifetime prediction, which helps the nodes to choose the best nodes for routing data. The Q-learning RL algorithm is applied to address the ever-changing VANET environment. RSAR assumes that the vehicles are distributed according to a log-normal distribution on a single line highway. This routing algorithm considers the direction of the vehicle to estimate the duration of the link. It is a practical assumption that vehicles moving in the opposite direction have a shorter valid-link duration, whereas the link duration is higher for vehicles moving in the same direction. The distance after a certain time can be calculated using the following equation:

$$\delta_{i,j} = \begin{cases} \Delta_j(t) + \Delta_i(t) + \delta_0 & \textit{same direction} \\ \Delta_j(t) - \Delta_i(t) + \delta_0 & \textit{opposite direction} \end{cases} \tag{12}$$

where $\delta_{i,j}$ is the distance at time $t$ between node $i$ and node $j$, *and* $\delta_0$ is the initial distance between nodes $i$ and $j$ at time $t_0$. $\Delta_j(t)$ is the displacement of node $j$ at time $t$, and $\Delta_i(t)$ is the displacement of a node $i$ at time $t$. The link is valid as long as $\delta_{i,j}$ is less than the communication range. RSAR considers two types of link disconnection scenarios, as shown in Fig. 8.

RSAR stores only one-hop neighbor information inside its Q-table. The first column of the Q-table contains the IDs of all neighboring nodes, and the first row contains the IDs of the destination nodes. The size of the table depends solely on the number of neighboring vehicles. Nodes gather information about the neighboring node using beacon packets. The learning process occurs in a distributed manner, which causes the algorithm to converge faster. Along with the position, velocity, and direction, the source node assumes the existence of maximum Q values of a node inside the beacon packet. At the start of the routing process, the source node first checks the destination node. If the destination node is available, the node with the maximum Q-value is selected as the next hop. If the destination node is not available in the Q table of the source node, the source node starts a route discovery process. The request beacons include the node information that is passed along the route. Upon receiving the first packet, the destination node replies with another control packet. The intermediary nodes modify the next-hop information, and a

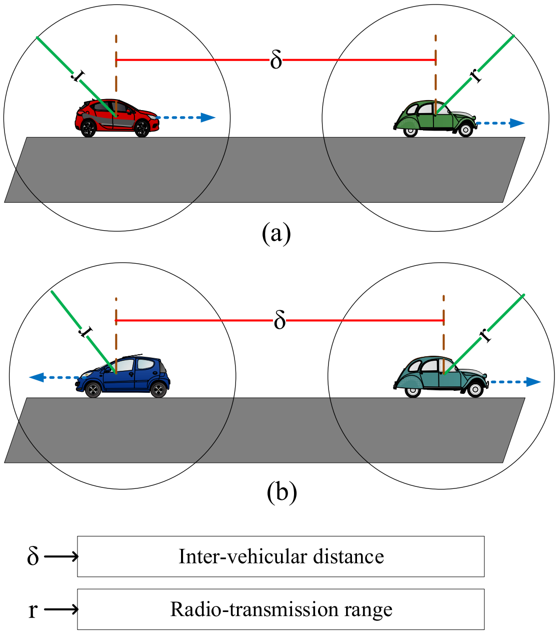

**FIGURE 8.** Link disconnection scenario: (a) vehicle in the same direction and (b) vehicle in opposite directions.

single-hop broadcast occurs. The receiving nodes update the Q-table value and discard the packet. When the source node receives the packet, a route is identified, and the Q-table is further updated for the source node. To keep the Q-table updated, the nodes periodically broadcast a route-update hello packet. The transmission delay is chosen to be a random number from 0.5 to 1.

Advantages: Every Q-value of a corresponding destination node has a timer. This helps the vehicle to update the Q-values of the node, and the freshest path is included. As a result, the packet drop ratio is dramatically decreased.

Disadvantages: Even for a single-hop broadcast of the RREP packet, the control packet overhead increases significantly. In a VANET scenario, control overhead is an important issue to be prioritized.

Application: The single-hop broadcast mechanism renders this routing algorithm more appropriate in a dense network topology.

### 5) Q-LEARNING-BASED AODV FOR VANET (QLAODV)

Wu *et al.* proposed a distributed RL-based vehicular routing technique called QLAODV [67]. This protocol is especially applicable to high-speed mobility scenarios. QLAODV utilizes the Q-learning technique to predict vehicular information and utilizes control packet unicasting to adapt to the dynamic scenario. QLAODV also considers the dynamic changing topology of the VANET scenario by implementing rapid action in the event of topology changes. At the beginning of routing a packet, QLAODV operates as a simple reactive routing protocol and looks for the destination node entry inside its routing table. In the case of the unavailability of the destination node, the source node initiates the discovery process to establish the route. Link state information is separately predicted by the Q-learning algorithm in QLAODV for all the vehicles. The vehicles act as agents in the RL environment. The Q-learning model takes the hop count, link lifetime, and data rate of a link as the selection parameters. The neighbors are considered as the states, and the state's transition is the packet transition from one vehicle to another. The authors in [67] contend that due to the absence of a global view, the centralized approach is not suitable for the VANET scenario. The reward mechanism is such that a node obtains a full reward when a packet reaches the destination. In contrast, if a node receives a hello packet from a destination, it also receives a reward of 1; otherwise, the reward is 0. QLAODV uses a dynamic Q-table, wherein the size is dependent on not only the neighbor node but also the destination vehicles. Upon receiving hello messages, the nodes update the Q-values inside the Q-tables. This approach for exploration based on the hello messaging system allows QLAODV to utilize greedy approaches while routing the data packet. Every node derives and utilizes a mobility factor to calculate its stability. The mobility factor can be calculated based on the following equation:

$$MF_x = \begin{cases} \sqrt{\dfrac{\left|N_x \cap N_x^p\right|}{\left|N_x \cup N_x^p\right|}}, & \textit{if } N_x \cup N_x^p \neq \emptyset \\ 0, & \textit{otherwise} \end{cases} \tag{13}$$

where $MF_x$ is the mobility factor of a node $x$, the set of neighbor nodes is denoted as $N_x$, and the set of neighbors when the last hello messages are sent is represented by $N_x^p$ for node x. The bandwidth factor is another important parameter used in the Q-learning process of the QLAODV routing algorithm. The bandwidth factor can be written as follows:

$$BF_x = \frac{\textit{Available bandwidth of } x}{\textit{Maximum bandwidth of } x} \tag{14}$$

where $BF_x$ is the bandwidth factor of a node. The hello messages include the bandwidth factor, mobility factor, and maximum Q-value inside it. The intermediary route change mechanism in QLAODV uses the RCNG-REQ control packet. If any intermediary node finds a better route, it immediately starts forwarding the data packet via the route.

Advantages: The utilized Q-learning algorithm implements a multi-parameter-based variable discount factor. The hop count, link condition, and free bandwidth of a link are taken as relevant parameters to derive the value of the discount factor. After propagating through a link, the value is discounted based on the experience of the bandwidth and link condition.

Disadvantages: The RL works best if the feedback mechanism is performed immediately, but in QLAODV, the feedback mechanism is enabled using the periodic hello messages. However, this mechanism results in a reduction in the number of control packets. The mobility factor calculation yields a relative result and not the exact mobility of the examining node. The use of RCNG-REQ and RCNG-REP control packets potentially exacerbates the control packet overhead problem.

Application: QLAODV is utilized in both the urban environment and freeways. The simulation is performed for both environments for a variable velocity, which validates the performance of the protocol in both cases.

### 6) PRACTICAL AND INTELLIGENT ROUTING PROTOCOL FOR VANET (PIRP)

Wu *et al.* proposed PIRP [68], a routing protocol for VANET architecture. To propose the routing protocol, the authors went through some experimental analysis and tried to find out the major flaws in the existing research works. As a result, they showed that the packet reception ratio depends on the size of the hello packets, the number of available nodes, and the distance between sender and receiver. Thus, the use of hello packets to indicate link quality might return an erroneous result. The Q-learning algorithm is used to find out the best modulation coding scheme so that the reception ratio is increased. In this learning phase, the network is the environment, each vehicle is an agent, and the reception ratio of the hello packet is the state. The state is discredited with an interval of 100. Selecting a modulation coding scheme is the action in this learning procedure. The $\epsilon-$greedy is taken to set the balance between exploration and exploitation.

The transfer learning technique is used to share knowledge among the vehicles and to speed up the convergence of the learning procedure. The knowledge transfer procedure starts when a node enters the region. A learned node requests the transfer process. The lifetime of these learned values is disproportional to the distance. In order to make a routing decision, PIRP uses both fuzzy logic and Q-learning mechanism. For a point-to-point connection, the fuzzy logic is used; otherwise, the assistance from Q-learning is taken. In the implementation of the Q-learning algorithm, the AODV is assumed to be the niche algorithm. Transmission rate, vehicles' mobility, the number of hops are taken to rank the discovered route with the help of RREQ and RREP packets. Indirectly, the vehicle's relative movement is also taken into consideration, as the hello packet reception ratio changes drastically in case of higher relative mobility. The link stability is calculated based on the following equation:

$$ST\left(c,x\right) = \left(1-\alpha\right) \times ST_{i-1}\left(c,x\right) + \alpha \times \left|HRR_{i}\left(c,x\right) - HRR_{i-1}\left(c,x\right)\right| \qquad (15)$$

where the hello reception ratio is denoted with $HRR$, $i$ indicates the time, $c$ *and* $x$ denote the examining neighboring node, and $\alpha$ denotes the learning rate. In the routing procedure, exploration and exploitation do not conflict with each other. All the sending nodes get the prior information about the links with the help of hello packets. As a result, the sender can choose the next hop greedily from the Q table.

Advantages: A modified procedure is used to determine the hello packet reception ratio. The packets are sent based on a fixed time window. In the receiver end, the reception ratio is set based on the last ten messages received. After the route discovery phase, the Q-table is updated based on the route switching in the route maintenance phase. This mechanism will, however, reduce the number of exchanged control packets.

Disadvantages: According to the analysis of link quality in the paper, the authors stated that hello packets can be erroneous. However, in order to use hello packet reception ratio as an indication of the link quality, the parameters (distance, packet size, and the number of hops) can be normalized for the calculation.

Application: Multi-modulation scheme learning procedure will help the routing protocol to adapt in a densely deployed environment.

### 7) HEURISTIC Q-LEARNING BASED VANET ROUTING (HQVR)

Yang *et al.* proposed HQVR [69], a routing algorithm for VANET architecture. The algorithm selects intermediate hop based on link reliability. The general implementation of the Q-learning algorithm is slow and tends to consume a good number of control packets. The heuristic procedure is used to speed up the convergence rate of the Q-learning algorithm. HQVR is a distributed algorithm and the learning procedure is carried out based on the information gathered by exchanging the beacon packets. To design the HQVR algorithm, the width of the road is not considered. The distribution of the vehicles is assumed as the log-normal distribution. According to the mobility pattern adopted in the paper, the link between two nodes can be broken in case they run in the same direction with different velocity or they run in a different direction. The maximum link maintenance time between two vehicles running in the same direction is denoted by the following equation:

$$t_{i,j} = \frac{-u_n - \sqrt{u_n^2 - 2a_n\left(R + S_0\right)}}{a_n} \qquad (16)$$

where the $a_n$ and $u_n$ is the acceleration and initial speed difference between node $i$ and $j$, respectively. $S_0$ is the initial distance between the examining node and $R$ is the transmission range. The maximum link maintenance time between two vehicles forwarding to the opposite direction can be given as:

$$t_{i,j} = \frac{-u_n - \sqrt{u_n^2 - 2a_n(R - S_0)}}{a_n} \qquad (17)$$

HQVR routing algorithm is a modified version over QLAODV, which is also described separately in [69]. The authors stated that the convergence of the Q-learning algorithm in VANETs depends on the rate of beacon messages, which mainly makes the convergence slower. In HQVR, the link duration ratio is considered as the learning rate. According to the functionality of the Q-learning procedure, the learning rate determines the amount of convergence. So, with a better-quality link, the necessity for exploration decreases. Different from the original QLAODV, HQVR implements a strategy to implement the exploration technique. The packets store the delay information. When a node finds that the new delay is better than the previous delay, the node simply switches to the new route. The feedback messages travel through multiple paths to reach the destination. Thus, the source has the flexibility to choose the best route over the multiple routes.

Advantage: The special design of the learning rate will reduce the impact of the node's mobility on the data delivery rate. The learning rate depends on link quality. As a result, when the agent finds a node with better link quality immediately decreases the convergence time. The quicker the convergence is, the lesser the impact of the node's mobility is.

Disadvantages: The exploration technique adopted in HQVR is based on a specific probability value. This assumption is not better as there can be a better exploration strategy and the minimization of exploration probability with the increasing amount of time. The packets store the delay information in every intermediary node and, thus, the size of the packets will vary depending on the size of the multi-hop nodes.

Application: There is no dependency shown in HQVR on the static infrastructure. However, the algorithm does not have any recovery policy. Adding the SCF mechanism will help HQVR to be operable in the sparse road condition area.

## *D. HIERARCHICAL ROUTING PROTOCOLS*

In hierarchical routing protocols, the responsibility of the nodes is distributed at different hierarchal levels. The clustering algorithm is a type of hierarchical algorithm in which the cluster head (CH) selects the route with other CHs. When a cluster member (CM) needs to transmit data, they are sent to the corresponding CH, which then sends the data to the CH of the receiver. Finally, the receiver's CH sends the data to the original receiver [70].

### 1) AGENT LEARNING-BASED CLUSTERING VANET ROUTING ALGORITHM (ALCA)

Kumar *et al.* proposed a cluster-based routing protocol called ALCA [71]. The routing algorithm utilizes an RL technique to form a cluster among the vehicles on the road. Vehicle mobility is taken into consideration during the training of the agent to implement the clustering and routing mechanisms. The agent is used to learn the optimal path; then, the information is shared with other vehicles, which allows the sender vehicle to propagate information along an optimal path. The agent is also used by vehicles to learn about the density of the road segments, resulting in better routing decisions. The CHs monitor and maintain information about the surroundings. Along with mobility and density considerations, the trust score is taken into consideration when selecting CHs. The agents are deployed to learn the traffic condition and the vehicle direction for different road segments. The agents are also able to communicate among themselves, which facilitates the enhancement of their learning experience. The rewarding and penalization schemes of the agent continue until the agent reaches an ultimate point. Four types of agents are considered in ALCA. They are the requested launcher agent (RLA), data update agent (DUA), zone selection agent (ZSA), and speed control agent (SCA). The RLA must initialize the request for finding the best route for a mobile vehicle. DUA takes the request generated by RLA and forwards it to ZSA. Zone detection is performed by the ZSA. The zone identified by the ZSA is then passed to the SCA using DUA. The SCA is responsible for calculating the traffic flow values. The data representation of the SCA agent also shows the mobility information and the volume of the vehicles in the respective zones. Fig. 9 shows the communication among the agents. It should be noted that all the interactions are bidirectional.

Advantages: The trust score for the selection of the CH adds security features in the ALCA.

Disadvantages: The agents are not fully defined. The definition of a zone is also not clear. The one-hop approach for long-distance delivery is not practical.

Application: This protocol will perform better in a region with short road segments and high vehicle density.

### 2) CONTEXT-AWARE UNIFIED ROUTING FOR VANETS (CURV)

Wu *et al.* proposed an RL-based VANET routing protocol called CURV [72], which attempts to optimize the transmission paradigm and the size of the packet QoS parameter. The transmission paradigm can be categorized into two types: unicast and broadcast. CURV utilizes a clustering technique to limit the hop count of exchanged control packets. Control packets are exchanged among only one-hop members for intra-cluster communication. Improvement of CH-to-CH communication is achieved using the RL algorithm. The main goal of CURV is to improve the performance of the VANET routing using clustering and RL algorithms while maintaining the routing overhead. CURV assumes that all vehicles are equipped with a GPS module. The beacon period is set to a 1-s interval. The vehicles obtain information about their neighboring vehicle using this periodical beacon. The authors contend that even though only two contexts are considered, the number and types of contexts can be increased in the future. Packet type and size information are propagated from the lower network stack to the network layer to improve the decision-making process. Different sizes of hello messages are used in CURV. After experimenting with a 100-s interval, CURV selects the appropriate size of the packet length and can use multiple packets to deliver data if the payload does not match the size of the selected packet. The link condition is updated for every hello packet received from a vehicle. For each payload size, all the nodes store the timestamp, average packet reception value, and inter-vehicular distance information, which is used later to relay data. On average, four intermediary nodes are considered while estimating the reception probability for data. To minimize the CH selection count, CRUV sets a higher probability for relatively slower vehicles that travel in the same direction with good link conditions. Fuzzy logic is used to perform a clustering decision

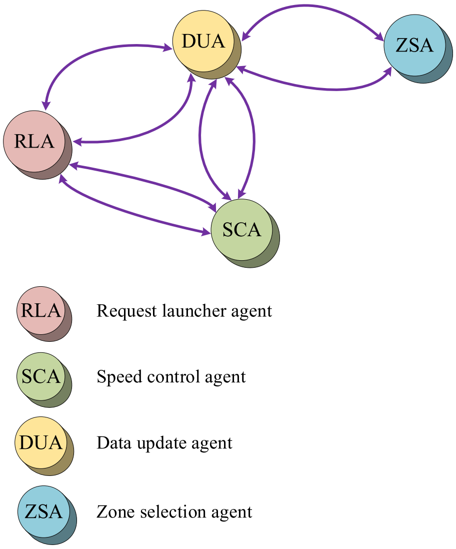

**FIGURE 9.** Internal communication paradigm of agents in ALCA.

based on the aforementioned parameters. The CH selection algorithm used in CRUV is a distributed clustering algorithm. The velocity factor was calculated based on the following formula:

$$VF\ (s,x) = \frac{|v(x)| - \min\limits_{y\in N_s} |v(y)|}{\max\limits_{y\in N_s} |v(y)|} \tag{18}$$

where $VF\ (s, x)$ indicates the velocity factor of node x from node s, $N_s$ is the set of neighbors of the node s being examined, $v(x)$ is the velocity of node x, and $v(y)$ is the velocity of node y. The channel condition factor is another important parameter that is derived based on the reception ratio of the hello packets. CURV uses a Q-learning algorithm to improve the first two-hop and last two-hop nodes based on the link condition parameters. Fig 10 depicts the two-hop optimization process. In this figure, the source did not choose the nearer CH to propagate its data; rather, it forwarded the data to vehicle F1. At the destination end, the CH forwards the data to F2, whereas F2 forwards the data to the destination.

Advantages: The test-bed experiment is performed based on IEEE 802.11 b/g/n to examine the packet receiving ratio by varying the size of the payload. This is at the core of the design of CURV.

Disadvantages: On average, the reception probability is assumed to be four, but the authors did not mention the impact of this assumption for different intermediary nodes. The cluster selection mechanism is context-dependent in CURV. It has been previously stated that the context consists of the packet size and packet type. In the protocol, the effect of the packet size context parameter on the cluster selection technique and performance is not clear. From a general perspective, it can be predicted that a vehicle with a good link quality will be able to successfully forward all packets of different sizes, whereas a vehicle with a bad connection will experience an increase in the packet drop ratio. The packet size does not affect performance.

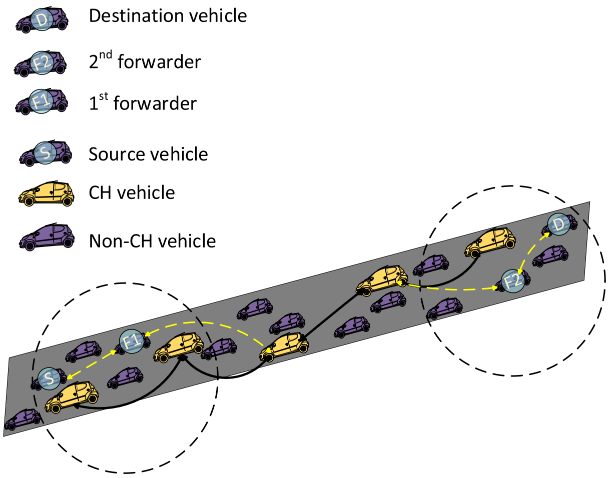

**FIGURE 10.** Two-hop optimization used in CURV.

Application: CRUV is a well-defined hierarchical routing protocol that is compatible with both dense and sparse networks. In particular, dense conditions are treated more carefully, and the data flow is hazardless in such a situation.

### 3) REINFORCEMENT LEARNING-BASED ROUTING PROTOCOL FOR CLUSTERED VANETS (RLRC)

Bi *et al.* proposed a VANET routing algorithm RLRC [73], especially for electric vehicles. Due to the shortage of electric vehicles, the authors segmented the total network into multiple clusters. An improved version of K-Harmonic Means (KHM) is used to form the cluster among the vehicles. To decrease the learning time, RLRC uses SARSA($\lambda$) RL algorithm. Electric vehicles are powered by batteries. Electric vehicles show greater trends towards automation and need to exchange a lot of packets. As RLRC forms clusters to reduce the number of state spaces, the CH will have to exchange a lot of data packets with other CHs and own cluster's CMs. Thus, RLRC considers the energy parameter of the vehicles for CH selection. To enable smooth connectivity, bandwidth is selected as the second parameter for electing the CH.

From a given road segment, RLRC first determines the number of clusters. The KHM is a variant of the K-Means clustering procedure. However, the biggest difference is that the algorithm replaces the minimum value with the harmonic mean. At first, the best positions for the centroid are calculated based on the partial derivatives. In each iteration, the value of the centroid is being improved. Based on the relative distance, the least distance node is selected as the CH. The nodes that are not selected as the CH obtains the minimum distance with all the CHs and joins as CM. Lastly, the average distance is calculated to remove the nodes having a significant amount of fluctuation. This ensures the longest lifetime of the clusters. The SARSA($\lambda$) model is used to optimize the routing process in the RLRC procedure. In this algorithm, the entire clustered VANET scenario is considered as the environment, and the CHs are considered as the agent. The Q-values are updated with the help of hello packets. The hello packets are sent periodically. The reward function is generated based on the next-hop link status.

$$ls\left(c,x\right)=\frac{\left(Bw_{max}-Bw_{hello}\right)}{Bw_{max}}\times e^{-ILD(c,x)} \qquad (19)$$

where $ls$ is the link status, $ILD\left(c,x\right)$ is the inverse link duration between node $c$ and node $x$, $Bw_{max}$ is the maximum bandwidth, and $Bw_{hello}$ is the bandwidth needed for hello packet exchanging. RLRC considers the hop count, the condition of the link, and the available bandwidth to compute the Q-value.

Advantages: By forming clusters, RLRC reduces the size of the state space. As a result, the convergence time is faster compared to the protocols that every node is a state. The average distance is considered while forming the clusters. This mechanism will increase the lifetime of the clusters.

Disadvantages: The hello packets are sent periodically. This will consume a significant amount of bandwidth, which will have an adverse effect on the throughput. However, it is a good practice that only CHs exchange this packet, which will reduce this adverse impact. The initial values are set as 0, leading to a longer time before convergence. Moreover, RLRC does not mention how to get the optimum value of the cluster.

Application: This algorithm does not consider the relative direction among the vehicles. This leaves the algorithm workable only for a single direction road segment. The simulation considers the traffic light scenario, which proves the compatibility for the urban road structure.

#### 4) RL AND GAME THEORY BASED VANET ROUTING (RGVR)

Wu *et al.* proposed RGVR [74], a routing algorithm for VANET architecture. RGVR implements a fuzzy-logic system to form stable clusters and game-theory principles to take the decision whether to join in a cluster or not. To form stable clusters, multiple parameters are taken into consideration such as the velocity of the vehicles, the movement pattern of the vehicles, and the link quality based on the received signal. The route selection mechanism is aided with an RL algorithm and game theory mechanism to improve the performance.

The vehicles are location aided and every vehicle knows about neighbors' information with the help of the hello packets. The interval of hello packets is set to be 1 second. The major responsibility of the CH is to distribute the data received from RSU. An RSU delivers its payoffs only to a CH. Based on the channel condition with the neighbors, neighboring degree, and the relative motion of the neighboring vehicles, the CHs are elected by implementing fuzzy logic. After receiving a hello packet from a node $m$, the mobility factor of a node $s$ is determined with the following equation:

$$MF\left(s,m\right)=\frac{\left|v\left(m\right)\right|-\min\limits_{y\in N_S}\left|v(y)\right|}{\max\limits_{y\in N_S}\left|v(y)\right|} \qquad (20)$$

where the set of neighboring vehicles is denoted with $N_s$ for a node $s$, $m$ represents the hello packet receiving node, and $v$ denotes the velocity. The CHs intends to deliver the payoffs received from the RSU in a multihop manner to the destination. To accomplish this goal, RGVR forms a coalition game based on the collision probability. The multihop decision is taken based on a Q-learning technique in RGVR. The Q-table is maintained by each RSU. Each entry in the Q-table represents a value for taking an intermediary node to reach the RSU. The Q-values are updated with hello packets. Q-values are attached inside the hello packet.

Advantages: To form the clusters, the velocity of the vehicles is considered. This mechanism will increase the stability of the clusters. Besides, the topology changes of the network will be also minimized. The clustering process does not involve the exchange of extra control packets. Thus, the amount of control overhead will be minimized in the RGVR.

Disadvantages: The optimization of multi-hop routing is conducted from the transport layer and the MAC layer perspectives. As the main drawback, the Q-value mechanism consumes a lot of control packets. In RGVR, the Q-learning mechanism is used twice but no performance evaluation for routing overhead is given.

Application: The protocol is mainly focused on data dissemination among vehicles and RSUs. A good infrastructure environment is necessary to implement the algorithm in a real-life scenario.

#### 5) REINFORCEMENT ROUTING IN SOFTWARE DEFINED VEHICULAR ROUTING (RL-SDVN)

Nahar *et al.* proposed RL-SDVN [75], an SDN based routing protocol for VANET architecture. In RL-SDVN, vehicles are grouped into clusters and assist each other within a cluster to find out the optimal route. RL-SDVN mostly focuses on the optimal clustering process. In order to do so, the authors used the Gaussian mixture model (GMM) and RL techniques together to predict a vehicle's mobility pattern such as speed and direction. To derive the features and fitness values, a classifier is designed. The packet forwarding decision is handled by the Q-values. A unique traffic flow model is introduced in this paper. The traffic flow model is constituted with the vehicle density, speed, and direction considering space and time. In RL-SDVN, the anomaly of vehicle movement is derived by the second-order differentiation of the displacement of the vehicles. Every vehicle transmits a safety message in every 100–300 seconds. This message contains information such as the current location of the vehicles, upcoming traffic signals, direction-changing information, and road condition. With the help of the GMM procedure, the clusters are formed by using a probability distribution. In the GMM procedure, a vehicle is selected for an arbitrary cluster. Then, based on the expected maximization procedure, the vehicle is assigned to every cluster and the values are examined.

The self-learning mechanism utilizes the information from the beacon message received every 100 ms. An adjacency matrix is used to determine the number of neighboring nodes. Each vehicle is enabled with dedicated short-range communication (DSRC) system. And the last parameter used is known as Queue occupancy. To be elected as CH, a node should have a high neighborhood and a low number of packets in the queue. After forming the clusters, the routing process begins. The SDN controller derives the optimal route based on the location information. The learning process takes place in every intermediary hop of the journey, from the source to the destination. The SDN controller runs the Q-learning mechanism to compute the route based on the stored information in the vehicles. The vehicles store information up to two-hop neighbors. Before sending a packet to the destination, a node checks the Q-value inside the packet. If a vehicle is able to forward the packet to the next-hop destination, a positive reward is given. On the other hand, a negative reward is added if no path is available. To compute the Q-value, the distance with the destination and the delay is taken into designing consideration.

Advantages: RL-SDVN is heavily dependent on the clustering process. The probabilistic clustering process can be tuned and a near-optimal solution can be accepted anytime based on the requirement. In such cases, tradeoffs can be done based on the available bandwidth of the links. However, this measurement is not taken into consideration in the routing protocol.

Disadvantages: The clustering mechanism will work best in a centralized manner and the SDN's controller mechanism also supports such architecture. However, RL-SDVN is designed to work in a distributed manner, which will increase the number of the exchanged control packets. The 100 ms timer for receiving a beacon packet will consume a high amount of bandwidth.

Application: Designing the SDN controller is the key factor of the application area of this routing algorithm. In several studies, a flying unit is formulated as the RSU. The normal designing factor considers the RSU as the local controller. Hence, considering the flying unit as the RSU will enable this routing protocol to be implemented in an infrastructure-less environment; otherwise, this protocol is only applicable to the urban area.

### E. SECURITY-BASED ROUTING PROTOCOLS

These routing algorithms offer secure data communication between nodes. The trust score evaluation of vehicles before sending and receiving a message is one of the popular security features of the routing protocols [76].

#### 1) TRUST-BASED DEEP RL-BASED VANET ROUTING PROTOCOL (TDDRL)

Zhang *et al.* proposed a DRL-aided trust-based VANET routing algorithm TDDRL, which is designed for the SDN network paradigm [77]. The SDN paradigm used in TDDRL is logically centralized, meaning that the data from one layer of this SDN architecture are abstracted from those of other layers. In TDDRL, the SDN controllers are used to learn the optimal path for routing. These controllers implement a deep neural network (DNN) to learn the optimal path from the source to the destination. The security feature is implemented by utilizing the trust score to choose the neighboring node for selection as a next-hop member. To formulate the DRL problem, TDRRL assumes that the network infrastructure has an SDN architecture environment and the control layer is the agent. This routing protocol assumes that the combination of the location and forwarding ratio of the vehicle serves as the state for the DRL mechanism. The state transition probability is given as follows:

$$p\left(s_t|s_{t+1}\right) = \prod_{n=0}^{N} p_n^{m_n m'_n} m_n, \quad m'_n \in 1, 2, \ldots, k, \tag{21}$$

where $p\left(s_t|s_{t+1}\right)$ is the probability of transition $p$ from state $s_t$ to the next state $s_{t+1}$. $p_n^{m_n m'_n}$ denotes the probability that a vehicle $n$ changes state from $m$ to $m'$ . The action is defined as the forwarding capability of the vehicle to any other vehicle in its vicinity. The trust value of a vehicle is considered to be the reward of the formulated DRL problem in TDDRL. Trust is computed using the following equation:

$$V_{ij}\left(t\right) = \varphi_1 VT_{ij}^{C}\left(t\right) + \varphi_2 VT_{ij}^{D}\left(t\right), \tag{22}$$

where the trust value of a vehicle is denoted as $V_{ij}$ . $V_{ij}$ is calculated based on the trust value acquired from the control packet $VT_{ij}^{C}$ from node $i$ to node $j$, and the trust value acquired from the data packet is denoted as $VT_{ij}^{D}$. $\varphi_1$ and $\varphi_2$ are the weighting factors used to derive the trust value. The control packets used in this protocol are kept the same as those in the AODV routing protocol, which includes RREQ, RREP, and RRER messages.

Advantages: The utilization of DQN in the VANET scenario will solve the state space-related problems that arise with Q learning approaches.

Disadvantages: The TDDRL assumes that the trust value of the sender vehicle will always be 1. This is a major security flaw that renders the horizon susceptible to receiving malicious messages from an intruder vehicle. Only the forwarding ratio is considered as the vehicle's trust value, However, justification of the inability of the node to change the forwarding ratio is not provided.

Application: SDN requires a continuous connection with the controller to forward the packet. Therefore, TDDRL is not functional in an infrastructure-less environment.

#### 2) SECURED VANET ROUTING WITH BLOCKCHAIN (SVRB)

Dai *et al.* presented SVRB [78], a secured routing protocol for VANET architecture. In SVRB, each vehicle is equipped with a trust evaluation technique. Blockchain technology is applied to prevent information manipulation in the transmitter end. The RL algorithm decides whether to choose a vehicle as the next hop or not after the evaluation of the trust score. To ensure fast convergence in the case of a new entry to the network, a hot booting technique is also applied. Originally this protocol is designed by taking the highway environment in mind. SVRB conceptualizes the arrival of communication request based on the poison distribution. A vehicle tries to deliver the data directly to the RSU. In case of the unavailability of the RSU, an OBU opts for multi-hop transmission to deliver the data to the RSU. SVRB gives protection against three kinds of attacks. They are eavesdropping, jamming, and spoofing. The malicious node creates such a wireless signal, which creates noise and jams the original signal. A vehicle can act as a real relay node and simply drop the message after receiving it. In the worst case, a relay that is intended to deliver the message, may not deliver at all.

With the help of beacon messages, a vehicle determines the channel gain of the neighboring vehicle. If the channel gain is within a threshold, SVRB considers the connection between the corresponding nodes as successful. High trust value is assigned to a vehicle that tends to relay the message of a node with high trust and does not intends to transmit the message of a vehicle with low trust. According to the blockchain mechanism, every vehicle forms a block that monitors the neighboring vehicles' activities and stores them in memory. A Merkle tree contains the trust values in the form of the hash values in the form of the leaves. Upon the creation of a new block, a vehicle informs the information to the other vehicles. Thus, each vehicle assists other vehicles in the trust management procedure. Before forming the chain, a block needs to be verified by the majority of the users. When the consistency of the trust does not match with previous blocks, the block is simply dropped.

Advantages: SVRB is a truly distributed secured routing protocol, which formulates the vehicles as the block to enable the blockchain technology. Compared to the SDN-based security protocol, this protocol does not depend on the RSU or any other third-party trust management system to enable the security for data dissemination.

Disadvantages: The routing protocol does not try to improve other routing performances such as throughput and PDR. Adding such capacity before data transmission will be good for future implementation.

Application: The routing protocol is truly distributed and will be applicable to most of the scenarios. However, it should be noted that the link quality, mobility, and direction are not taken into design consideration.

### 3) RL-BASED ANTI-JAMMING VANET ROUTING PROTOCOL (RAVR)

Xiao *et al.* proposed RAVR [79], a routing protocol to enable protection against jamming for VANET architecture. The architecture is aided with a UAV, which works as the RL-agent to take the right action to protect the data packet from malicious nodes. The jammer or malicious node is assumed to have smart power management capability, to effectively jam the transmission of the UAV. The UAV receives data from the vehicles and acts as a relay to deliver the message to the right RSU. A game is formulated between the UAV and the jammer to take the routing decision of the OBU's message. By finding out the Nash equilibria of the game, the optimal strategy of the relay strategy is selected. Policy hill climbing (PHC) based solution is given to take and change the relay strategy adaptively. The PHC strategy does not require any jamming or channel model to take optimal routing strategy.

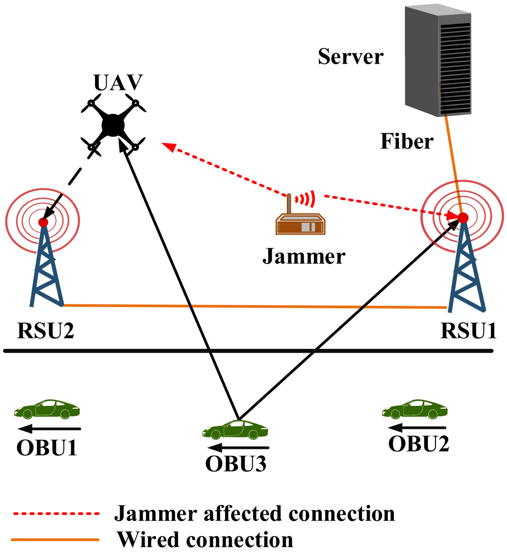

FIGURE 11. Communication mechanism in RAVR.

As shown in Fig. 11, the OBUs try to send the data to the server. OBU3 first uploads the message to the RSU1. The same message is also received by the UAV. The UAV designs the game and makes the decision whether to relay the message to RSU2 or not. Based on the bit error rate of the received message from RSU1 to the UAV and from OBU3 to the UAV, the UAV decides whether to relay the message or not. However, the jammer is equipped with a smart jamming mechanism and can tune to the control frequency of the OBUs' transmission. The interactions between the jammer and the UAV are the anti-jamming game and the final decision is made based on the received message quality such as SINR and BER. The jamming action taken by the jammer can be modeled as the MDP model. The PHC based hot booting technique is used to initialize the relay strategy.

Advantages: Implementation of the policy-based RL solution enables the UAV to take secured routing decisions. The UAV does not need to have any knowledge about the prior jamming model, and a hot booting model is used to initialize the system with a sub-optimal solution.

Disadvantages: Introducing special equipment (UAV) to enable routing security might not be a feasible option for implementation. The packets might be directly delivered to the RSU2 in a multihop manner with other OBUs, instead of delivering to RSU1, which still raise the security vulnerability.

Application: UAVs and RSUs are mandatory to implement RAVR. Thus, the application scenario is limited to URBAN areas only. This routing protocol can be applied on top of a general routing protocol where the security is more important.

### 4) SOFTWARE-DEFINED TRUST-BASED DEEP REINFORCEMENT LEARNING ROUTING PROTOCOL (TDRL-RP)

Zhang *et al.* proposed TDRL-RP [80], a DRL assisted secured routing protocol for VANETs. The algorithm exploits the convolution neural network (CNN) in the SDN controller in order to find out the most suitable routing path. A trust model is proposed to evaluate the neighboring behavior before the routing decision is made. While selecting a vehicle as a neighbor, the trust value is also taken into consideration along with the speed and direction of a vehicle. In this routing algorithm, the network infrastructures are taken as the environment and the controllers work as the agent for the DRL mechanism. The DRL technique is used to discover the path from the source to the destination and to make the packet disseminating decision as well. The DRL agent needs a vehicle's current position and the delivery ratio for the route decision making process.

TDRL-RP forms two distinct matrices consisting of the vehicle's position and forwarding ratio. The action of the controller is to pick the right vehicle for data transmission. The reward is given based on the trust value of the selected neighbor. The trust value is updated after a specific time limit. By this mechanism, the trust value of a node keeps changing and a trusted vehicle can become an untrusted vehicle and an untrusted vehicle can become a trusted vehicle based on the behavior. The control packets are used to determine the trust value of the vehicles. The trust value can be computed based on the following equation:

$$NT_{ij}\left(t\right) = \omega_1 CT_{ij}\left(t\right) + \omega_2 DT_{ij}(t) \tag{23}$$

where the direct trust of the control packets is indicated with $CT_{ij}$, the trust with the data packet is indicated with $DT_{ij}$ between the vehicles $i$ and $j$, and $\omega_1$ and $\omega_2$ are the normalizing weighting factors. In TDRL-RP, the values of the weighting factors are kept equal. Based on the link condition, a source discovers the path, and the trust value of the computed path is checked by the centralized controller. The rectifier nonlinearity activation (ReLU) is used as the activation function in the DQN.

Advantages: The centralized mechanism enables the controller to have an eagle's eye view of the entire topology. This mechanism will help the vehicle to take the best route decision.

Disadvantages: The route discovery process is kept as similar to the reactive protocols such as AODV. However, if the vehicles are connected with the centralized server, only giving the routing information to the controller would be enough. The centralized server would have computed the route and send back to the vehicle. By computing the route with the central server would save a good amount of bandwidth.

Application: There is no recovery process involved, and the proposed protocol is based on the centralized controller. To enable the security feature of TDRL-RP, the infrastructure is a must to present. However, the niche algorithm of TDRL-RP is AODV, which can be operated in a distributed manner.

### 5) BLOCKCHAIN AND RL BASED VANET ROUTING PROTOCOL (BRL-RP)

Zhang *et al.* proposed BRL-RP [81], a secured routing protocol for VANET architecture. The security feature in BRL-RP is implemented via cutting edge blockchain technology. The authors stated that SDN technology can increase the security of VANET architecture vastly but, due to the less infrastructure in the roadside region, a VANET suffers from security threats. An optimization problem is developed concerning the trust features, computational capability, and the degree of consensus node. The optimization model mimics the famous MDP model. With the help of dueling DQL (DDQL), the optimization problem is solved. According to the DDQL, the vehicles deliver the trust scores to the area controller, and the area controller delivers the message to the domain controller. The blockchain is interfaced with the domain control layer, and the proposed consensus protocol is liable for information collection and synchronization among the different controllers. The entire architecture can be divided into the three layers of device, area, and domain. The area controller collects the data from the device controller. The domain controller interacts with the blockchain services, and the trust values are sent back to the vehicles again. The training procedure in the controllers is a continuous process, and the throughput gets increased with time.

To compute the trust, the previous interactions among the vehicles are considered. The vehicles assess the trust of the intermediary hop by the data sending behavior of the vehicles. A data packet used in the BRL-RP uses the sequence number to justify the lifetime of the packet. The header of a data packet includes neighbors' ID, vehicles' position, velocity, available throughput, trust value, last sequence number, and packet buffer. The neighboring table is formulated based on the received hello messages, and the trust values are stored for a corresponding neighboring node. Direct trust that is the trust attained by the direct interaction with the neighbor is derived using the following equation:

$$T_{v_b v_b'}(t) = \frac{f^C_{v_b v_b'}\left(t\right)}{f_{v_b v_b'}\left(t\right)}, \quad t \leq W \tag{24}$$

where direct trust is expressed with $T_{v_b v_b'}$ for a vehicle $v_b$ for its neighbor $v_{b'}$, the number of totals sent packet at time $t$ is denoted with $f_{v_b v_b'}$, and the number of packets which are forwarded correctly is denoted with $f^C_{v_b v_b'}$. The correctness of a packet is judged based on the sequence number within a time window $t$ that must be equal or smaller to the window threshold $W$. A packet that is forwarded properly from a sender increases its direct trust value. The trust value of a node varies from 0 to 1, where 0 means malicious node and 1 means fully trusted node.

Advantages: Blockchain is creating a new era of security enhancement. Including blockchain in VANETs will ensure secure data delivery. SDN has a global view of the network, so implementing such security in the SDN controller will not create any extra burden on the vehicles. Local trust value computation will also benefit when the controllers will not present.

Disadvantages: Even though blockchain is a distributed mechanism and removes the problem of central dependency, the integration of the SDN mechanism brought back the problem of central dependency.

Application: A good infrastructure area where enough endpoints will be present to act as the area controller will only be suitable to run this routing protocol.

### *F. DTN-BASED ROUTING PROTOCOLS*

In delay-tolerant networking (DTN), the connectivity from the source to the destination is not ensured. In VANETs, the SCF mechanism is used to design a routing protocol for the DTN scenario [82]. Road segments with a small number of vehicles or RSUs raise such a situation in VANETs.

#### 1) Q-LEARNING BASED VANET DELAY TOLERANT ROUTING PROTOCOL (QVDRP)

Wu *et al.* proposed a Q-learning based delay-tolerant routing protocol called QVDRP for VANETs [83]. This routing protocol is especially applicable for delivering VANET's data from the source to the cloud destination through multiple gateways. The routing technique implements a position prediction technique for the availability of the destination. This enables QVDRP to adopt an adaptive data duplication technique. QVDRP utilizes RSU as gateways to communicate with the cloud servers. Thus, this routing algorithm differs from the traditional VANET routing algorithms and it aims to only disseminate the vehicle-generated data to the RSUs. While keeping the delay under a threshold level, the QVDRP tries to maximize the packet delivery probability. The authors in [66] argued that the generated data in VANETs are incomplete due to the fragile communication among the vehicles and thus Q-learning is suitable for routing. QVDRP assumes the network as the environment and the vehicles as the agents. The learning process involves exchanging data with other nodes in the network. Next-hop selection works as the action for the Q-learning agent in QVDRP. Each node maintains a Q-table where the Q-values of other nodes are stored. Updating Q-table in the learning process is an important task. This routing algorithm adopts two different approaches to complete the task. In the case of connectivity, the nodes exchange periodic hello messages to update the Q-table whereas, in the case of a neighbor-less situation, the Q-table is updated after every 10 minutes. Like most of the routing protocols, reward 1 is given if the sending vehicle is directly connected to the destination vehicle. If a node gets to hear from a node before a threshold time, the nodes get a discounted positive reward; otherwise, the Q-value is set to the default value of 0.75. Encounter probability uses the inbound and outbound direction prediction technique for each road segment. This encounter probability plays an important role to reduce packet duplications.

Advantages: QVDRP tries to minimize the number of duplicate copies, which is a mandatory characteristic of delay-tolerant protocols. To implement these features, QVDRP considers the Q-value and the relative velocity.

Disadvantages: This algorithm follows a greedy state selection technique based on the current Q-values stored in the node. For this reason, the algorithm might converge into local optima.

Application: This routing protocol is especially applicable for post-disaster areas. In such areas, the network infrastructures usually get destroyed. So, delay-tolerant routing protocols like QVDRP can play a good role to collect data for future uses.

## IV. COMPARISON

In this section, we present three comparison tables for the investigated routing protocols from different perspectives. A critical analysis and discussion of each table are also presented.

### *A. KEY FEATURES OF RL-BASED ROUTING PROTOCOLS*

Table 1 presents the key features of the reviewed articles. The particular properties that are highlighted in this table are the main performance-controlling features of the routing protocols. According to the special feature of the QTAR algorithm, we can infer that with the increment of time, the performance of the routing algorithm improves. After a specific time during which learning is completed, QTAR begins to perform better than the underlying geographic routing algorithms. RHR uses modified and improved hello packet structures; this facilitates the acquisition of information about the available links. However, extra information requires extra bandwidth. RHR should adopt a special broadcasting technique to conserve as much bandwidth as possible. Communication channel measurement and consideration of the vehicle's direction lead to a positive impact of PFQ-AODV on the performance metrics. QGRID uses historical data based on taxies in Shanghai. Since the implementation is offline, the routing algorithm will have a pre-converged condition; thus, an initial learning time is not necessary. For a specific region, an offline learning algorithm is ready to launch beforehand. ALCA implements hierarchical routing by forming clusters among the vehicles, which reduces the state-space size. Multiple parameters are chosen for the selection of the next hop in the PP-AODV algorithm. This ensures the minimum standard for all performance metrics related to the parameters. The greedy forwarding technique used in RLZRP increases the probability of successful packet transmission in the case of the breakage of the pre-calculated route. The ARPRL algorithm ensures that there are no broadcast storms, owing to the innovative features mentioned in Table 1. Parameter dueling ensures that TDRRL chooses the appropriate next state, which is an outcome of the innovative idea presented in Table 1.

Hierarchical routing algorithms tend to form clusters among the vehicles and the CHs are selected as the agent mostly. RLRC and RL-SDVN both try to elongate the lifetime of the clusters in different ways, mentioned in Table 1. Increasing the clusters' lifetime will reduce the number

**TABLE 1.** Summary of key features of RL-based routing protocols.

| Protocol | Innovative idea |
|---|---|
| QTAR [60] | Utilizes the Q-learning technique separately for RSU and the vehicle. Modifies geographic routing algorithm and proposed Q-greedy geographical forwarding algorithm (QGGF) to learn traffic conditions of each intersection. |
| RHR [54] | Uses packet-carry-on information to keep the routing table updated. Implements a two-level routing table. |
| PFQ-AODV [65] | Implements a communication channel measurement technique based on bandwidth, the quality of the link, and the movement criteria of the vehicles along with the direction. |
| QGRID [55] | Historical data are used to train the model generated by taxis in Shanghai. Optimizes grid forwarding technique using Q-learning and greedy forwarding within the grid |
| ALCA [71] | Uses four different agent types: RLA, DUA, ZSA, and SCA. Utilizes learning mechanisms to form optimal clusters between the vehicles. |
| PP-AODV [64] | Bandwidth, delay, and packet collision probability are taken as the selection parameters of the RL algorithm for choosing an intermediary hop to the destination. |
| RLZRP [56] | Facilitates layer-two switch table learning mechanism in routing table implementation technique. Utilizes greedy forwarding technique in case of link disconnection |
| ARPRL [66] | Optimizes loop avoiding mechanism by updating the structure of the hello packet. |
| TDRRL [77] | Implements dueling deep Q-network based on the Q value of the static states. |
| ADOPEL [57] | The reward function is designed considering the aggregable packets. |
| RSAR [2] | Considers link status by applying traffic rules, multiple intersections, multiple lanes, variable speed, and direction of the vehicles. |
| PbQR [61] | Maintains the size of the Q-table, which contains only one-hop neighbor information, by applying a greedy selection policy PbQR. |
| QVDRP [83] | Applies data duplication technique based on the prediction of the availability of the destination node. |
| VRDRT [58] | The optimal neighbor selection process is performed in multiple equal slices of time using a prediction technique, which ensures adaptability with rapidly changeable traffic. |
| QLAODV [67] | Proposes route altering mechanism to reduce the number of route discovery phases in a highly dynamic scenario. |
| CURV [72] | Implements a broadcast mitigation technique with context-aware routing. |
| RLRC [73] | Consideration of average speed is adopted to form the clusters in RLRC; for this reason, the lifetime of the clusters will be higher. |
| RL-SDVN [75] | Applied GMM based probabilistic clustering mechanism with a CH selection criteria based on distance and queuing size parameters. |
| HQVR [69] | Reduces the convergence time by storing feedback information inside all the intermediary nodes |
| Q-LBR [62] | Establishes load-balancing multipath routing based on queue size |
| PIRP [68] | Implements a modulation coding scheme learning technique to evaluate the link condition |
| RGVR [74] | A game-theoretic approach to establish collision-free transmission |
| TDRL-RP [80] | Centralized trust management system to enable secured transmission |
| BRL-RP [81] | Implementation of controller distributed security with blockchain technology considering delay and throughput |
| SVRB [78] | A true distributed routing protocol where the blocks are formed by the vehicles |
| RAVR [79] | Implementation of UAV based security by forming a game with the jammer and RL-based learning technique. |

of exchanged control packets. The backpropagation of the reward value is one of the main challenges in RL-based routing protocol. HQVR tries to optimize the reward propagation by storing the values inside the intermediary vehicles. RL has the potentials to establish an effective multiple QoS routing mechanism. However, only Q-LBR explicitly implemented such technology. SDN-based architecture is mostly centralized. The SDN-based secured protocols have the potentials to give superior security mechanisms. BRL-RP and TDRL-RP are such protocols where the security is maintained centrally. However, in such a protocol, the infrastructure should be ensured. The blockchain-based solutions have the mechanisms to implement a distributed trust management system. Such a technique is adopted in SVRB. RAVR implements the security mechanism with a game-theoretic approach, and the agent is the UAV. For a tactical region, this protocol has the potentials to serve the military needs.

### B. APPLIED OPTIMIZATION CRITERIA AND ADOPTED TECHNIQUEs

The investigated RL-based VANET routing protocols are intended to optimize the performance from different perspectives. Given that the optimization criteria have trade-offs, a routing protocol should attempt to maximize the outcome of the expected performance metrics while also minimizing the negative impact on other performance metrics. In Table 2, the intended optimization criteria are highlighted. They are also described in detail in this subsection.

EED optimization of a routing protocol ensures message delivery from a source node to a destination node in the minimum time [84]. From Table 2, it is evident that QTAR, RHR, PP-AODV, and RLZRP protocols have adopted special techniques to optimize the EED performance metrics. However, the optimization of EED depends on the total number of links and the total delay of the network. The quality of a particular link depends on its availability, longevity, and bandwidth. The various types of delays include propagation, queuing, and internal processing delays. Thus, the EED can be optimized based on any of the variables on which it is dependent. The

**TABLE 2.** Comparison of Optimization criteria and techniques used in RL-based routing protocols.

| Protocol | RA VR | SV RB | BR L-RP | TDR L-RP | RG VR | PI RP | Q-LB R | HQ VR | RL-SD VN | RL RC | QT AR | RHR | PFQ-AODV | QGR ID | AL CA | PP-AODV | ARP RL | TDR RL | RLZ RP | ADOP EL | RS AR | Pb QR | QVD RP | VRD RT | QLAO DV | CR UV |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| End-to-end delay consideration | × | × | ✓ | × | ✓ | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | × | ✓ | × | × | ✓ | ✓ | × | × | × | ✓ | × | × |
| Connection reliability | ✓ | × | × | × | × | ✓ | × | ✓ | × | ✓ | ✓ | ✓ | ✓ | × | × | × | × | × | ✓ | ✓ | ✓ | ✓ | × | × | ✓ | ✓ |
| Store–carry–forward scheme | × | × | × | × | × | × | × | × | × | × | ✓ | × | × | × | × | × | × | × | ✓ | × | × | × | ✓ | × | × | × |
| Extra consideration for intersections | × | × | × | × | × | × | × | × | × | × | ✓ | × | × | × | ✓ | × | × | × | × | × | ✓ | × | ✓ | ✓ | × | × |
| Broadcast mitigation technique | × | × | × | × | × | ✓ | ✓ | × | × | × | × | ✓ | × | ✓ | × | × | × | × | × | × | × | × | × | × | × | ✓ |
| Dependency on RSU | ✓ | × | ✓ | ✓ | ✓ | × | ✓ | × | ✓ | × | ✓ | × | × | × | ✓ | × | × | ✓ | × | ✓ | × | × | ✓ | ✓ | × | × |
| Freshest path consideration | × | × | × | ✓ | ✓ | ✓ | × | ✓ | × | × | × | ✓ | × | × | × | × | ✓ | × | × | × | ✓ | × | × | × | ✓ | ✓ |
| Vehicle's position prediction | ✓ | ✓ | × | × | × | × | × | × | ✓ | × | × | ✓ | ✓ | × | × | ✓ | × | × | × | × | × | × | ✓ | ✓ | × | × |
| Vehicle direction consideration | × | ✓ | × | ✓ | ✓ | ✓ | × | × | × | × | × | × | ✓ | ✓ | ✓ | × | ✓ | × | × | ✓ | ✓ | ✓ | ✓ | × | ✓ | ✓ |
| Offline learning based on historical data | × | × | ✓ | ✓ | × | × | × | × | × | × | × | × | × | ✓ | × | × | × | × | × | × | × | × | × | ✓ | × | × |
| Security feature | ✓ | ✓ | ✓ | ✓ | × | × | × | × | × | × | × | × | × | × | ✓ | × | × | ✓ | × | × | × | × | × | × | × | × |
| Mobility awareness | × | × | × | × | × | ✓ | × | × | ✓ | ✓ | × | × | × | × | × | × | ✓ | × | × | × | × | × | × | × | × | × |
| Quick convergence mechanism | ✓ | ✓ | × | × | × | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | × | × | × | × | ✓ | × | × | × | × | × | × | × | × | × |
| Routing loop avoidance technique | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | ✓ | × | × | × | × | × | × | × | × | × |
| Traffic light consideration | × | × | × | × | × | ✓ | × | × | × | × | × | × | × | × | × | × | × | × | × | ✓ | × | ✓ | × | ✓ | × | × |
| Recovery policy | × | × | × | × | × | ✓ | ✓ | × | × | × | × | × | × | × | × | × | × | × | × | ✓ | ✓ | × | × | × | × | × |
| Node degree evaluation | × | × | × | × | ✓ | ✓ | × | × | ✓ | × | × | × | × | × | × | × | × | × | × | × | × | ✓ | × | ✓ | × | ✓ |
| Multipath | × | × | × | × | × | ✓ | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × |
| Load balancing | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × | × |

routing protocols tend to optimize these variables to obtain a better EED outcome.

The identification of reliable connections will have positive impacts on almost every performance metric [85]. QTAR, RHR, PFQ-AODV, RAVR, PIRP, HQVR, RLRC, and RLZRP routing protocols have adopted techniques to obtain reliable connections before transmitting data via a link. This optimization technique is mainly focused on the rewarding mechanism of RL algorithms. However, the identification of a reliable connection can result in the use of more control packets. These phenomena lead to a poor result for the control packet overhead of the routing protocols. However, a reliable connection can be accessed by multiple nodes for the transmission of their data towards the destination [86]. This will increase link-sharing among multiple nodes, and consequently, the propagation delay will increase. This can also lead to data collision among the transmitted data at the receiver end.

SCF techniques are used as recovery processes in the routing algorithm of the VANET architecture. In the case of the unavailability of the next-hop node for transmitting data, the host vehicle carries the information for a limited time before discarding the data [87]. Even though SCF techniques are used in QTAR and RLZRP routing algorithms, the quality of the SCF mechanisms is not optimized. The SCF mechanism can be improved by implementing an adaptive TTL for the message system, considering the destination, and utilizing the vehicle's direction and position. A good SCF mechanism will have a positive impact on the PDR performance metric.

Road-intersections play a vital role in VANET architecture, and by considering the intersections, the performance of the routing algorithm can be improved to a significant level.

Usually, the vehicle density is relatively higher at intersections [88]. The possibility of interference and network congestion also increases at intersections. Moreover, the chance of NLOS increases when the carrying vehicle transmits a message to a vehicle in another segment. This NLOS problem can be solved if the vehicle gives preference to the vehicle near the intersection. Among the reviewed routing protocols, QTAR and ALCA consider the intersection, especially in terms of the routing decision.

The broadcast storm is a trivial problem in VANET architecture. Reactive routing protocols frequently discover a routing path based on hello packet broadcasting [89]. Thus, the probability of a broadcast storm is higher in such routing protocols. Surprisingly, among the discussed routing protocols, only RHR, PIRP, Q-LBR, and QGRID have explicitly adopted broadcast-storm mitigation techniques. The broadcast storm results in high usage of the bandwidth of the network, which leads to the poor performance of the routing algorithm.

The existence of RSUs is logical only if the application scenario is an urban area. SDN-based routing protocols such as TDRRL depend on the RSUs, which can be a bottleneck of the routing algorithms [90]. In addition to considering RSUs, a superior routing algorithm should adopt recovery policies in the case of the unavailability of the RSUs. Among the discussed routing protocols, QTAR, ALCA, RAVR, BRL-RP, TDRL-RP, RGVR, Q-LBR, RL-SDVN, and TDRRL algorithms utilize the RSU and are thus meant only for urban implementation.

The freshest path consideration involves checking for a new path and examining the preexisting paths after a specific time interval [91]. This technique improves the quality of PDR, EED, and throughput but also increases the chance of control packet usage [92]. The utilization of the control packet should be maintained and kept below a threshold value to avoid any negative impact on the available bandwidth for data transmission. Among the proposed routing algorithms TDRL-RP, RGVR, PIRP, HQVR, and RHR consider the freshest path before data transmission.

For successful data transmission, vehicle position prediction is important for both sparse and dense conditions [93]. In the case of this optimization criterion, the vehicles have prior knowledge of the future position of the destination node as well as the intermediary next hop. As a result, the chance of data failure is significantly reduced. PFQ-AODV, QGRID, SVRB, TDRL-RP, RGVR, PIRP, and ALCA algorithms implement these techniques.

Among the algorithms reviewed in this survey RAVR, SVRB, BRL-RP, TDRL-RP, ALCA, and TDRRL have security features. These protocols mostly implement trust-based security features [94]. In addition to the aforementioned routing protocols, some other data dissemination mechanisms have been proposed for VANETs, which mostly focus on RL-aided blockchain-based solutions. Blockchain technology has opened a new horizon to implement the distributed trust management system in VANET architecture. SDN-based security might ensure a superior trust management system with the help of third-party services but, for an infrastructure-less scenario, the study of distributed architecture is more important.

For a general-purpose VANET routing, the consideration of mobility variation, routing loop avoidance, node degree evaluation, and multipath routing are important factors. The multipath routing and load balancing mechanism will reduce the number of route discovery process initiations. The route discovery process is one of the major reasons for increasing routing overhead. The RL algorithm has the potentials to implement an efficient multipath routing algorithm, as multiple routes are being evaluated before selecting the best route. However, only PIRP implemented such a routing mechanism.

The QoS-based routing protocols increases the chance of a routing loop. None of the routing protocols except ARPRL have adopted the loop avoidance technique. Ensuring QoS is the main goal behind implementing the RL algorithm in the routing protocol. Thus, RL-based VANET routing algorithms should care for this problem, and the performance evaluation should also reflect this optimization.

### C. PERFORMANCE EVALUATION TECHNIQUES

Table 3 lists the simulation-related parameters used in the protocols investigated in this research.

From this table, it is evident that most of the protocols use well-known simulators, including NS-2, NS-3, QualNet, and OPNET [95]. Among them, QualNet and OPNET are available as paid versions only whereas NS-2 and NS-3 are freely available. Topology refers to the street layout used for the simulations.

The topology is one of the most important factors in VANET simulation [96]. The simplest topologies are grid-based ones for which the road segments intersect with each other, and the length of the segments is mostly fixed. Realistic topologies include the geographical position of the snippet from a real-world map. An open-street map is a type of geographic information system (GIS) wherein a real-life road topology can be generated [97]. In the discussed protocols, QTAR uses random and grid point topologies, whereas RHR uses OpenStreetMap for topology generation. PFQ-AODV uses the Midtown Manhattan map. QGRID uses data generated in Shanghai city, and QualNet uses the Manhattan grid scenario. The Manhattan street grid [98] is an imaginary road topology created by the Greenwich village. It consists of 155 cross streets. However, ALCA, PP-AODV, RLZRP, and TDRRL did not employ any road topology. Real-life geolocation-based simulations are more practical, and the output can also be mapped to real-life locations. Q- LBR used the riverbed modeler from the OPNET simulator. This is a well-accepted model not only in literature but also for the industry.

One of the main differences between WSN and VANET is mobility. In the case of simulations, mobility generation is an important task that is often difficult [99]. SUMO is an excellent vehicle mobility generator that is used in several studies on VANET. Among the compared protocols, RHR

**TABLE 3.** Comparison of performance evaluation parameters and techniques.

| Protocol | Simulator | Topology | Mobility | Simulation area (m*m) | Simulation time (s) | Velocity (m/s) | Number of vehicles | Communication range (m) | Compared routing protocols | MAC protocol | Propagation model | Payload (Bytes) | $\alpha$ | $\gamma$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RAVR | – | – | – | – | – | 4-18 | – | – | Q-learning and hot booting PHC technique | – | – | – | – | – |
| SVRB | – | – | – | – | – | 16.67 - 25 | 8 | – | Varied own parameter | 802.11p | – | – | 0.7 | 0.5 |
| BRL-RP | OPNET | Random and Grid | – | 5000 * 5000 | | Static | 32 | – | Varied own paramaters | 802.11p | – | 1024 | 4 MHz | 0.05 MHz |
| TDRL-RP | OPNET | Random and Grid | – | 5000 * 5000 | 900 | Static | Aug-32 | – | AODV, and own paremters | 802.11a | – | 1024 | – | – |
| RGVR | NS-2 | Free way | SUMO and TraNS | 1700 * 1700 | 1200 | 27.7778 | 400 | 250 | CDS-SVB, AODV-ETX | IEEE 802.11p | Nakagami Model | 1000 | 0.7 | 0.9 |
| PIRP | NS-2 | Four lanes | [95] | 1500 * 1500 | 600 | [95] | 10 | 40 (test-bed) | AODV-ETX, HILAR | IEEE 802.11p | – | 1024 | – | – |
| Q-LBR | OPNET | Riverbed modeler layout | Random waypoint | 1000 * 1000 | – | – | 11 | – | U2RV, and own parameters | IEEE 802.11 | Urban Propagation Model | 256 - 1000 | 0.3 | 0.7 |
| HQVR | NS-2 | Two-way lane | SUMO | 1800 * 1800 | 200 | 5-25 | 50 - 250 | 250 | GPSR, QLAODV | IEEE 802.11 | – | 512 | – | – |
| RL-SDVN | NS-3 | Oslo city, Norway | SUMO | 5000 * 5000 | 150 | 0-20 | 25-200 | 100-1000 | CPB, DMCA, M.Ren, SCF | 802.11p | – | 512 | – | – |
| RLRC | Python | Grid | – | 2000 * 1200 | – | – | 100 – 300 | 300 | QLAODV, AODV | – | – | – | 0.1 | 0.9 |
| QTAR | QualNet | Grid | VANET MobiSim | 3040 * 3040 | – | 5–35 | 50–500 | – | RTAR, iCar-II, GyTAR, GPSR, LAR | 802.11p | Street Microcell/LOS | – | 0.1–1 | 0.1–1 |
| RHR | NS-3 | OpenStreetMap | SUMO | 1500 * 900 | – | 10–50 | 30–110 | – | AODV, IGPSR | 802.11p | – | 64–100 | 0.2 | 0.8 |
| PFQ-AODV | NS-2 | Midtown Manhattan map | [96] | 2500 * 2500 | 500 | – | 80 | – | AODV, AODV-L, QLAODV | 802.11p | Nakagami | 512 | 0.7 | 0.9 |
| Qgrid | – | Shanghai street | Real data | 1200 * 1200 | – | – | – | 100 | Qgrid_G, Qgrid_M, HarpiaGrid, GPSR | – | – | – | 0.8 | 0.3-0.9 |
| ALCA | – | – | VANET MobiSim | 2000 * 2000 | 500 | 13.5–22.5 | 400 | 200 | VOP | – | – | – | – | – |
| PP-AODV | NS-2 | – | – | 1500 * 1000 | – | 25–40 | 70 | 250–400 | AODV | 802.11p | – | 512 | – | – |
| RLZRP | NS-2 | – | SUMO and MOVE | 1000 * 1000 | 1000 | 0 - 16.6 | 100 | 250 | GPCR, JBR, JMSR | 802.11p | – | 512 | – | – |
| ARPRL | QualNet | Manhattan map | VANET MobiSim | 2000 * 2000 | 900 | 0–30 | 50-350 | 250 | QLAODV, AODV, QROUTI-, GPSR | 802.11p | Two ray | 512 | – | – |
| TDRRL | OPNET | Grid | Static | 5000 * 5000 | 900 | – | 8-32 | – | VOP | 802.11a | – | – | – | – |
| ADOPEL | MATLAB | Freeway | Freeway | – | – | – | 200–400 | 200 | VOP | – | – | – | 0.8 | 0.8 |
| RSAR | NS-2 | Grid | – | 1500* 1500 | 300 | 8.3–25 | 60-120 | 250 | SLB,QLAODV, GPSR | 802.11 | Two ray | 512 | – | – |
| PbQR | NS-2 | Grid | – | 2000* 2000 | 250 | 15 | 40 | – | GPSR, AODV | 802.11p | – | – | 1(Greedy) | Variable |

**TABLE 3.** *(Continued.)* **Comparison of performance evaluation parameters and techniques.**

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| QVDRP | ONE | Helsinki map | – | 4500 * 4500 | 43,200 | 0–16.67 | 40-120 | 200 | Epidemic, Spray-and-Wait, PROPHET-based | - | Nakagami | 0.5–1 MB | 0.3 | 0.8 |
| VRDRT | NS-2 | – | Vanet MobiSim | 1000* 1000 | 400 | 5–30 | – | 250 | CLAR, DRL, HQL | 802.11p | Two ray | 512 | – | – |
| QLAODV | NS-2 | Midtown Manhattan map | Freeway | 1000 * 1000 | 500 | 5 | 80 | 500 | AODV, AODV-HPDF, NRD | 802.11b | – | 512 | 0.8 | Variable |
| CRUV | NS-2 | – | SUMO and TraNS | 1700* 1700 | 500 | – | 619 | 250 | – | 802.11p | Nakagami | 56, 512, 1024 | 0.7 | 0.9 |

Note: "–" indicates that the data were not mentioned in the corresponding literature. "VOP" stands for "varied own parameters."

and RLZRP use the SUMO mobility generator. ALCA and ARPRL use VANET mobisim [100]which is also a well-known mobility generator. TDRRL, BRL-RP, and TDRL-RP uses static mobility, which is not recommended. PFQ-AODV uses a mobility model from [101] which is improved and specially designed for the VANET. Given that QGRID uses historical data to train its model, the mobility model does not apply to the protocol.

The simulation area and vehicle number [102] are related to each other. A large simulation area and a small number of vehicles are indicative of a sparse VANET network. A routing algorithm that yields superior results in these cases will also yield better results in a real-life scenario. Among the compared protocols, TDRRL uses the highest region of interest (ROI) but the least number of vehicles. Given that TDRRL uses a static mobility model, there is no opportunity to create any variation in the intra-vehicular distance. Most of the protocols simulate their models for an area between 1000 m × 1000 m and 3000 m × 3000 m. The ROIs mostly have a square shape, whereas PP-AODV and RHR use unequal numbers for the length and breadth. However, this should not impact the result. A test-bed solution is done for the PIRP algorithm, which ensures the real-life performance of the routing protocol.

Simulation time is important for routing protocols that apply online learning techniques. Likely, the initial time of simulation will not show a good result as long as the RL algorithm converges. The learning factor is directly related to the optimal simulation time [103]. For QTAR, RHR, PFQ-AODV, and QGRID, the learning factors are 0.1–1, 0.2, 0.7, and 0.8, respectively. Therefore, the convergence time as well as the simulation time may be a minimum for QGRID and PFQ-AODV routing protocols with a chance of premature convergence. However, other parameters such as the state space or action space must be the same for all the cases being compared.

Velocity is the most important factor for link breakage among the vehicles [104]. High-velocity vehicles in a two-way road segment are prone to frequent link disconnections. In the case of vehicles that travel in opposite directions, the message transmission window is smaller [105]. The communication range is also an important factor in the lifetime of a link between two vehicles, regardless of the direction of the vehicles [106]. QTAR exhibits acceptable performance with a reasonable area parameter and velocity section. The higher the speed used in the simulation, the higher the credibility of the protocol. RLZRP is simulated using a vehicle speed of 0–16.7 m/s, which is not compatible with a real-life highway scenario.

For a robust simulation, the speed should be varied from the lowest to the highest value. The lower the vehicle's speed, the higher the chances of link disconnection [107]. However, the higher the number of vehicles in the same area, the higher the chance of network interference and packet collision. From this perspective, QTAR utilizes the most widely accepted velocity for the simulation. RHR and ARPRL also adopt an acceptable range for the number of vehicles to test the performance of the protocol.

Most of the protocols use the 802.11p MAC protocol, except for TDRRL. TDRRL uses the IEEE 802.11a MAC protocol. However, 802.11p is a well-accepted MAC protocol among VANET researchers [108].

The learning rate and the discount factor are among the key parameters for the simulation of an RL algorithm. The number of convergences depends on the learning rate, and the discount factor determines the look-ahead reward for computing the reward for the current state and the corresponding action [109]. QTAR shows the best result. It varies the learning rate and the discount factor and measures the performance. However, the exact values of these two parameters are not mentioned in the reviewed paper. The BRL-RP algorithm expressed the discount factor in the Hz unit, which is different and interesting as well compared to other algorithms.

The protocols should also indicate the road topology of the network [96]. The length of the road segment and the intersection count should be given. In VANETs, intersections and traffic signals significantly affect the performance. Therefore, the exact number of intersections and traffic signals should also be indicated in addition to the other aforementioned simulation parameters. Among the discussed protocols, only

QGRID utilizes the road segment length. Some protocols indicate the path-loss or propagation model that is used. QTAR uses the street microcell/LOS propagation model, PFQ-AODV uses the Nakagami model, and APRL uses a two-ray model. The indication of the path-loss or the propagation model improves the utility of the simulation in the research community [110].

Additionally, the validation of a routing algorithm should be done by comparing the protocol with well-defined and widely-accepted protocols. However, this is massively missing, in the case of a trust-based protocol such as RAVR, SVRB, and BRL-RP. A protocol can produce different outputs with different PPS. The indication of this parameter is also important. Among the reviewed protocols, the PPS value is indicated for only RHR and QGRID.

## V. RECOMMENDATIONS

In table IV, the building blocks of RL designs in the discussed routing protocols are given. Building blocks refer to the state, agent, action, and reward parameters of the RL algorithms. In this section, the RL algorithms used in the reviewed routing protocols are critically analyzed based on the design of their RL building blocks. After that, the authors' recommendation on the configuration of the parameters is addressed. In the last subsection, the learning techniques are analyzed in terms of the application scenario under different conditions.

### A. ANALYSIS OF THE ROUTING PROTOCOLS BASED ON RL PARAMETERS

In the case of the formation of state and action, QTAR [56] follows the common paradigm, where the state space contains the neighboring vehicles and the action is defined as forwarding packet to the next available vehicle. However, the design of the reward function is interesting. They have considered link quality, link expiration time, and delay as the reward calculation matrix. The reward function also has some weighting factors and, thus, can be tuned according to the need and environments.

RHR [49] uses next-hop neighbors as the available states which ensure the limited size of the Q-table. However, in this research, other parameters are not stated properly. PFQ-AODV [60] uses an idle time ratio to calculate the bandwidth factor, which is more practical compared to the process where the available bandwidth is calculated based on one-time data only.

QGRID [51] is a grid-based protocol and the grids are considered as the states for the implemented Q-learning procedure. Even though the agent is mentioned as the conceptualized virtual agent, the vehicles themselves work as the agents according to the working procedure. However, a better grid is chosen with the help of a cleverly designed discounted factor.

ALCA [64] considers speed and angle to evaluate the value of traffic flow. Four virtualized agents (i.e., DUA, RLA, ZSA, and SCA) are conceptualized in this protocol. The working procedure of the agents is given in Section III. The learning factor is calculated for each agent. They learn by interactive actions among themselves based on the positive and negative rewards they receive. In ALCA, the state, reward, and actions are not mentioned explicitly. Hence, the values in the table are given based on the authors' inference.

Like ALCA, RL parameters are not precisely given in PP-AODV [59]. However, the parameters for learning are mentioned. Even though RLZRP [52] implements a zone-based routing protocol, less information is given in the literature. The assumptions for ARPRL [61] on states, agent, and action is trivial and similar to other approaches. However, ARPRL is a proactive routing protocol. The mechanism for updating Q-value is not suitable for a high-speed scenario like VANET. It could have employed advantages from the dynamic discount factor but, according to the protocol, it is fixed and the value is 1. TDRRL [67] uses a central mechanism and exploits the mechanism of DRL. In a central control-based situation, DRL will perform better than the normal RL procedure. ADOPEL [53] considers the neighboring degree for deriving the reward. This approach will reduce the amount of data to be transmitted over links. Furthermore, the same strategy can also be used to select CHs.

In RSAR [2], the usage of the bandwidth factor is shown differently. However, the parameter is affected by the immediate reward and forces the algorithm to update the link entry with a better bandwidth. The PbQR routing algorithm [57] considers the computational capacity of the node as the agent. However, according to the procedure, this is just another way of mentioning the decision-making capacity of the vehicles.

QVDRP [69] uses a reward system where, if the forwarder is connected to any gateway (RSU), the sender gets an immediate reward of 1; otherwise, 0. A node that is not directly connected to the RSU gets a discounted reward from the directly connected vehicle. The Q-value is updated every 10 minutes, which is not feasible, as the distance and connection directly depend on the distance only. This might decrease the number of control packet exchanges but it is not an efficient way to update the Q-values. If a node wants to update its Q value of an RSU, the value should be updated based on the distance rather than time only.

VRDRT [54] uses the RL technique to predict the vehicle density in the road. The DRL algorithms run in RSUs. As the RSUs are fixed, the prediction can be propagated to vehicles. On the other hand, RSUs can be easily equipped with more computational power. Besides, they are also connected through wires with each other. Road segment vehicle's density prediction with each other will help the entire network to choose the intermediate road junction. The DRL technique used in VRDRT is a spatiotemporal solution and the implementation feasibility is also higher.

In QLAODV [62], the RL design is a little tricky. This design implements the mobility factor and bandwidth factor to determine the discount factor, which practically works as the reward function, as shown for other protocols. However, the design needs to exchange periodic hello packets. This will create an adverse effect on the bandwidth, and this is not

**TABLE 4.** Comparison of RL algorithm parameters.

| Protocol | RL algorithm | Agent | State | Action | Reward | Discounted factor |
|---|---|---|---|---|---|---|
| RAVR | Policy hill climbing | UAV | SINR and BER between UAV and vehicle | Relaying to RSU or not | Channel power gain | – |
| SVRB | Q-Learning | Vehicles | Reputation vector, location, and velocities | Selecting the next hop | Attack | fixed |
| BRL-RP | DQN | SDN domain controller | Throughput and delay | Selection of primary node, computing server, consensus node, trusted neighbor | Selecting a route with better throughput and delay | – |
| TDRL-RP | DQN | SDN controller | Vehicle position, and forwarding ratio | Selecting next hop node | Trust value | – |
| RGVR | Q-Learning | Network nodes | RSUs | Selecting next hop for data transmission | 1 if directly connected with destination; otherwise, 0 | Number of hops, payoffs amount, link quality |
| PIRP | Q-Learning | Each packet | Each node | Forwarding packet to a one-hop neighbor | Transmission rate, vehicles' mobility | Number of hops and link status |
| RLRC | Sarsa-Lambda | CHs | Other CHs | Forwarding packet to a CH | Link status, bandwidth | Number of hop |
| HQVR | Q-Learning | All packets | All nodes | Set of neighbors | Number of hops, link reliability, bandwidth | – |
| RL-SDVN | Q-Learning | SDN-controller | Vehicles | Forwarding packet to a specific neighbor | Distance, delay | N/A |
| Q-LBR | Q-Learning | UAV | Tuple of congestion in UAV and ground vehicle | Selection of URPA | Congestion threshold | – |
| QTAR | Q-Learning | Each packet | Vehicles | Forwarding packet to a specific neighbor | Link quality, link expiration time, and delay | Variable |
| RHR | Q-Learning | – | Vehicles | Receiving various kinds of packets related to current next hop | Broadcast overhead | Fixed |
| PFQ-AODV | Q-Learning | All packets | One- and two-hop neighbors | Set of one-hop neighbors | Bandwidth factor, relative mobility factor, and link quality factor | Fixed |
| Qgrid | Q-Learning | Conceptualized virtual agent | Each grid | Moving in between grids | 100 if packet reach destination; otherwise, 0 | Number of vehicles in the grid |
| ALCA | Modified RL | Four virtualized concepts (DUA, RLA, ZSA, and SCA) | Vehicles position | Exchanging perceived information among each other's | Traffic flows | N/A |
| PP-AODV | Q-Learning | – | Vehicles | Transmitting control packet | Bandwidth, delay, and packet collision probability | – |
| RLZRP | – | Vehicles | A tuple of junction and vehicles | Transmitting control packet | – | – |
| ARPRL | Q-Learning | Each packet | Each vehicle | One-hop neighbor | Hello message reception ratio, and route reliability | Fixed |
| TDRRL | DRL | SDN controller | Combination of forwarding ratio and location | Selection of next-hop neighbors | Trust score based on forwarding ratio of control packets | Time-dependent |
| ADOPEL | Q-Learning | Each vehicle | Current vehicle | Selecting next relay | Maximizing aggregation ratio, number of neighboring vehicles, and delay | Link stability factor based on stable neighbors |
| RSAR | Q-Learning | Each vehicle | All nodes | Transmission of beacon packet | 1 if directly connected with destination; otherwise, 0 | Link reliability, available bandwidth, and number of hops |
| PbQR | Q-Learning | Total computational capacity of each node | Each node | Set of neighbors | Stability factor and continuity factor | Distance factor |

**TABLE 4.** *(Continued.)* **Comparison of RL algorithm parameters.**

| | | | | | | |
|---|---|---|---|---|---|---|
| QVDRP | Q-Learning | Each node | – | Selecting next hop | Direct connection with RSU | Elapsed time with encounter and number of hops |
| VRDRT | DRL | An operator in RSU | Environment condition in each time slot | Selecting route | Link utilization | N/A |
| QLAODV | Q-Learning | All packets | All nodes | Set of neighbors | 1 if directly connected with destination; otherwise, 0 | Mobility factor and bandwidth factor |
| CURV | Q-Learning | All packets | All nodes | Set of one-hop neighbors | 1 if directly connected with destination; otherwise, 0 | Number of hops and link quality |

Note: "–" indicates that the data are not mentioned in the corresponding literature. "N/A" stands for "not applicable."

a characteristic of the reactive routing protocol. CURV [65] also uses the discount factor similar to QLAODV. This design also ensures the least number of hops.

RAVR [79] uses UAV as the RL agent. It is a secured routing protocol proposed for VANET architecture. The message evaluation technique depends on the SINR and BER values. Thus, the RL tries to switch to a better state by selecting the next node where better SINR and BER values will be found. The reward is the channel power gain. However, this is a clever design, as a greater channel gain will ensure better BER and SINR values.

SVRB [78] implements the RL mechanism in the blockchain environment. Thus, it is legit that the vehicles are designed as the agents and the tuple of reputation, and vehicles' positional information is used as the states. The agent gets penalized when it faces any attack. BRL-RP [81] and TDRL-RP [80] follow the same RL design. As they follow logically central architecture, DRL is a suitable form of RL technique for both approaches. However, the main difference is in the reward mechanism. BRL-RP focuses on the network performance parameter whereas TDRL-RP focuses on the trust value. From this analysis, we can say that BRL-RP will perform better if we measure the performances from a routing perspective.

Sarsa(λ)-learning technique is used in the RLRC routing algorithm. The TD(λ) based solution raises a propagation delay problem. However, RLRC optimizes the problem by forming clusters.

Both RGVR [74] and RL-SDVN [75] assume the static infrastructure as the agent. They give the routing protocol the freedom of putting more computation power. Though RL-SDVN will learn about the routes with lesser distance and delay due to the special design of the reward function.

### B. RECOMMENDATION ON RL PARAMETERS

To compare the foundation block of the RL procedure, it can be seen that most of the protocols assume the next-hop neighbors as the available states. Hence, delivering the packet to the available states becomes the action. There are some differences that can be seen in the assumptions of the agent. Some of the protocols conceptualize the packets as the agent whereas the vehicle is considered as the agent for most of the protocols. This definition might raise ambiguity. For a centralized architecture, however, the formulation of the agent is easier and the controllers fulfill the duties. The main controlling parameter is the reward function and the discount factor. The routing protocols find the optimality of the protocols based on the tuning parameters. Link quality, delay, link expiration time, traffic flows, available bandwidth, neighbor degree, link stability factor, and last listening time are used as the parameters for both reward and discount factor mechanism [111]. The fixed discount factor does not help to choose a better link for next-hop neighbors. However, to deploy some sort of parameter optimization, there is a chance to increase the number of exchanged control packets.

### C. SUITABLE APPLICATION SCENARIO AND RECOMMENDATION ON THE LEARNING TECHNIQUES

From Table 4 , we can observe that mostly two types of RL algorithms of Q-learning and DRL are utilized to design VANET routing protocols. From the definition, we can say that Q-learning is a distributed algorithm. The design of state, action, and the agent is relatively easier [112] compared to other forms of RL variants. Because Q-learning is a model-free learning algorithm, it does not require any prior knowledge about the environment. However, for this reason, the consumption of control packets is higher [113]. The design and implementation of DRL in the VANET environment are more interesting. In the case of the high-speed mobility scenario, however, the Q-learning algorithm will have to go for a higher learning rate. More importantly, designing the building blocks such as state, action, and the agent is a tough task. Any kind of extra information sharing means the exchange of extra packet transmission [114]. Thus, a vanilla Q-learning approach might create an adverse effect before optimizing any intended QoS parameters.

The SDN-based centralized routing protocols implement DRL-based prediction in RSUs. This is a cost-effective solution [115]. According to the working nature of the DRL algorithm, they are computation-resource-hungry procedure. However, as RSUs are fixed and the number of RSUs is less than the number of vehicles on the road, implementing DRL in the RSUs are more practical. DRL approach is more of an offline-based learning technique, and it is suitable for scenarios where the number of states is large.

The action value-based solutions need to store a large number of values. To use such techniques, the routing protocols need to define the states carefully. A protocol that conceptualizes the controllers or the CHs as the state might be a good scenario, where this tabular solution can be applied. However, for a continuous state, the policy-based solutions are much preferable. The PHC algorithm can find the optimal policy within a shorter amount of time. PHC or other policy-based solutions can be used for learning purposes where each vehicle is conceptualized as the state.

The end-to-end delay is a major factor for the high mobility scenarios like VANETs. The TD($\lambda$) based solutions opt for multiple future rewards. This increases the propagation delay, and the feedback mechanism also gets complicated as well. Thus, TD($\lambda$) based solution such as SARSA($\lambda$) is not appropriate for high mobility scenarios. However, if the states are the static infrastructures such as RSU or central controller or even a special unit like UAV, careful design can bring out efficient routing performance [44].

A superior routing performance can be ensured if a routing protocol implements both DRL prediction in the RSU end and the Q-learning based distributed algorithm in the vehicle end. The convergence time and buffer condition analysis should also be a design consideration for the RL based routing algorithms.

## VI. OPEN RESEARCH ISSUES AND CHALLENGES

Open research issues and challenges are discussed in this section. These are important for future researchers since they serve as initial points for new research ideas. The highlighted challenges are still under investigation and are important in VANET research. Every issue mentioned in the section contains three different parts: lessons learned, limitations and challenges, and future direction and recommendation.

### A. OVERHEAD CONTROL MECHANISM

Routing overhead refers to the extra burden that the routing protocols create over the wireless links by exchanging relatively smaller control packets to establish and maintain the routes [116].

#### 1) LESSONS LEARNED

To update the routing table, the Q-learning-based routing algorithms require a feedback mechanism for the QoS parameters. For example, a protocol can consider the delay, link quality, distance, and energy as the parameters to be optimized using the Q-learning algorithm.

#### 2) LIMITATIONS AND CHALLENGES

The feedback mechanism often leads to an increase in network overhead. Due to the narrow bandwidth, the overhead limits the channel capacity for the transmission of data packets, resulting in poor routing performance [117].

#### 3) FUTURE DIRECTION AND RECOMMENDATION

To enable an RL-based routing protocol, a feedback mechanism is necessary. However, different parameters such as available bandwidth, link condition, mobility information, link quality factor, and neighboring degree can be taken into consideration based on different inference or prediction mechanism. Specialized hierarchical routing can limit the consumption of control packets to some extent. Transmission circle for neighbor discovery can also be reduced in order to suppress the control overheads.

### B. STATE LIMITATION PROBLEM

According to the original definition of the RL paradigm, a state represents the current situation of the environment that the agent is acting upon [118].

#### 1) LESSONS LEARNED

The investigated routing algorithms in this survey formulate the states differently. Most of the RL-based routing protocols assume that the vehicles are states. The number of states depends on the number of neighboring vehicles on the road in a specific period [119].

#### 2) LIMITATIONS AND CHALLENGES

With the increase in the number of vehicles, the number of states also increases and, thus, the exploration time is increased to determine the best possible states. The algorithms discussed in this survey reduce the number of states based on a random choice using a static threshold. This mechanism can lead to convergence to a local optimum, in which the best neighboring states may remain hidden.

#### 3) FUTURE DIRECTION AND RECOMMENDATION

In the case of a large state space, the optimality can be compromised for time-constrained operation. Other approaches can include a threshold-based solution. In such a solution, after reaching the threshold value, an agent might not explore any more. This approach will also bring positive results in terms of control overhead.

### C. Q-TABLE MAINTENANCE

The Q-Learning algorithm stores the Q-values inside a table called Q-table. In a traditional Q-table, the rows contain the states and the columns contain actions. Each cell contains the corresponding Q-value for a specific state for taking the specific action [120].

#### 1) LESSONS LEARNED

A routing protocol may contain multiple destinations for a single source. However, in the routing protocols investigated in this survey, any specific criteria to control the size of the Q-table have not been mentioned.

#### 2) LIMITATIONS AND CHALLENGES

With a larger Q-table, more control packets and longer delays are needed to maintain the Q-table. With the increase in the number of states in the case of multiple destinations, the size of the Q-table increases exponentially. The minimization of the Q-table length and the identification of the best route for a specific destination are additional challenges [24].

#### 3) FUTURE DIRECTION AND RECOMMENDATION

The number of states is directly related to the size of the Q-table. One way to keep the Q-table shorter is to keep the number of states low. In case of the necessity of keeping a large Q-table, an efficient updating procedure is a must. To reduce this problem, DQN is introduced through DQN is a computation hungry algorithm.

### *D. TRAFFIC PREDICTION FOR ONLINE APPLICATIONS*

Traffic prediction includes the prediction of vehicles at a specific time in a road segment.

#### 1) LESSONS LEARNED

An SDN-based routing algorithm, TDRRL, has shown the primary implementation of traffic density prediction procedure. Other protocols do not take any assistance from any kind of traffic prediction mechanism.

#### 2) LIMITATIONS AND CHALLENGES

Traffic prediction can actively save a lot of bandwidth of the fragile wireless links in VANETs. Even though TDRRL has shown the primary implementation of the prediction mechanism, but none of the algorithms have implemented any distributed prediction system.

#### 3) FUTURE DIRECTION AND RECOMMENDATION

The algorithms discussed in this survey try to realize the traffic condition based on instant broadcasting by filling out the routing learning table. In online learning mechanisms, the routing algorithms learn a route based on QoS parameters for a specific moment in order to deliver the routing packet to the destination [121]. Given that a VANET is implemented in a highly dynamic environment, traffic depends on time, road segments, environment, and geographical infrastructure. An accurate prediction of traffic for a specific geographical environment will lead to faster convergence and superior QoS optimization [122]. Besides the unsupervised learning algorithm, the supervised learning algorithms can be implemented in a distributed manner to enable the prediction for both urban and rural areas.

### *E. ONLINE AND OFFLINE LEARNING*

Online learning refers to the learning mechanism, where the agent learns actively based on the current interaction and no historical data is fed into the learning system. However, offline learning involves some pre-knowledge assistance for taking any decision.

#### 1) LESSONS LEARNED

Several routing protocols implement learning procedure based on historical data, which can be described as offline learning. However, we have not found any solution which works both online and offline.

#### 2) LIMITATIONS AND CHALLENGES

Although this type of learning is very helpful in network channel utilization, unpredictable roadside events can lead to poor routing performance.

#### 3) FUTURE DIRECTION AND RECOMMENDATION

To improve performance, online and offline learning is necessary for VANET scenarios. The offline-based solution can be implemented in the infrastructures such as RSU, and the vehicle can implement online-based solutions.

### *F. CONVERGENCE TIME*

The convergence time refers to the iteration count that an algorithm takes to find out the optimal solution. It should be kept in mind that a sub-optimal solution does not necessarily mean a bad solution [123]. Considering other parameters such as time and energy constraints, a sub-optimal solution can also be a desirable solution.

#### 1) LESSONS LEARNED

The investigated online-based RL routing algorithm in this survey opted for optimal solutions. However, some algorithms aim to use the greedy solution by keeping aside the exploration capability.

#### 2) LIMITATIONS AND CHALLENGES

None of the routing protocols showed any tradeoffs between optimality and sub-optimality by considering different situations. Reward tuning mechanisms are available in some algorithms such as QTAR, but the weighting factor assignments are not done dynamically based on situation analysis.

#### 3) FUTURE DIRECTION AND RECOMMENDATION

In the case of online learning, routing packets need to be sent within a minimum time interval. If the convergence time is long, there is a possibility that the states will change their position and the selected hops will yield relatively poor results. In Q-learning, the appropriate estimation of the learning rate is required to reduce the convergence time [48]. Considering sub-optimal solutions can be a great feature when a critical situation arises, such as delivering warning messages.

### *G. FIXING EXPLORATION AND EXPLOITATION STRATEGY*

In Q-learning-based solutions, exploration means taking an action from a state, for which the Q-value is unknown. The exploitation is such a scenario where the agent acts on the already evaluated actions and does not search for the Q-value of other non-evaluated actions.

#### 1) LESSONS LEARNED

In the investigated routing protocols, a mixture of exploration and exploitation was witnessed. The balance is brought with the help of the learning rate. A higher learning rate means quick convergence with a chance of premature convergence whereas a lower learning rate means a greater time before convergence and more exploration.

#### 2) LIMITATIONS AND CHALLENGES

For a highly dynamic scenario, the environment of a VANET system changes rapidly, which can invalidate the precalculation of the previous-hop result [124]. The exploitation mechanism is as important as the exploration strategy, especially for the VANET environment. The optimality and tradeoffs between exploration and exploitation should be further investigated for such a dynamic environment.

#### 3) FUTURE DIRECTION AND RECOMMENDATION

Depending on the need, situation, and packet type, the routing protocols can select a variable learning rate. The generated message can be categorized into multiple types. Based on the requirement, the route discovery process time can be fixed. The velocity and relative velocity should also be kept in mind before selecting the learning rate for the next route searching process.

### H. SECURITY

Security in VANET routing involves message spoofing, replay attack, integrity attack, impersonation attack, and denial of service [125].

#### 1) LESSONS LEARNED

The Q-value mechanism depends on the sent information from neighbors. Any node can easily provoke the sender by advertising a higher Q-value, where breaching security becomes easier for intruder nodes. None of the investigating protocols takes any extra precaution to measure the security.

#### 2) LIMITATIONS AND CHALLENGES

Security is one of the most important concerns for any network system. In VANETs, fake and selfish nodes can be easily implemented, which may exhibit greater QoS privileges and can be chosen by the sender as the intermediary nodes [126].

#### 3) FUTURE DIRECTION AND RECOMMENDATION

Trust management can be utilized to enhance the security measure of the general-purpose RL routing protocols for VANETs. Moreover, there should be a greater emphasis on the implementation of a robust RL-based routing protocol [127].

### I. QoS BALANCE

The main purpose of the RL-based VANET routing protocol is to meet the QoS requirements. The QoS requirements include PDR, EED, throughput, jitter, and priority [128].

#### 1) LESSONS LEARNED

Different QoS parameters are considered for the investigated routing protocols presented in this survey. Most of the protocols aim to optimize a single QoS parameter of the routing mechanism.

#### 2) LIMITATIONS AND CHALLENGES

The QoS parameters have tradeoffs and sometimes contradict each other [129]. Even though most of the protocols aim to optimize one or multiple performance metrics, but only QTAR has the flexibility to balance the weight based on the need.

#### 3) FUTURE DIRECTION AND RECOMMENDATION

Depending on the environment and requirements, QoS parameter considerations should be carefully handled to obtain an optimum routing result. Message priority selection can help to determine which parameter should be given more priority over others.

### J. POSITION PREDICTION

Position prediction explains a prediction mechanism, where an intelligent agent can predict the position of a vehicle in a near future [130].

#### 1) LESSONS LEARNED

The DRL-based solution and offline solution have some supervised knowledge about the road condition. However, none of the protocols have implemented any mechanism to predict the near future location of the destination or intermediary nodes of the selected route.

#### 2) LIMITATIONS AND CHALLENGES

Vehicle position prediction leads to an improved result in the delivery of data to the destination node. Given that the nodes are highly mobile and the topology of the road structures might cause problems in the propagation line, predicting the destination vehicle's position in addition to the intermediary vehicles' position leads to the improved results and faster convergence.

#### 3) FUTURE DIRECTION AND RECOMMENDATION

For the simplest solution, a spatiotemporal prediction can be brought based on the vehicle's current position, velocity, direction, and vehicle type. A complex solution may introduce a Markovian chain-based solution.

### K. ROAD TOPOLOGY-AWARE ROUTING

Road topology knowledge includes information about the road segments, roadside obstacles, intersections, traffic lights, zebra crossing, and lane numbers. An effective version may include information about bus stoppages and timing [131].

#### 1) LESSONS LEARNED

Though the investigated protocols only assume some sort of roadside infrastructures such as RSUs, they do not take any assistance from the road topology information.

#### 2) LIMITATIONS AND CHALLENGES

The topological information about roads can drastically aid the learning mechanism, especially the traffic light, vehicle density, and intersection-aware protocols. Some common inferences can be drawn based on the information about roadside obstacles. As an example, if there is a school ahead, a vehicle will likely reduce the speed limit.

#### 3) FUTURE DIRECTION AND RECOMMENDATION

Most protocols are applied to urban road topology. However, other topologies such as highways and rural roads should be considered. The traffic condition, sparse road condition, zone disconnection, RSU unavailability, and intersection or junction should be considered in future designs to achieve the improved performance of RL-based VANET routing algorithms.

### *L. TEST-BED EXPERIMENTS*

Test-bed experiments refer to the paradigm of the experiment, where the proposed method is validated by collecting data with a real-life setup[132]

#### 1) LESSONS LEARNED

In none of the RL-based routing protocols discussed in this survey, no test-bed experiment was conducted.

#### 2) LIMITATIONS AND CHALLENGES

The VANET simulators have gone through a lot of advancements and also mimics the natural environment. However, real-life road condition changes based on countless variables.

#### 3) FUTURE DIRECTION AND RECOMMENDATION

The first validation should be done by the simulators. However, for an industry-grade protocol, the protocol should be validated with test-bed experiments. As none of the RL-based VANET routing protocols have done test-bed experiments so far, the test-bed experiment is more vital to this paradigm.

### *M. ADAPTIVE HELLO INTERVAL*

Starting from the process of route discovery to route recovery, the routing protocols take the help of hello packets. The hello packets are smaller compared to data packets [133].

#### 1) LESSONS LEARNED

Among the investigating protocols, only QVDRP uses two types of hello intervals.

#### 2) LIMITATIONS AND CHALLENGES

Adaptive hello interval can play a major role to maintain the QoS parameters of VANET routing. The RL routing protocols use hello packets to maintain the neighboring information. Without an adaptive interval, the vehicles will keep receiving and transmitting such packets, resulting in huge consumption of the limited wireless bandwidth.

#### 3) FUTURE DIRECTION AND RECOMMENDATION

The vehicles working as the agent should adopt an adaptive hello-interval time. The time interval can be processed based on the requirements, neighboring mobility, relative direction, message types, and priority.

## VII. CONCLUSION

Increasing the efficiency of the VANET routing algorithm is one of the core concerns of researchers. The RL algorithm is the only branch of ML wherein the efficiency of a certain system continues to increase with time. The most significant difference between RL algorithms and other AI algorithms is that with more experience, RL algorithms can continuously improve performance, whereas other paradigms are limited by the given information. In this report, we surveyed the VANET routing protocol, which is proposed based on RL algorithms. The routing algorithms are discussed in addition to their advantages, disadvantages, and the most suitable applications. The algorithms are also critically compared by discussing their optimization criteria and core principles in a tabular format. The impact of the optimization criteria is also outlined, and the opinion of the authors is presented. We show that the applicability, validity, and acceptance of a proposed protocol depend on the validation policy. For convenience, the simulation environment and other parameters are presented in a tabular format, and they are subsequently discussed. For future researchers, the research gaps and the areas that require critical improvement are emphasized as open research issues. By critically analyzing the core ideas and performances of the protocols presented in the reviewed papers, this report undertakes a comprehensive survey of RL-based VANET routing algorithms. The analysis, discussion, comparison, and future research direction highlighted in this investigation will provide VANET researchers with an in-depth overview of existing RL-based VANETs. Thus, this survey will play a crucial role in future studies in related fields.

## ACKNOWLEDGMENT

The authors wish to thank the Editor and anonymous referees for their helpful comments in improving the quality of this article.

## REFERENCES

[1] S. Sharma and A. Kaul, “A survey on intrusion detection systems and honeypot based proactive security mechanisms in VANETs and VANET cloud,” *Veh. Commun.*, vol. 12, pp. 138–164, Apr. 2018.

[2] D. Zhang, T. Zhang, and X. Liu, “Novel self-adaptive routing service algorithm for application in VANET,” *Int. J. Speech Technol.*, vol. 49, no. 5, pp. 1866–1879, May 2019.

[3] K. Mehta, L. G. Malik, and P. Bajaj, “VANET: Challenges, issues and solutions,” in *Proc. Int. Conf. Emerg. Trends Eng. Technol. (ICETET)*, Dec. 2013, pp. 78–79.

[4] C. Cooper, D. Franklin, M. Ros, F. Safaei, and M. Abolhasan, ''A comparative survey of VANET clustering techniques,'' *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 657–681, 1st Quart., 2017.

[5] Y. Du, L. Yue, and S. Liu, ''Optimization of combined horizontal and vertical curves design of mountain road based on vehicle-road coordination model,'' in *Proc. 5th Int. Conf. Transp. Inf. Saf. (ICTIS)*, Jul. 2019, pp. 16–24.

[6] H. A. Cozzetti, C. Campolo, R. Scopigno, and A. Molinaro, ''Urban VANETs and hidden terminals: Evaluation through a realistic urban grid propagation model,'' in *Proc. IEEE Int. Conf. Veh. Electron. Saf. (ICVES)*, Jul. 2012, pp. 93–98.

[7] M. Singh and J. Sharma, ''Performance analysis of secure & efficient AODV (SE-AODV) with AODV routing protocol using NS2,'' in *Proc. 3rd Int. Conf. Rel., Infocom Technol. Optim. Trends Future Directions (ICRITO)*, 2015, pp. 1–6.

[8] L. Pan, ''An improved the DSR routing protocol in mobile ad hoc networks,'' in *Proc. 6th IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Sep. 2015, pp. 591–594.

[9] M. Manjunath and D. H. Manjaiah, ''Spatial DSDV (S-DSDV) routing algorithm for mobile ad hoc network,'' in *Proc. Int. Conf. Contemp. Comput. Informat. (IC3I)*, Nov. 2014, pp. 625–629.

[10] C. Fenhua and J. Min, ''Improved GPSR routing algorithm and its performance analysis,'' in *Proc. IEEE Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Jul. 2010, pp. 49–52.

[11] H. Geng, H. Zhang, X. Shi, Z. Wang, X. Yin, J. Zhang, Z. Hu, and Y. Wu, ''A hybrid link protection scheme for ensuring network service availability in link-state routing networks,'' *J. Commun. Netw.*, vol. 22, no. 1, pp. 46–60, Feb. 2020.

[12] R. A. Nazib and S. Moh, ''Routing protocols for unmanned aerial vehicle-aided vehicular ad hoc networks: A survey,'' *IEEE Access*, vol. 8, pp. 77535–77560, Apr. 2020.

[13] R. Mitchell, J. Michalski, and T. Carbonell, *An Artificial Intelligence Approach*. Berlin, Germany: Springer, 2013.

[14] P. Sharma, H. Liu, H. Wang, and S. Zhang, ''Securing wireless communications of connected vehicles with artificial intelligence,'' in *Proc. IEEE Int. Symp. Technol. Homeland Secur. (HST)*, Apr. 2017, pp. 1–7.

[15] A. M. Alrehan and F. A. Alhaidari, ''Machine learning techniques to detect DDoS attacks on VANET system: A survey,'' in *Proc. 2nd Int. Conf. Comput. Appl. Inf. Secur. (ICCAIS)*, May 2019, pp. 1–9.

[16] R. T. Rodoshi, T. Kim, and W. Choi, ''Resource management in cloud radio access network: Conventional and new approaches,'' *Sensors*, vol. 20, no. 9, p. 2708, May 2020.

[17] V. Krundyshev, M. Kalinin, and P. Zegzhda, ''Artificial swarm algorithm for VANET protection against routing attacks,'' in *Proc. IEEE Ind. Cyber-Phys. Syst. (ICPS)*, May 2018, pp. 795–800.

[18] M. El Amine Fekair, A. Lakas, and A. Korichi, ''CBQoS-VANET: Cluster-based artificial bee colony algorithm for QoS routing protocol in VANET,'' in *Proc. Int. Conf. Sel. Topics Mobile Wireless Netw. (MoWNeT)*, Apr. 2016, pp. 1–8.

[19] J. Dowling, E. Curran, R. Cunningham, and V. Cahill, ''Using feedback in collaborative reinforcement learning to adaptively optimize MANET routing,'' *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 35, no. 3, pp. 360–372, May 2005.

[20] S. S. Doddalinganavar, P. V. Tergundi, and R. S. Patil, ''Survey on deep reinforcement learning protocol in VANET,'' in *Proc. 1st Int. Conf. Adv. Inf. Technol. (ICAIT)*, Jul. 2019, pp. 81–86.

[21] J. Liu, Q. Wang, C. He, K. Jaffrès-Runser, Y. Xu, Z. Li, and Y. Xu, ''QMR: Q-learning based multi-objective optimization routing protocol for flying ad hoc networks,'' *Comput. Commun.*, vol. 150, pp. 304–316, Jan. 2020.

[22] W. P. Coutinho, M. Battarra, and J. Fliege, ''The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review,'' *Comput. Ind. Eng.*, vol. 120, pp. 116–128, Jun. 2018.

[23] L. P. Kaelbling, M. L. Littman, and A. W. Moore, ''Reinforcement learning: A survey,'' *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, May 1996.

[24] S. Chettibi and S. Chikhi, ''A survey of reinforcement learning based routing protocols for mobile ad-hoc networks,'' in *Recent Trends in Wireless and Mobile Networks* (Communications in Computer and Information Science), vol. 162. Springer, 2011, pp. 1–13.

[25] C. Wu, X. Chen, Y. Ji, F. Liu, S. Ohzahata, T. Yoshinaga, and T. Kato, ''Packet size-aware broadcasting in VANETs with fuzzy logic and RL-based parameter adaptation,'' *IEEE Access*, vol. 3, pp. 2481–2491, 2015.

[26] C. Wu, Y. Ji, X. Chen, S. Ohzahata, and T. Kato, ''An intelligent broadcast protocol for VANETs based on transfer learning,'' in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, May 2015, pp. 1–6.

[27] B. Yu, C.-Z. Xu, and M. Guo, ''Adaptive forwarding delay control for VANET data aggregation,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 1, pp. 11–18, Jan. 2012.

[28] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[29] M. Van Otterlo and M. Wiering, ''Reinforcement learning and Markov decision processes,'' in *Adaptation, Learning, and Optimization*, vol. 12. New York, NY, USA: Springer-Verlag, 2012, pp. 3–42.

[30] M. P. Deisenroth, ''A survey on policy search for robotics,'' *Found. Trends Robot.*, vol. 2, nos. 1–2, pp. 1–142, 2011.

[31] W. Rei, M. Gendreau, and P. Soriano, ''A hybrid Monte Carlo local branching algorithm for the single vehicle routing problem with stochastic demands,'' *Transp. Sci.*, vol. 44, no. 1, pp. 136–146, Feb. 2010.

[32] S. Chettibi and S. Chikhi, ''Adaptive maximum-lifetime routing in mobile ad-hoc networks using temporal difference reinforcement learning,'' *Evolving Syst.*, vol. 5, no. 2, pp. 89–108, Jun. 2014.

[33] M. Maleki, V. Hakami, and M. Dehghan, ''A model-based reinforcement learning algorithm for routing in energy harvesting mobile ad-hoc networks,'' *Wireless Pers. Commun.*, vol. 95, no. 3, pp. 3119–3139, Aug. 2017.

[34] J. Chen, B. Yuan, and M. Tomizuka, ''Model-free deep reinforcement learning for urban autonomous driving,'' in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 2765–2771.

[35] S. B. Thrun and S. B. Thrun, ''Efficient exploration in reinforcement learning,'' Northwestern Univ., Evanston, IL, USA, Tech. Rep. NU-CSS-93-14, Nov. 1993.

[36] R. J. Williams and L. C. Baird, ''Tight performance bounds on greedy policies based on imperfect value functions,'' School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CS-CMU-92-102, 1992.

[37] T. Michel, ''Adaptive-greedy exploration in reinforcement learning based on value differences,'' in *Proc. Annu. Conf. Artif. Intell.* Berlin, Germany: Springer, 2010, pp. 203–210.

[38] K. Asadi and M. L. Littman, ''An alternative softmax operator for reinforcement learning,'' in *Proc. PMLR*, Jul. 2017, pp. 243–252.

[39] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári, ''Convergence results for single-step on-policy reinforcement-learning algorithms,'' *Mach. Learn.*, vol. 38, no. 3, pp. 287–308, Mar. 2000.

[40] S. Fujimoto, D. Meger, and D. Precup, ''Off-policy deep reinforcement learning without exploration,'' in *Proc. PMLR*, May 2019, pp. 2052–2062.

[41] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel, ''Trust region policy optimization,'' in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1889–1897.

[42] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, ''Proximal policy optimization algorithms,'' Jul. 2017, *arXiv:1707.06347*. [Online]. Available: https://arxiv.org/abs/1707.06347

[43] T. L. Google, D. Schuurmans, G. Ai, and C. Boutilier, ''Non-delusional Q-learning and value iteration,'' in *Proc. NIPS*, 2018, pp. 9971–9981.

[44] A. Habib, M. I. Khan, and J. Uddin, ''Optimal route selection in complex multi-stage supply chain networks using SARSA(λ),'' in *Proc. 19th Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dec. 2016, pp. 170–175.

[45] Y. Xu, M. Lei, M. Li, M. Zhao, and B. Hu, ''A new anti-jamming strategy based on deep reinforcement learning for MANET,'' in *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*, Apr. 2019, pp. 1–5.

[46] C. Gaskett, D. Wettergreen, and A. Zelinsky, ''Q-learning in continuous state and action spaces,'' in *Proc. Australas. Joint Conf. Artif. Intell.* Berlin, Germany: Springer, 1999, pp. 417–428.

[47] C. J. C. H. Watkins and P. Dayan, ''Q-learning,'' *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.

[48] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, ''Applications of deep reinforcement learning in communications and networking: A survey,'' *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, 4th Quart., 2019.

[49] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, ''Deep reinforcement learning: A brief survey,'' *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

[50] R. Tasnim Rodoshi, T. Kim, and W. Choi, ''Deep reinforcement learning based dynamic resource allocation in cloud radio access networks,'' in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2020, pp. 618–623.

[51] Y. Li, X. Hu, Y. Zhuang, Z. Gao, P. Zhang, and N. El-Sheimy, ''Deep reinforcement learning (DRL): Another perspective for unsupervised wireless localization,'' *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6279–6287, Jul. 2020.

[52] M. Bowling and M. Veloso, "Rational and convergent learning in stochastic games," in *Proc. IJCAI Int. Jt. Artif. Intell.*, 2001, pp. 1021–1026.

[53] G. A. Walikar and R. C. Biradar, "A survey on hybrid routing mechanisms in mobile ad hoc networks," *J. Netw. Comput. Appl.*, vol. 77, pp. 48–63, Jan. 2017.

[54] X. Ji, W. Xu, C. Zhang, T. Yun, G. Zhang, X. Wang, Y. Wang, and B. Liu, "Keep forwarding path freshest in VANET via applying reinforcement learning," in *Proc. IEEE 1st Int. Workshop Netw. Meets Intell. Computations (NMIC)*, Jul. 2019, pp. 13–18.

[55] R. Li, F. Li, X. Li, and Y. Wang, "QGrid: Q-learning based routing protocol for vehicular ad hoc networks," in *Proc. IEEE 33rd Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2014, pp. 1–8.

[56] P.-J. Chuang and M.-C. Liu, "Advanced junction-based routing in vehicular ad-hoc networks," in *Proc. 9th Int. Conf. Future Gener. Commun. Netw. (FGCN)*, Nov. 2015, pp. 17–20.

[57] A. Soua and H. Afifi, "Adaptive data collection protocol using reinforcement learning for VANETs," in *Proc. 9th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jul. 2013, pp. 1040–1045.

[58] M. Saravanan and P. Ganeshkumar, "Routing using reinforcement learning in vehicular ad hoc networks," *Comput. Intell.*, vol. 36, no. 2, pp. 682–697, May 2020.

[59] S. Boussoufa-Lahlah, F. Semchedine, and L. Bouallouche-Medjkoune, "Geographic routing protocols for vehicular ad hoc NETworks (VANETs): A survey," *Veh. Commun.*, vol. 11, pp. 20–31, Jan. 2018.

[60] 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.

[61] Y. Sun, Y. Lin, and Y. Tang, "A reinforcement learning-based routing protocol in VANETs," in *Communications, Signal Processing, and Systems* (Lecture Notes in Electrical Engineering), vol. 463. Singapore: Springer, 2019, pp. 2493–2500, doi: 10.1007/978-981-10-6571-2_303.

[62] B. S. Roh, M. H. Han, J. H. Ham, and K. Il Kim, "Q-LBR: Q-learning based load balancing routing for UAV-assisted VANET," *Sensors*, vol. 20, no. 19, pp. 1–17, 2020.

[63] D. N. Patel, S. B. Patel, H. R. Kothadiya, P. D. Jethwa, and R. H. Jhaveri, "A survey of reactive routing protocols in MANET," in *Proc. Int. Conf. Inf. Commun. Embedded Syst. (ICICES)*, Feb. 2014, pp. 1–6.

[64] G. M. Valantina and S. Jayashri, "Q-learning based point to point data transfer in VANETs," *Procedia Comput. Sci.*, vol. 57, pp. 1394–1400, 2015.

[65] C. Wu, S. Ohzahata, and T. Kato, "Flexible, portable, and practicable solution for routing in VANETs: A fuzzy constraint Q-learning approach," *IEEE Trans. Veh. Technol.*, vol. 62, no. 9, pp. 4251–4263, Nov. 2013.

[66] J. Wu, M. Fang, and X. Li, "Reinforcement learning based mobility adaptive routing for vehicular ad-hoc networks," *Wireless Pers. Commun.*, vol. 101, no. 4, pp. 2143–2171, Aug. 2018.

[67] 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.

[68] C. Wu, Y. Ji, F. Liu, S. Ohzahata, and T. Kato, "Toward practical and intelligent routing in vehicular ad-hoc networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5503–5519, 2015.

[69] X. Yang, W. Zhang, H. Lu, and L. Zhao, "V2 V routing in VANET based on heuristic Q-learning," *Int. J. Comput. Commun.*, vol. 15, no. 5, pp. 1–17, Jul. 2020.

[70] J. Sucec and I. Marsic, "Hierarchical routing overhead in mobile ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 1, pp. 46–56, Jan. 2004.

[71] N. Kumar, N. Chilamkurti, and J. H. Park, "ALCA: Agent learning-based clustering algorithm in vehicular ad hoc networks," *Pers. Ubiquitous Comput.*, vol. 17, no. 8, pp. 1683–1692, Dec. 2013.

[72] Y. Ji, C. Wu, and T. Yoshinaga, "Context-aware unified routing for VANETs based on virtual clustering," in *Proc. IEEE 27th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Sep. 2016, pp. 8–13.

[73] X. Bi, D. Gao, and M. Yang, "A reinforcement learning-based routing protocol for clustered EV-VANET," in *Proc. IEEE 5th Inf. Technol. Mechatronics Eng. Conf. (ITOEC)*, Jun. 2020, pp. 1769–1773.

[74] C. Wu, T. Yoshinaga, Y. Ji, and Y. Zhang, "Computational intelligence inspired data delivery for vehicle-to-roadside communications," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12038–12048, Dec. 2018.

[75] A. Nahar and D. Das, "Adaptive reinforcement routing in software defined vehicular networks," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2020, pp. 2118–2123.

[76] A. M. Pushpa, "Trust based secure routing in AODV routing protocol," in *Proc. IEEE Int. Conf. Internet Multimedia Services Archit. Appl. (IMSAA)*, Dec. 2009, pp. 1–6.

[77] D. Zhang, F. R. Yu, R. Yang, and H. Tang, "A deep reinforcement learning-based trust management scheme for software-defined vehicular networks," in *Proc. 8th ACM Symp. Design Anal. Intell. Veh. Netw. Appl. (DIVANet)*, 2018, pp. 1–7.

[78] C. Dai, X. Xiao, Y. Ding, L. Xiao, Y. Tang, and S. Zhou, "Learning based security for VANET with blockchain," in *Proc. IEEE Int. Conf. Commun. Syst. (ICCS)*, Dec. 2018, pp. 210–215.

[79] L. Xiao, X. Lu, D. Xu, Y. Tang, L. Wang, and W. Zhuang, "UAV relay in VANETs against smart jamming with reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4087–4097, May 2018.

[80] D. Zhang, F. R. Yu, and R. Yang, "A machine learning approach for software-defined vehicular ad hoc networks with trust management," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[81] D. Zhang, F. R. Yu, and R. Yang, "Blockchain-based distributed software-defined vehicular networks: A dueling deep Q-learning approach," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 1086–1100, Dec. 2019.

[82] M. Y. Arafat and S. Moh, "Location-aided delay tolerant routing protocol in UAV networks for post-disaster operation," *IEEE Access*, vol. 6, pp. 59891–59906, 2018.

[83] C. Wu, T. Yoshinaga, D. Bayar, and Y. Ji, "Learning for adaptive anycast in vehicular delay tolerant networks," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 4, pp. 1379–1388, Apr. 2019.

[84] J. He, L. Cai, J. Pan, and P. Cheng, "Delay analysis and routing for two-dimensional VANETs using carry-and-forward mechanism," *IEEE Trans. Mobile Comput.*, vol. 16, no. 7, pp. 1830–1841, Jul. 2017.

[85] A. Boukerche, C. Rezende, and R. W. Pazzi, "A link-reliability-based approach to providing QoS support for VANETs," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2009, pp. 1–5.

[86] M. Y. Arafat, M. A. Habib, and S. Moh, "Routing protocols for UAV-aided wireless sensor networks," *Appl. Sci.*, vol. 10, no. 12, p. 4077, Jun. 2020.

[87] Y. Ohta, T. Ohta, E. Kohno, and Y. Kakuda, "A store-carry-forward-based data transfer scheme using positions and moving direction of vehicles for VANETs," in *Proc. 10th Int. Symp. Auto. Decentralized Syst. (ISADS)*, Mar. 2011, pp. 131–138.

[88] H. Saleet, R. Langar, K. Naik, R. Boutaba, A. Nayak, and N. Goel, "Intersection-based geographical routing protocol for VANETs: A proposal and analysis," *IEEE Trans. Veh. Technol.*, vol. 60, no. 9, pp. 4560–4574, Nov. 2011.

[89] N. Wisitpongphan, O. K. Tonguz, J. S. Parikh, P. Mudalige, F. Bai, and V. Sadekar, "Broadcast storm mitigation techniques in vehicular ad hoc networks," *IEEE Wireless Commun.*, vol. 14, no. 6, pp. 84–94, Dec. 2007.

[90] H. Gao, C. Liu, Y. Li, and X. Yang, "V2 VR: Reliable hybrid-network-oriented V2 V data transmission and routing considering RSUs and connectivity probability," *IEEE Trans. Intell. Transp. Syst.*, early access, Apr. 13, 2020, doi: 10.1109/TITS.2020.2983835.

[91] M. Y. Arafat, S. Poudel, and S. Moh, "Medium access control protocols for flying ad hoc networks: A review," *IEEE Sensors J.*, vol. 21, no. 4, pp. 4097–4121, Feb. 2021.

[92] C. Pu, "Jamming-resilient multipath routing protocol for flying ad hoc networks," *IEEE Access*, vol. 6, pp. 68472–68486, 2018.

[93] L. N. Balico, A. A. F. Loureiro, E. F. Nakamura, R. S. Barreto, R. W. Pazzi, and H. A. B. F. Oliveira, "Localization prediction in vehicular Ad Hoc networks," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2784–2803, 4th Quart., 2018.

[94] H. Zhu, R. Lu, X. Shen, and X. Lin, "Security in service-oriented vehicular networks," *IEEE Wireless Commun.*, vol. 16, no. 4, pp. 16–22, Aug. 2009.

[95] F. J. Martinez, C. K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni, "A survey and comparative study of simulators for vehicular ad hoc networks (VANETs)," *Wireless Commun. Mobile Comput.*, vol. 11, no. 7, pp. 813–828, Jul. 2011.

[96] F. J. Martinez, M. Fogue, C. K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni, "Computer simulations of VANETs using realistic city topologies," *Wireless Pers. Commun.*, vol. 69, no. 2, pp. 639–663, Mar. 2013.

[97] M. Haklay and P. Weber, "OpenStreet map: User-generated street maps," *IEEE Pervasive Comput.*, vol. 7, no. 4, pp. 12–18, Oct. 2008.

[98] E. Spaho, M. Ikeda, L. Barolli, F. Xhafa, V. Kolici, and M. Takizawa, "Performance evaluation of OLSR protocol in a grid manhattan VANET scenario for different applications," in *Proc. 7th Int. Conf. Complex, Intell., Softw. Intensive Syst. (CISIS)*, Jul. 2013, pp. 47–52.

[99] K.-C. Lan and C.-M. Chou, "Realistic mobility models for vehicular ad hoc network (VANET) simulations," in *Proc. 8th Int. Conf. ITS Telecommun.*, Oct. 2008, pp. 362–366.

[100] J. Härri, M. Fiore, F. Filali, and C. Bonnet, "Vehicular mobility simulation with VANETMobiSim," *Simulation*, vol. 87, no. 4, pp. 275–300, Apr. 2011.

[101] F. Bai, N. Sadagopan, and A. Helmy, "IMPORTANT: A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks," in *Proc. IEEE INFOCOM 22nd Annu. Joint Conf. IEEE Comput. Commun. Soc.*, Mar. 2003, pp. 825–835.

[102] H. C. Lau, M. Sim, and K. M. Teo, "Vehicle routing problem with time windows and a limited number of vehicles," *Eur. J. Oper. Res.*, vol. 148, no. 3, pp. 559–569, Aug. 2003.

[103] M. Bowling and M. Veloso, "Multiagent learning using a variable learning rate," *Artif. Intell.*, vol. 136, no. 2, pp. 215–250, Apr. 2002.

[104] S. M. Abuelenin and A. Y. Abul-Magd, "Empirical study of traffic velocity distribution and its effect on VANETs connectivity," in *Proc. Int. Conf. Connected Vehicles Expo (ICCVE)*, 2014, pp. 391–395.

[105] R. Adrian, S. Sulistyo, I. W. Mustika, and S. Alam, "MRV-M: A cluster stability in highway VANET using minimum relative velocity based on K-medoids," in *Proc. 5th Int. Conf. Sci. Technol. (ICST)*, Jul. 2019, pp. 1–5.

[106] S. Goli-Bidgoli and N. Movahhedinia, "Determining vehicles' radio transmission range for increasing cognitive radio VANET (CR-VANET) reliability using a trust management system," *Comput. Netw.*, vol. 127, pp. 340–351, Nov. 2017.

[107] M. Naresh, A. Raje, and K. Varsha, "Link prediction algorithm for efficient routing in VANETs," in *Proc. 3rd Int. Conf. Comput. Methodologies Commun. (ICCMC)*, Mar. 2019, pp. 1156–1161.

[108] H. Menouar, F. Filali, and M. Lenardi, "A survey and qualitative analysis of mac protocols for vehicular ad hoc networks," *IEEE Wireless Commun.*, vol. 13, no. 5, pp. 30–35, Oct. 2006.

[109] T. M. Cook and R. A. Russell, "A simulation and statistical analysis of stochastic vehicle routing with timing constraints," *Decis. Sci.*, vol. 9, no. 4, pp. 673–687, Oct. 1978.

[110] E. M. Van Eenennaam, "A survey of propagation models used in vehicular ad hoc network (VANET) research," Course Mobile Radio Commun., Univ. Twente, Enschede, The Netherlands, Tech. Rep., 2009, pp. 1–7.

[111] Z. Mammeri, "Reinforcement learning based routing in networks: Review and classification of approaches," *IEEE Access*, vol. 7, pp. 55916–55950, 2019.

[112] L. Busoniu, R. Babuška, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 2, pp. 156–172, Mar. 2008.

[113] Y. Saleem, K.-L.-A. Yau, H. Mohamad, N. Ramli, M. H. Rehmani, and Q. Ni, "Clustering and reinforcement-learning-based routing for cognitive radio networks," *IEEE Wireless Commun.*, vol. 24, no. 4, pp. 146–151, Aug. 2017.

[114] N. Marchang and R. Datta, "Light-weight trust-based routing protocol for mobile ad hoc networks," *IET Inf. Secur.*, vol. 6, no. 2, pp. 77–83, Jun. 2012.

[115] 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.

[116] A. Abuashour and M. Kadoch, "Control overhead reduction in cluster-based VANET routing protocol," in *Ad Hoc Networks*. Cham, Switzerland: Springer, 2018, pp. 106–115.

[117] K. Abboud and W. Zhuang, "Impact of microscopic vehicle mobility on cluster-based routing overhead in VANETs," *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5493–5502, Dec. 2015.

[118] A. K. Kalakanti, S. Verma, T. Paul, and T. Yoshida, "RL SolVeR pro: Reinforcement learning for solving vehicle routing problem," in *Proc. 1st Int. Conf. Artif. Intell. Data Sci. (AiDAS)*, Sep. 2019, pp. 94–99.

[119] S. D. Whitehead, "Complexity and cooperation in Q-learning," in *Machine Learning Proceedings*. Amsterdam, The Netherlands: Elsevier, 1991, pp. 363–367.

[120] N. Kantasewi, S. Marukatat, S. Thainimit, and O. Manabu, "Multi Q-table Q-learning," in *Proc. 10th Int. Conf. Inf. Commun. Technol. Embedded Syst. (IC-ICTES)*, 2019, pp. 1–7.

[121] A. M. Nagy and V. Simon, "Survey on traffic prediction in smart cities," *Pervasive Mobile Comput.*, vol. 50, pp. 148–163, Oct. 2018.

[122] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying deep learning approaches for network traffic prediction," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 2353–2358.

[123] Q. Wei, F. L. Lewis, Q. Sun, P. Yan, and R. Song, "Discrete-time deterministic *Q*-learning: A novel convergence analysis," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1224–1237, May 2017.

[124] T. Jiang, D. Grace, and P. D. Mitchell, "Efficient exploration in reinforcement learning-based cognitive radio spectrum sharing," *IET Commun.*, vol. 5, no. 10, pp. 1309–1317, Jul. 2011.

[125] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, "VANet security challenges and solutions: A survey," *Veh. Commun.*, vol. 7, pp. 7–20, Jan. 2017.

[126] Y. Chen, S. Huang, F. Liu, Z. Wang, and X. Sun, "Evaluation of reinforcement learning-based false data injection attack to automatic voltage control," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 2158–2169, Mar. 2019.

[127] M. S. Sheikh and J. Liang, "A comprehensive survey on VANET security services in traffic management system," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–23, Sep. 2019.

[128] A. Mchergui, T. Moulahi, B. Alaya, and S. Nasri, "A survey and comparative study of QoS aware broadcasting techniques in VANET," *Telecommun. Syst.*, vol. 66, no. 2, pp. 253–281, Oct. 2017.

[129] W. Tong, A. Hussain, W. X. Bo, and S. Maharjan, "Artificial intelligence for vehicle-to-everything: A survey," *IEEE Access*, vol. 7, pp. 10823–10843, 2019.

[130] R. K. Jaiswal and C. D. Jaidhar, "PPRP: Predicted position based routing protocol using Kalman Filter for vehicular Ad-Hoc network," in *Proc. ACM Int. Conf. Proc.*, 2017, pp. 1–8.

[131] C. H. Lee, K. G. Lim, B. L. Chua, R. K. Y. Chin, and K. T. K. Teo, "Progressing toward urban topology and mobility trace for vehicular ad hoc network (VANET)," in *Proc. IEEE Conf. Open Syst. (ICOS)*, Oct. 2016, pp. 120–125.

[132] H. Ahmed, S. Pierre, and A. Quintero, "A flexible testbed architecture for VANET," *Veh. Commun.*, vol. 9, pp. 115–126, Jul. 2017.

[133] M. Naderi, F. Zargari, and M. Ghanbari, "Adaptive beacon broadcast in opportunistic routing for VANETs," *Ad Hoc Netw.*, vol. 86, pp. 119–130, Apr. 2019.

**REZOAN AHMED NAZIB** received the B.Sc. degree in computer science from BRAC University, Bangladesh, in 2017. He is currently pursuing the M.Sc. degree with the Mobile Computing Laboratory, Chosun University, South Korea. His current research interests include ad hoc networks and unmanned aerial networks with a focus on network architectures and protocols.

**SANGMAN MOH** (Member, IEEE) received the M.S. degree in computer science from Yonsei University, South Korea, in 1991, and the Ph.D. degree in computer engineering from the Korea Advanced Institute of Science and Technology (KAIST), South Korea, in 2002. Since late 2002, he has been a Professor with the Department of Computer Engineering, Chosun University, South Korea. From 2006 to 2007, he was on leave at Cleveland State University, Cleveland, OH, USA. Since then, he has also been working with the Electronics and Telecommunications Research Institute (ETRI), South Korea, as a Project Leader, until 2002. His research interests include mobile computing and networking, ad hoc and sensor networks, cognitive radio networks, unmanned aerial vehicle networks, and parallel and distributed computing systems. He is a member of ACM, IEICE, KIISE, IEIE, KIPS, KICS, KMMS, IEMEK, KISM, and KPEA.