mathematics

Review

Reinforcement Learning-Based Routing Protocols in Vehicular Ad Hoc Networks for Intelligent Transport System (ITS): A Survey

Jan Lansky [1], Amir Masoud Rahmani [2,*] and Mehdi Hosseinzadeh [3,*]

[1] Department of Computer Science and Mathematics, Faculty of Economic Studies, University of Finance and Administration, 101 00 Prague, Czech Republic
[2] Future Technology Research Center, National Yunlin University of Science and Technology, Yunlin, Douliou 64002, Taiwan
[3] Pattern Recognition and Machine Learning Lab, Gachon University, 1342 Seongnamdaero, Sujeonggu, Seongnam 13120, Republic of Korea
* Correspondence: rahmania@yuntech.edu.tw (A.M.R.); mehdi@gachon.ac.kr (M.H.)

Abstract: Today, the use of safety solutions in Intelligent Transportation Systems (ITS) is a serious challenge because of novel progress in wireless technologies and the high number of road accidents. Vehicular ad hoc network (VANET) is a momentous element in this system because they can improve safety and efficiency in ITS. In this network, vehicles act as moving nodes and work with other nodes within their communication range. Due to high-dynamic vehicles and their different speeds in this network, links between vehicles are valid for a short time interval. Therefore, routing is a challenging work in these networks. Recently, reinforcement learning (RL) plays a significant role in developing routing algorithms for VANET. In this paper, we review reinforcement learning and its characteristics and study how to use this technique for creating routing protocols in VANETs. We propose a categorization of RL-based routing schemes in these networks. This paper helps researchers to understand how to design RL-based routing algorithms in VANET and improve the existing methods by understanding the challenges and opportunities in this area.

Keywords: vehicular ad hoc network (VANET); reinforcement learning (RL); artificial intelligence (AI); machine learning (ML); wireless networks

MSC: 68M18

check for updates

Citation: Lansky, J.; Rahmani, A.M.; Hosseinzadeh, M. Reinforcement Learning-Based Routing Protocols in Vehicular Ad Hoc Networks for Intelligent Transport System (ITS): A Survey. *Mathematics* **2022**, *10*, 4673. https://doi.org/10.3390/math10244673

Academic Editor: Daniel-Ioan Curiac

Received: 16 October 2022
Accepted: 5 December 2022
Published: 9 December 2022

1. Introduction

Intelligent transport system (ITS) has a great contribution to modern life. This system offers new services to control adverse events such as road accidents and improve traffic management [1,2]. Rapid progress in wireless communication technologies helps create such a system because vehicles equipped with these wireless technologies can efficiently make connection links with other vehicles as well as roadside units (RSUs). Vehicular ad hoc network (VANET) aims to provide these wireless connections between network nodes (i.e., vehicles or roadside infrastructures). Recently, this network has attracted the attention of researchers because of its potential role in ITS. VANET has many applications such as passenger safety improvement, traffic efficiency optimization, autonomous driving, access to Internet of vehicles (IoV), collecting real-time data to control traffic and road protection systems, paying road tolls automatically, and entertainment applications [3,4]. VANETs have specific features such as frequent disconnections, dynamic topology, and moving nodes. Table 1 summarizes these features for vehicular ad hoc networks.

Table 1. Important features of vehicular ad hoc networks.

Feature	Description
Dynamic mobility scenarios	In this network, the moving nodes have dynamic mobility scenarios, which are different from low velocity (below 50 km/h) to high velocity (above 500 km/h). Therefore, this reduces the lifetime of the discovered paths between the nodes.
Dynamic topology	In VANET, network topology changes rapidly, the lifetime of the connection links is short, and the network density varies depending on the location and time.
Frequent failure of links	Inadequate climate conditions, along with the high velocity of the moving nodes and the repeated topology changes, cause the failure of links between the nodes.
Energy and computing resources	In this network, nodes do not have energy restrictions because every vehicle is equipped with a battery with long-lifetime. However, this is a challenge in real-time environments.
Predictability	The movement of vehicles is restricted by urban maps. Therefore, their movement is predictable.
Different quality of services (QoS) requirements	In VANET, different data services, like multimedia entertainment and video games have different QoS needs such as reliability, delay, and data rates.
Large-scale networks	A large number of vehicles participate in VANET. Therefore, they create a large-scale network, particularly in dense urban areas, city centers, entrances to large cities, and highways.
Various network densities	Changing traffic flow causes changes in the network density, meaning that in some areas, such as rural areas, the network density is very low or in some areas such as city centers, gridlock, or heavy traffic, the network density is very high.

Designing an efficient routing approach is a serious issue in VANET [5,6]. Routing protocols are responsible for determining paths between source-destination pairs [7]. Also, when breaking the discovered paths, routing protocols are responsible for forming an alternative route. In such a case, if the routing path is not properly selected, it diminishes network performance. Path efficiency is measured based on the participation of nodes in the data transmission process. Due to very dynamic topology and high-speed vehicles, efficient routing solutions are a serious challenge in these networks. Therefore, many researchers have attempted to modify the existing routing schemes in VANETs. Despite many efforts in this regard, routing protocols are still vulnerable and are not complete.

Recently, machine-learning (ML) is a new field originated from artificial intelligence (AI) that includes efficient and strong techniques [8,9]. They can be applied to integrate autonomous decision-making systems in vehicular ad hoc networks to solve their various challenges and issues such as routing. ML can produce more intelligent machines trained on past experiences without human interference. This means that they do not need explicit programming. Machine learning involves three branches: supervised, unsupervised, and reinforcement learning. The first class (i.e., supervised learning) consists of an input dataset and corresponding outputs (i.e., labels). Techniques related to this class seek to form a learning model to explore the relationship between data samples and labels and produce a function to map the data to the labels. This model is used for predicting unlabeled data. In unsupervised learning, there is no output related to the inputs, meaning that the data is unlabeled. Unsupervised learning must find the existing patterns and relationships between data samples. In reinforcement learning (RL), the agent and the dynamic environment work in relation to each other. These interactions determines the ideal behavior of the agent with regard to the reward-penalty produced by environment [10]. In VANET, ML techniques, especially RL, try to make vehicles take self-decisions for networking operations such as routing [11,12]. The agent must obtain knowledge in relation to the environment dynamics based on the collected data to find the most suitable action and achieve a certain purpose, like discovering routes with minimum delay. RL can be used to optimize various issues in VANET such as predicting traffic conditions, estimating network traffic, controlling network congestion, discovering routes, enhancing network security, and resource allocation.

Reinforcement learning algorithms are attractive solutions for modifying routing methods in VANET. However, researchers need more research in this area because machine learning, especially reinforcement learning is a significant research subject in VANETs. Note that most review papers related to machine learning and VANET do not focus on RL applications in designing routing protocols. For example, in [13], authors have reviewed various routing methods based on RL in VANETs. Ref. [14] studied various applications of RL and DRL in vehicular network management, but did not consider their applications for improving routing approaches. In [15], authors investigated the importance of artificial intelligence techniques in different areas of VANET, especially routing. However, they do not well explain how they use reinforcement learning techniques for improving vehicular communication. In [16], authors have studied different RL and DRL applications in various Internet of things (IoT) systems. In [17], authors have examined how to use multi-agent reinforcement learning techniques in different VANET applications such as resource allocation, caching, and data offloading. Overall, our studies show that few review papers are presented in the field of RL applications for designing routing schemes in VANETs. Thus, this important issue requires further research to better identify future research directions and their challenges. We believe that our survey can help researchers to understand how to create routing protocols based on RL in VANET. In this review, we propose a categorization of RL-based routing schemes with regard to learning framework (single (or multiple) agent(s)), learning model (model-based and free-model), learning algorithm (RL or DRL), learning process (centralized and distributed), and routing algorithm (position-based, cluster-based, topology-based (proactive, reactive, and hybrid)). Then, we present the latest routing approaches according to the proposed classification.

The organization of this paper includes several sections: Section 2 expresses several review papers in this area. Section 3 reviews reinforcement learning and Markov decision process in summary. In Section 4, VANETs and their applications are introduced briefly. In this section, we focus on the routing operation and its issues in VANETs. Section 5 proposes a categorization for RL-based schemes in VANETs. In Section 6, several RL-based schemes are investigated in VANETs. Section 7 presents a discussion of RL-based routing methods. Section 8 demonstrates the major challenges and open issues in this area. Ultimately, in Section 9, the conclusion is stated.

2. Related Works

Today, researchers study on machine learning, especially reinforcement learning because it is a significant research subject in VANETs. Table 2 summarizes some review papers in this field. Note that most review papers related to machine learning and VANET do not focus on RL applications in designing routing protocols.

Table 2. Some review articles related to RL applications in VANETs.

Review Paper	Publication Year	Classification	Application	Network
[13]	2021	Based on the routing algorithm, including hybrid, zone-based, geographical, topology-based, hierarchical, secure, and DTN	Routing protocols	VANET
[14]	2021	Based on application	Vehicular network management	VANET
[15]	2021	Based on three AI techniques, including machine learning methods, deep learning, and swarm intelligence	Routing, security, resource and access technologies, and mobility management	VANET
[16]	2021	Based on seven applications	Routing, scheduling, resource allocation, dynamic spectrum access, energy, mobility, and edge caching	IoT systems
[17]	2019	Resource allocation, caching and data offloading	Streaming applications and mission-critical applications	VANET
[18]	2022	Based on the learning algorithm, the routing algorithm, and the data dissemination process	Routing protocols	FANET
Our survey	2022	Based on learning framework, learning model, learning algorithm, learning process, and routing algorithm	Routing protocols	VANET

In [13], authors have reviewed different RL-based routing protocols in VANETs. In this paper, authors claim that their survey is the first review paper, which has analyzed RL-based routing algorithms in VANETs. It is a comprehensive review and is very suitable for researchers in this field. In [13], routing methods are divided into seven categories, including hybrid, zone-based, geographical, topology-based, hierarchical, secure, and DTN. However, this category is very limited and does not evaluate the routing algorithms in terms of learning structure and RL algorithm.

In [14], authors have studied the RL and DRL applications in vehicular network management. Firstly, they have introduced vehicular ad hoc networks. Then, they review the RL and DRL concepts. Finally, they have carefully studied the newest applications of these learning techniques in two different areas: vehicular resource management and vehicular infrastructure management. Note that this paper emphasizes vehicular network management using Rl approaches and does not investigate the use of these approaches for improving the routing schemes.

In [15], authors have examined the importance of artificial intelligence (AI) techniques in various fields of VANETs. In this paper, they have briefly explained three AI techniques, including machine learning methods (especially, RL), deep learning (especially DRL), and swarm intelligence. Then, they have studied various AI techniques to solve different challenges in VANETs. They are carefully examined in six areas including application, routing, security, resource and access technologies, mobility management, and architecture. However, the authors do not well explain how to use reinforcement learning techniques for improving vehicular communication.

In [16], authors have reviewed reinforcement learning techniques and deep reinforcement learning techniques in various IoT systems including wireless sensor networks (WSNs), wireless body area networks (WBANs), underwater wireless sensor networks (UWSNs), Internet of vehicles (IoV), and Industrial Internet of things (IIoT). Then, they have divided them into seven different categories: routing, scheduling, resource allocation, dynamic spectrum access, energy, mobility, and caching. However, RL-base and DRL-based routing methods are only investigated in wireless sensor networks.

In [17], authors have examined multi-agent reinforcement learning (MARL) techniques to solve various problems in VANET. In this paper, various research works are focused on resource allocation, caching and data offloading in VANET. Also, the authors have explained how to use MARL techniques in streaming applications and mission-critical applications. Finally, they have presented the challenges related to these systems in VANETs. However, this paper does not focus on the MARL applications for designing routing protocols in VANETs.

In [18], the authors have investigated how to apply reinforcement learning (RL) to build routing approaches in flying ad hoc networks (FANETs). For this purpose, they explained these networks, their constraints, main components, especially drones, and applications in different fields and specified the routing challenges in these networks in detail. Finally, a classification of routing approaches was presented. It includes three main fields, namely learning algorithm, routing algorithm and data dissemination process. According to the presented classification, the latest RL-based routing approaches in FANET have been reviewed.

Overall, our studies show that few review papers are presented in the field of RL applications for designing routing schemes in VANETs. Thus, we focus on the Rl-based routing protocols in VANETs and review their learning structure. Additionally, we propose a categorization of RL-based routing schemes with regard to learning framework (single (or multiple) agent(s)), learning model (model-based and free-model), learning algorithm (RL and DRL), learning process (centralized and distributed), and routing algorithm (position-based, cluster-based, topology-based (proactive, reactive, and hybrid)).

3. Overview of Reinforcement Learning (RL)

Here, we explain RL, Markov decision framework, and their features in summary.

3.1. Reinforcement Learning

According to this technique, an agent finds the most suitable policy by interacting with the environment. In each iteration, the agent selects an action a_t in accordance with its state (s_t), gets reward r_t from the environment, and shifts to the latter state s_{t+1}. This process is continues until the agent accumulates the rewards received from the environment and maximizes the expected discounted return from any situation [19]. A reinforcement learning system consists of four main parts:

- **Policy:** It indicates a set of stimulus-action rules. According to these rules, each state is mapped to a group of actions that can be chosen by the agent. The simplest case is to implement this policy by a lookup table. However, it has high computational complexity for search processes [19,20].
- **Reward signal:** It can be a random function that indicates the response of environment proportional to the action and state of the agent. After responding to actions, the environment creates a new state of agent and reward. The agent attempts to maximize the total reward resulting from relationship between itself and environment. Reward is an important element for improving the policy. If the selected action causes a weak reward, the agent can select another action in the same condition in the future to get other possibilities [19,20].
- **Value function:** Note that the reward signal indicates whether the last performed action is good, while the value function calculates the value of a specific state with regard to the sum of rewards collected by the agent when placing in that state. Thus, actions are chosen in accordance with the highest values, not the maximum rewards. Note that the computation of values is much more complicated than rewards because the agent obtains rewards immediately from the environment. However, values must be predicted by searching for past interactions between the agent and the environment. Value estimation is a challenging issue in all RL algorithms [19,20].
- **Model:** It describes the environment's function. This means that model can predict the latter state and the obtained reward by looking and choosing one of the allowed

actions in a particular state. According to this concept, there are two model-based and model-free RL techniques. In model-based approaches, a model is created to solve the desired issue. In contrast, model-free RL approaches utilize a trial and error methodology to learn the most suitable policy [19,20].

In this area, there are various challenges that are expressed as follows:

- **Exploration and exploitation:** It is an important challenge in RL problems. Exploration means searching for unknown actions to acquire new knowledge. On the other hand, exploitation means utilizing the discovered actions that can produce high feedback. The agent can perform both operations. This means that it can perform the exploitation operation based on its current knowledge and obtains the suitable value. Also, it can perform the exploration operation to learn actions that have never been experienced so far to enhance its knowledge and earn better rewards in the future. Thus, this is not correct to focus only on exploration or extraction. The agent needs to experience different actions to gradually select those with the maximum value [21].
- **Uncertainty:** It is another challenge in RL problems because the agent and the environment work in relation to each other, and this interaction can cause uncertainty when changing the state and obtaining the reward. Whereas, a RL problem must learn a policy that increases the collected rewards (value function) over time [21].

3.2. Markov Decision Framework

Reinforcement learning formulates the desired issue as a Markov decision process (MDP) [22,23]. Definitions related to reinforcement learning are presented below:

Definition 1. *MDP consists of a tuple* (S, A, P, R, γ)*:*

- *S: A limited state space*
- *A: A limited action space*
- *R: A reward function, like* $R = \mathbb{E}[R_{t+1}|S_t = s, A_t = a]$.
- *P: A state transition probability, like* $P = \mathbb{P}[S_{t+1} = s'|S_t = s, A_t = a]$.
- γ*: Discount factor so that* $\gamma \in [0, 1]$.

When resolving such a problem, the subsequent state is obtained from the former state. This means that it is not dependent on past states.

$$\mathbb{P}[S_{t+1} - S_t] = \mathbb{P}[S_{t+1} - S_1, ..., S_t] \tag{1}$$

The action and state sets and environment dynamics define a finite MDP. Equation (2) *defines the probability of each state-reward pair:*

$$p(s', r|s, a) \doteq \Pr\{S_{t+1} = s, R_{t+1} = r|S_t = s, A_t = a\} \tag{2}$$

where (s, a) *and* (s', r) *indicate the former state-action and latter state-reward pairs, respectively.*

The sum of the collected rewards (i.e., $G_t = R_{t+1} + R_{t+2} + \cdots + R_T$ *so that T is the final time step) must be maximized by the agent. This function can be used when the task is episodic. This means that it has a final state. However, if the task is continuous, then there is no final state, i.e.,* $T = \infty$.

Definition 2. G_t *(obtained from Equation* (3)*) is the total long-term returns after considering the discount factor.*

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{3}$$

so that γ^k *(i.e., discount factor) is between* $[0, 1]$. R_{t+k+1} *also reflects the reward at time* $t + k + 1$.

Definition 3. *A probability distribution called the policy π is defined with regard to the actions taken for the existing states. In fact, once the agent reaches a state, it chooses the subsequent action with regard to the policy.*

$$\pi(a|s) \doteq \mathbb{P}[A_t = a|S_t = s] \tag{4}$$

so that a is the action selected for the state s.

Definition 4. *Assuming the agent in state s and listening the policy π, State-Value function $v_\pi(s)$ is defined according to Equation (5):*

$$v_\pi \doteq \mathbb{E}_\pi[G_t|S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s\right] \tag{5}$$

Definition 5. *Assuming the agent in state s, taking the action a, and listening the policy π, Action-Value function $q_\pi(s,a)$ is defined according to (6):*

$$q_\pi(s,a) \doteq \mathbb{E}_\pi[G_t|S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s, A_t = a\right] \tag{6}$$

The Bellman equation is met by both value functions (i.e., $v_\pi(s)$ and $q_\pi(s,a)$):

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_\pi(s')\right], \ \forall \ s \in S \tag{7}$$

When a RL algorithm discovers the best policy π^*, it is converged. In this case, two optimal state-value and action-value functions are calculated in accordance with the optimal policy π^*.

Definition 6. *Optimal state-value function $v_*(s)$ represents the best value of $v_\pi(s)$ at all policies.*

$$v_*(s) = \max_\pi v_\pi(s), \ \forall \ s \in S \tag{8}$$

Definition 7. *Optimal action-value function $q_*(s,a)$ represents the best value of $q_\pi(s,a)$ at all policies.*

$$q_*(s,a) = \max_\pi q_\pi(s,a), \ \forall \ s \in S \tag{9}$$

See [10,22] for more details about RL techniques.

4. Vehicular Ad Hoc Networks

Vehicular ad hoc networks (VANETs) include moving entities (vehicles) and fixed entities (roadside units). These entities work together to share important traffic information about roads [14,24]. The emergence of new technologies such as 5G mobile networks has provided new services for VANET to connect vehicles with everything (V2X). In this regard, the 3rd Generation Partnership Project (3GPP) has proposed the new radio V2X standard (NR-2X) for SideLink (SL) communication to directly communicate different items such as vehicles and personal devices without connecting to RSUs [25]. 3GPP mainly emphasizes high reliability, maximum coverage, low delay, and energy-saving, especially for battery-based equipment. According to Figure 1, 3GPP supports four modes of V2X connections: vehicle-to-vehicle (V2V), vehicle-to-pedestrian (V2P), vehicle-to-infrastructure (V2I), and vehicle-to-network (V2N).

- V2V and V2P provide the direct connection between vehicles-user equipment (UE) and as well as between vehicles and vulnerable road users (VRU) such as cyclists, bikers, and wheelchairs.
- V2I focuses on the connection between vehicles and road infrastructure.
- V2N allows UEs to communicate with a server.

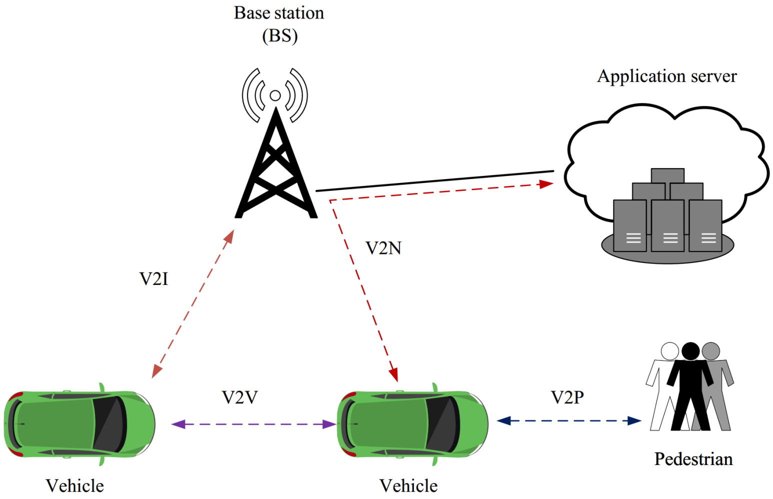

Figure 1. Vehicular ad hoc networks (VANETs).

4.1. Applications

In this section, we introduce some VANET applications:

- **Traffic and safety efficiency:** Exchanging beacon and event-driven messages in V2V and V2P communication allows other vehicles and road users to control the environment. These messages include the location of sender vehicles and systematic parameters. For example, when an accident warning is sent by the driver to inform other drivers [26,27].
- **Autonomous driving:** For accurate routing by smart vehicles, they must be aware of their location, the surrounding environment, and the neighboring vehicles because they move at high speeds (i.e., above 200 km/h).
- **Tele-operated driving:** This application is used in dangerous environments for example, nuclear events, earthquakes, snowfall, and road construction. In this case, the driver is out of the vehicle and controls his driving operations using a camera and sensor data.
- **Entertainment and Internet of Vehicles (IoV):** This application provides comfortable services for drivers and pedestrians such as access to mobile Internet, messengers, dialog between vehicles, and collaborative games on the network.

4.2. Routing

Routing means the formation of a route between two vehicles called the source node and the destination node. In VANETs, routing protocols ensures that both two nodes can exchange their data packets with each other. These protocols are responsible for determining a suitable route between the source to the destination for sending data packets. Also, they are responsible to find an alternative route when failing routes. In such a case, if the routing path is not properly selected, the network performance will be severely weakened because existing links are constantly disconnected and new links are created. A routing approach must choose the most stable route to enhance network performance and reduce the need to rebuild existing paths [28,29]. VANETs have specific features such as repeated change in topology and high-speed vehicles, which have challenged the routing process [30,31]. In the following, we express some of these features:

4.2.1. Highly Dynamic Topology

In VANET, unlike other ad hoc networks or sensor networks, the network topology is very dynamic because of the speed of vehicles in the network. This means that they can directly communicate with each other for a short time interval. This shows that it is very difficult to make a communication link between them.

4.2.2. Frequent Disconnection of Network

In VANET, vehicles are moving. This changes the network density and leads to breaking the communication links repeatedly. This phenomenon often occurs in the low-density network areas and in when there are radio obstacles in the network because in these areas, there is a repeated disconnection that causes a high rate of link breakage, high delay in the data transmission, or even data loss. To ensure the quality of the connections, an efficient routing protocol must quickly detect the link failure and find an alternative link [30,32].

4.2.3. Mobility Modeling and Predicting

To create a path between vehicles, we must acquire information about the position of nodes and their movement pattern. However, it is difficult to predict their movement because vehicles have different movement patterns. Thus, designing a mobility model based on a predefined road model including network traffic, speed of nodes, and their behavior is very essential for creating an efficient routing protocol in the network. In this case, a traffic simulator can be beneficial when there are no real vehicular traces. However, this traffic simulator can affect the network performance. When researchers use the real vehicular traces in their simulation, the packet delivery rate is significantly lowered in comparison with the results of unrealistic traces. Therefore, when the mobility modes are closer to the real model, evaluation results of the routing protocol are more reliable [31,33].

4.2.4. Propagation Model

Researchers should not consider the propagation model as a free space in VANET because this network includes buildings, trees, and other vehicles that can play the role of obstacles. Also, researchers must regard the interference of wireless connections related to other vehicles or personal access points when selecting the propagation model in VANET [30,34].

4.2.5. Communication Environment

In this network, the communication environment for vehicles is limited to a road infrastructure on highways or urban area. The movement pattern on highways is different from that in urban environments. Highway environments include a long straight line in which vehicles are moving at high-speed. In contrast, urban environments include a large number of streets, intersections, and obstacles in which vehicles are moving at medium speed. These environments have different effects on vehicle-to-vehicles communication [31].

4.2.6. Delay

In VANET, various applications have different requirements. For example, an application such as safety warning applications may not need excellent data rate, but it is severely restricted in terms of delay. In this case, if an alarm message reaches the destination, too late and after a long period of time, it may not be effective to keep vehicles from happening accidents or overturn. Therefore, it is very important to provide acceptable time delay when designing routing protocols [28].

4.2.7. Bandwidth

In VANET, there is no central coordinator that manages bandwidth consumption and messages sent in the network. Therefore, designing a routing protocol in this network is very challenging, especially in areas with high density because network congestion is high.

4.2.8. Quality of Service (QoS)

It includes a set of services, which must be considered when transmitting data. Due to features of VANET, it is very challenging to support the quality of services. Each application has unique QoS requirements. Therefore, it is necessary to create adaptive QoS routing methods to quickly form updated paths when breaking old paths. Note that the link breakage is due to the change in the speed and position of vehicles and dynamic network topology [26].

5. Proposed Classification

Here, we illustrate our classification of Rl-based routing schemes. It includes the following items:

- Learning framework (single-agent and multi-agent)
- Learning model (model-based and free-model)
- Learning algorithm (traditional reinforcement learning and deep reinforcement learning)
- Learning process (centralized and distributed)
- Routing algorithm (position-based (i.e., delay-tolerant networking (DTN) and non-delay tolerant networking (Non-DTN)), cluster-based, topology-based (proactive, reactive, and hybrid))

In the following, our suggested classification is shown in Figure 2.

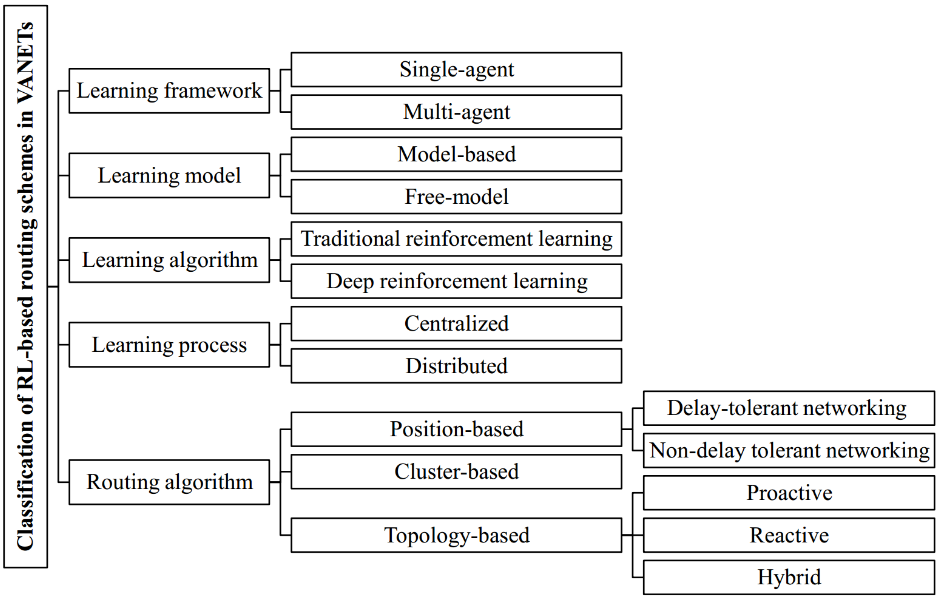

Figure 2. Our classification for RL-based routing approaches in VANETs.

5.1. Classification of RL-Based Routing Schemes with Regard to Learning Framework

According to the mentioned classification, RL-based routing approaches can be categorized into single-agent and multi-agent with regard to the learning framework. Table 3 summarizes the advantages and disadvantages of these approaches.

- **Single-agent RL-based routing approach:** In this type of routing scheme, an agent alone interacts with the network (i.e., learning environment) to learn its best behavior (i.e., the most suitable path) and maximize the rewards obtained from the environment. In this case, the routing methods utilize a single-agent RL system, as shown in Figure 3. In these routing protocols, if the learning environment is complex (i.e., the network is large-scale or includes a very high number of states and actions), the learning capability of the agent is greatly reduced, and the agent requires a longer time to earn the most suitable response. Thus, in this case, the agent suffers from a low convergence rate and may never get an optimal response. However, implementing a single-agent scheme is much easier than the multi-agent method, and it has a low computational cost [35,36].

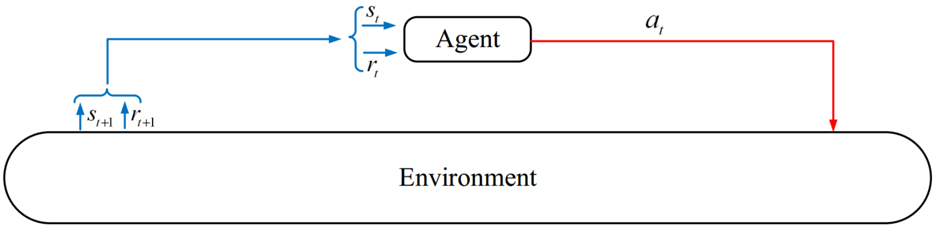

Figure 3. A single-agent system.

- **Multi-agent RL-based routing method:** In this type of routing method, several agents (for example, vehicles in the network) interact with the network environment to learn optimal behavior (i.e., the best path). In this case, the routing schemes use a multi-agent RL system. See Figure 4. In these protocols, environment dynamics can be affected by the behavior of other agents on the network. As a result, it is very challenging to coordinate agents with each other in multi-agent RL-based routing protocols. However, these methods are beneficial when agents have a connection link with each other to share their experiences. In this case, the learning ability of the multi-agent systems will be extremely accelerated because agents perform the computing process using a parallel manner [35,36]. The important advantage of multi-agent routing protocols is that they are fault-tolerance. These protocols are more suitable for the complex environment and have high learning strength. However, they suffer from high computational costs in comparison with single-agent systems so if the number of states and actions increases, then these multi-agent systems experience an exponential increase in their computational complexity [35].

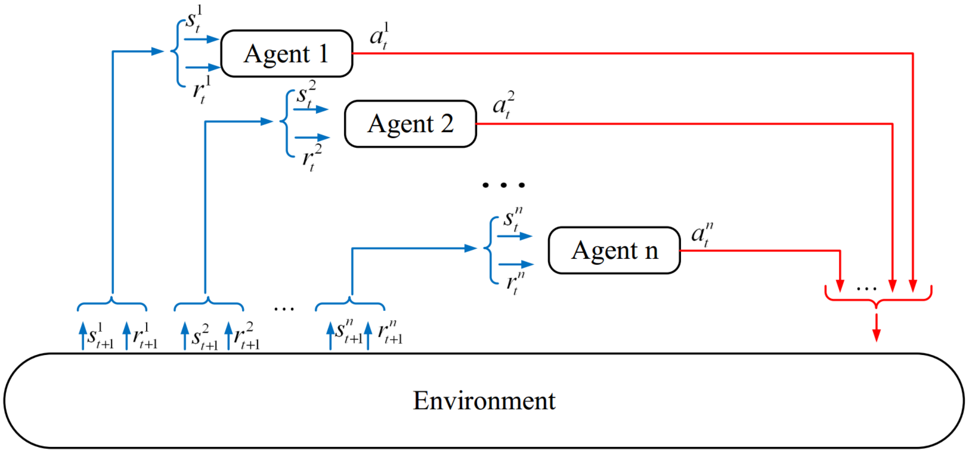

Figure 4. A multi-agent system.

Table 3. Single-agent and multi-agent routing.

Routing Scheme	Advantages	Disadvantages
Single-agent	Simplicity, low computational complexity	Inappropriate for large-scale VANETs, low convergence speed, low fault tolerance, low learning ability
Multi-agent	High convergence speed, high fault tolerance, suitable for large-scale VANETs, high learning ability	Difficulty in creating coordination between agents, high computational complexity

5.2. Classification of RL-Based Routing Schemes with Regard to Learning Model

According to our classification, RL-based approaches are classified in two classes (i.e., model-based and free-model) with regard to learning model. Table 4 compares these classes.

- **Model-based RL routing scheme:** In this approach, the agent learns a model of the environment based on experience to forecast the value function. See Figure 5. An advantage of these schemes is data efficiency. This means that these methods can learn a more accurate estimation of value function with few interactions with the environment. Moreover, they are flexible against sudden changes in the environment and can be well adapted to these changes. For example, if a road section is blocked in VANET, model-based routing protocols will quickly adapt to this change in the network. However, these methods have high computational costs and are not appropriate for time-sensitive applications. This proves that model-based reinforcement learning methods are efficient when sufficient computational resources are available. However, these methods have a poor performance for large-scale applications such as VANETs where the state space is very big [37,38].

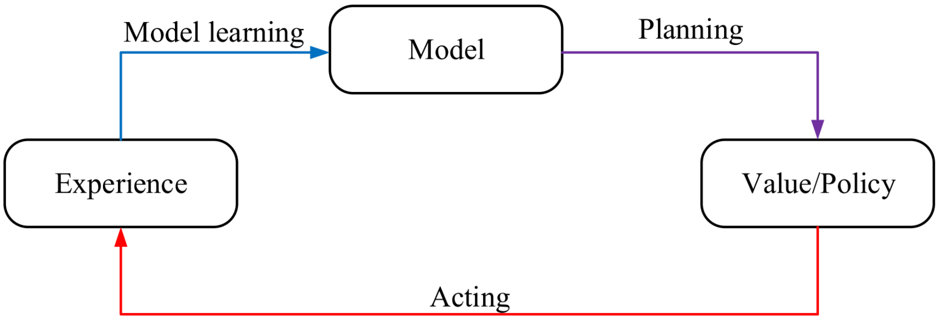

Figure 5. A model-based RL system.

- **Free-model RL-based routing approaches:** In this class, the agent directly estimates the value function in accordance with the obtained experiences so that these methods do not build any network model. Figure 6 represents a free-model reinforcement learning system. The main advantage of the free-model routing methods is that they are efficient in terms of computing. Therefore, they are more suitable for large-scale and time-sensitive applications in VANET. However, they have two important weaknesses: they need more experience to achieve an optimal response compared to model-based approaches and are less flexible against the sudden network changes [39].

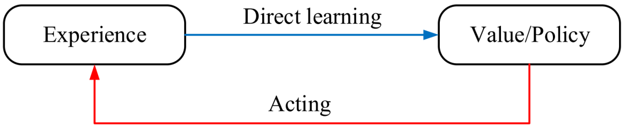

Figure 6. A free-model reinforcement learning system.

Table 4. Comparison of model-based and free-model routing methods.

Routing Scheme	Advantages	Disadvantages
Model-based RL routing method	Data efficiency, high flexibility against sudden changes in VANETs	High computational complexity, inappropriate for large scale networks, inappropriate for time-sensitive applications
Free-model RL-based routing method	Low computational complexity, suitable for large-scale networks and time-sensitive applications	Low flexibility against sudden changes in the network environment, need more experiences to achieve an optimal response

5.3. Classification of RL-Based Routing Methods with Regard to Learning Algorithm

According to the proposed classification, RL-based routing schemes are divided into traditional RL-based and DRL-based in accordance with the learning algorithm. Table 5 compares the strengths and weaknesses of RL and DRL-based routing protocols.

- **Traditional RL-based routing schemes:** These approaches allow the learning agent to learn the network environment without any information about it to get the most suitable path. Figure 7 displays the traditional reinforcement learning system. These routing methods have a good performance and can find the best route at an acceptable time when the state and action sets are small. However, VANETs are usually large-scale and the state space and the action space are large. Under these conditions, these route protocols have a low convergence speed and require more time to obtain the best response [40,41].

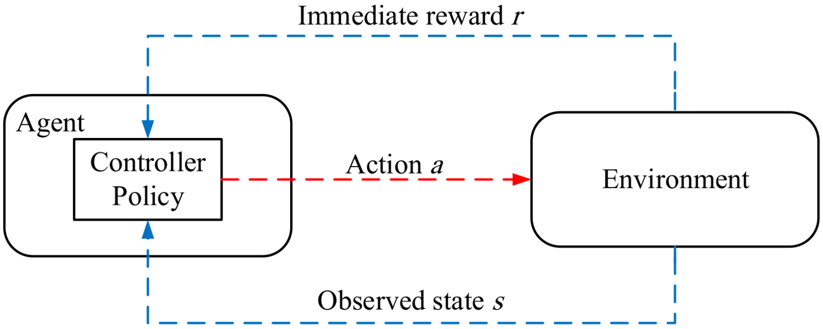

Figure 7. Traditional reinforcement learning system.

- **DRL-based routing methods:** They use deep learning (DL) to improve the learning rate. DRL integrates deep learning and reinforcement learning (RL). Figure 8 shows a DRL system. It is an improved version of RL that can solve complex computational processes. In complex environments such as VANETs, it is very complex to estimate the value function and the policy [38,42]. For this reason, a deep network is used to approximate these values. It accelerates the learning ability of the agent to optimize this policy. These routing protocols have a good learning rate and are an appropriate option for large-scale networks [40,43].

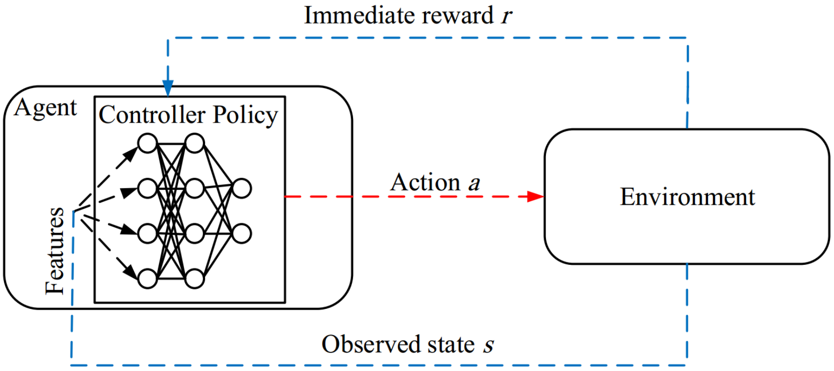

Figure 8. Deep reinforcement learning system.

Table 5. RL and DRL-based routing schemes.

Routing Scheme	Advantages	Disadvantages
RL-based routing method	Suitable for small VANETs with small state and action spaces	Inappropriate for large scale networks, low convergence speed, poor learning ability
DRL-based routing Method	Suitable for large-scale networks, Good convergence speed, High learning ability	The problem of instability in the model, sensitive to low changes in Q-value and consequently sudden changes in the policy

5.4. Classification of RL-Based Routing Approaches with Regard to Learning Process

According to this classification, RL-based routing methods are centralized or distributed in accordance with learning process. Table 6 presents the advantages and disadvantages of centralized and distributed routing protocols.

- **Centralized routing protocols:** In these protocols, RL algorithm is executed by a central agent, for example, a central server to obtain the optimal path. Then, the central agent loads this discovered route in the memory of the source node. This path is used for transferring data packets to the destination. The most important advantage of these routing schemes is to decrease communication overhead and computational costs because these methods do not need to exchange control packets between nodes. However, they have a major problem: These routing schemes may damage from a single point of failure. This means that if the central agent fails for any reason, the whole network will be disrupted. On the other hand, VANET is an extremely dynamic network and the central agent cannot adapt itself to topological changes in real-time. Furthermore, these methods have not scalability and are not suitable for large-scale VANET.
- **Distributed routing methods:** In these protocols, RL algorithm is locally executed on the network to learn the optimal path. In these methods, computational costs and communication overhead are greater than that in centralized routing approaches because vehicles must share local information about the network topology. However, these approaches are scalable and suitable for the dynamic network environment and real-time applications.

Table 6. Centralized and distributed routing methods.

Routing Scheme	Advantages	Disadvantages
Centralized routing method	Suitable for small VANETs, low computational costs and communication overhead	Inappropriate for large scale networks, single point of failure, not adapting to dynamic network topology
Distributed routing method	Suitable for large-scale networks, Adaptability to dynamic network environment	High communication overhead and computational costs

5.5. Classification of RL-Based Routing Methods with Regard to Routing Algorithm

According to the presented classification, RL-based routing approaches are categorized into three main groups based on routing algorithms: position-based, cluster-based, and topology-based. Table 7 expresses the strengths and weaknesses of different routing algorithms.

- **Position-based routing schemes:** These approaches use the geographical information of nodes. Thus, each node connects to a positioning system, like the global positioning system (GPS) to obtain its spatial information at any time [1,6]. These routing methods do not require all information of the network and utilizes local information. This improves communication costs, bandwidth, and consumed energy in these methods. Therefore, they are very suitable for highly dynamic networks such as VANET. These approaches are classified into two groups:

  - *Delay-tolerant networking (DTN) routing approaches:* These methods can manage the challenges caused by frequent disconnections in VANETs. This problem leads to breaking the paths created to the destination node. In most cases, these approaches use the store-carry-forward technique when the node cannot select a routing path to other nodes [44,45]. This technique greatly reduces communication overhead because it does not use any additional control packet. However, it increases delay when transferring data process [46,47].
  - *Non-delay tolerant networking (non-DTN) routing methods:* These protocols are utilized in networks with high connectivity so that the density of the nodes is relatively high. However, if the connectivity is not guaranteed in the network, the performance of these protocols will be weakened. Also, they use a greedy forwarding technique for the data transmission process [48]. According to this technique, the transmitters send data packets to the neighbor closest to the destination. However, if the sender does not find a neighbor close to the destination compared to itself, the data delivery process fails and a recovery strategy is used to manage this condition. These methods have a good performance in high-density networks. In addition, they have a low communication overhead, high scalability, and low memory requirement. The most important challenge in these approaches is to obtain accurate location information because, if the location of the nodes is not available and/or is not accurately calculated, these protocols have weak performance. In these methods, all nodes are equipped with GPS, which requires a lot of bandwidth.

- **Cluster-based routing protocols:** In these methods, vehicles have different responsibilities on the network [28,49]. Therefore, vehicles are categorized into different groups called clusters. In each group, a cluster head node (CH) manages the cluster and inter-cluster communication. These routing methods greatly lower the number of control messages and prevent network congestion [50]. Therefore, they are scalable. However, the challenges of this type of routing approaches, especially for dynamic networks such as VANET, are CH selection and cluster management.
- **Topology-based routing protocols:** In these approaches, topological information of nodes is used for transmitting data packets in the network [51,52]. They create a suitable path before starting the data transfer procedure. Topology-based routing methods are classified into three groups:

  - *Proactive routing methods:* These approaches are also known as table-driven protocols. In this technique, each vehicle transfers the newest routing information to other vehicles and does not consider whether they have data packets for sending? The routing information is kept in the routing tables of vehicles and is regularly refreshed and shared with network nodes. The proactive routing is not suitable for VANETs because they cannot well react against repeated topological changes and have high route breakage [53].
  - *Reactive routing methods:* These approaches are on-demand. In thses schemes, a vehicle begins the route discovery process only if it has a data packet, which must be delivered to the destination and there is no path for this work. In these protocols, vehicles maintain only the routing information about valid paths. As a result, a path maintenance system checks valid paths and eliminates invalid paths. When the network topology is updated, the failed paths will be eliminated and the route discovery process begins again. The reactive routing protocols are more efficient in terms of bandwidth consumption compared to proactive routing methods because routing tables are updated periodically [53].
  - *Hybrid routing protocols:* It combines proactive and reactive approaches and is suggested to overcome their weaknesses. The hybrid routing method reduces

communication overhead compared to proactive routing protocols and improves delay in the path discovery process compared with reactive routing schemes. They are especially suitable for large-scale networks [53].

Table 7. Comparison of different routing algorithms.

Routing Scheme		Advantages	Disadvantages
Position-based routing algorithm	DTN	Low routing overhead, high scalability, low bandwidth consumption, low memory consumption, high packet delivery rate	High delay in the routing process
	Non-DTN	Suitable for high-density and dynamic networks, low routing overhead, low bandwidth consumption, low memory consumption, high scalability	Needing accurate location computation of nodes, poor performance for networks with low density
Cluster-based routing algorithm		Low routing overhead, managing network congestion, high scalability	Difficulty in cluster management and CH selection
Topology-based routing algorithm	Proactive	Low delay in the data delivery process	Low scalability, inefficiency in resource consumption such as bandwidth, high memory consumption, high routing overhead for updating routing tables, unsuitable for VANET, not adaptability with repeated changes in topology due to frequent link breakages
	Reactive	Bandwidth efficiency, low routing overhead	High delay in the route discovery process, flooding control messages, network congestion, and high bandwidth consumption
	Hybrid	Routing overhead control, reducing delay in the routing process, suitable for large-scale networks	Difficulty in arranging and managing the network

6. Investigating Several RL-Based Routing Methods

Here, we study the latest RL-based routing approaches in VANETs.

6.1. IV2XQ

Luo et al. in [54] have introduced the intersection-based V2X routing using Q-learning (IV2XQ) for VANETs. IV2XQ is a geographic routing scheme, which utilizes a hierarchical routing framework and consists of two main parts: multi-dimensional Q-learning-based routing strategy for selecting the best road segments at intersections and the greedy routing strategy for selecting the best relay node in the selected road segment. In this scheme, the central server acts as the agent, and the learning environment is the network. The central server uses historical traffic data to explore the network environment to choose the most suitable path. In IV2XQ, the state space represents all network intersections. Thus, this scheme can well lower the number of states of Q-learning and makes it easier to implement. As a result, the algorithm has a high convergence speed. The action set consists of the road segments connected to one intersection, namely north road segment, south road segment, east road segment, and west road segment. The environment allocates the reward R to the agent after taking an action (i.e., selecting a road segment for forwarding a data packet from the previous intersection to the next intersection). If the data packet reaches the destination intersection, then $R = \varphi$ (where, $\varphi > 0$); otherwise, $R = 0$. In IV2XQ, the learning rate is empirically selected while, the discount factor is dynamically determined in accordance with the density of vehicles and distance to the desired node. Figure 9 shows the learning structure in IV2XQ. Additionally, in this method, the RSUs at intersections are equipped with monitoring modules to record the number of packets and calculate the network load locally. Therefore, if the network load is more than a load threshold, then the packet is sent to an alternative road segment. This congestion control mechanism prevents network congestion and increases the packet delivery rate. When discovering paths at each road segment, the enhanced greedy forwarding methodology is also applied to discover the most suitable relays in each road section. This algorithm is defined for both V2V and V2I communication. The greedy forwarding strategy selects the neighboring vehicle nearest to the desired target as a relay node. If the forwarding strategy fails and the algorithm is involved with the local optimum problem, the carry-and-forward technique will be used to recover the route.

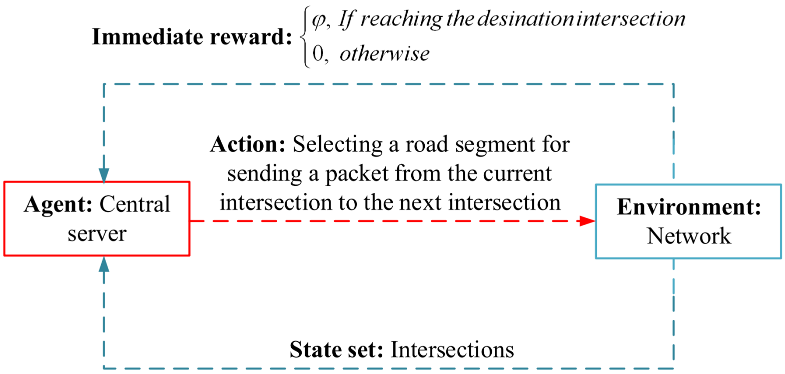
Immediate reward: $\begin{cases} \varphi, & \textit{If reaching the desination intersection} \\ 0, & \textit{otherwise} \end{cases}$
Action: Selecting a road segment for sending a packet from the current intersection to the next intersection
Agent: Central server
Environment: Network
State set: Intersections

Figure 9. Learning structure in IV2XQ.

6.2. QAGR

Jiang et al. in [55] have suggested the Q-learning-based adaptive geographic routing (QAGR) for VANETs. It uses an aerial network for helping the routing operation in VANET. This aerial network consists of several unmanned aerial vehicles (UAVs) to prevent the selection of wrong routes by vehicles due to their limited communication radius. UAVs calculate a global path for data transfer and send this path to the desired vehicles. As a result, vehicles can filter some of their neighbors when choosing the next-hop node. This increases the convergence speed and improves network performance. QAGR includes two main components: aerial component and ground component. In the aerial network, UAVs employ the depth-first-search algorithm (DFS) and fuzzy logic to obtain the global route. They use the fuzzy technique to calculate a fitness value for each road. This fuzzy system is dependent on the number distribution factor (NDF), the velocity distribution factor (VDF), and the total number proportion (TNP). In the ground component, when creating a new path, the source vehicle first transfers a route request to the corresponding UAV to calculate the global route to destination based on DFS algorithm and fitness value of road segments. Then, the UAV responds to this route request message. After receiving the response message, the vehicle inserts this route into the header of its data packets and employs Q-learning and the global path to choose the next-hop node. In Q-learning algorithm, each vehicle plays the role of an agent and tries to choose the most appropriate node among its neighbors. In this problem, the state space includes a two-dimensional array including the distance between each vehicle and its neighbors and the neighboring degree. In this algorithm, the reward function is obtained from received signal strength indication (RSSI), transmission distance, and collision between vehicles. The learning structure of QAGR is shown in Figure 10. After the algorithm has converged and finished, the node with the highest Q-value obtains a more chance to be selected as the next-hop node. Moreover, if multiple nodes have the same Q-value, QAGR selects a next-hop node among them. In QGAR, if the vehicle cannot find the next-hop node, it sends its packet to the corresponding UAV to perform the data transmission process via the aerial network.

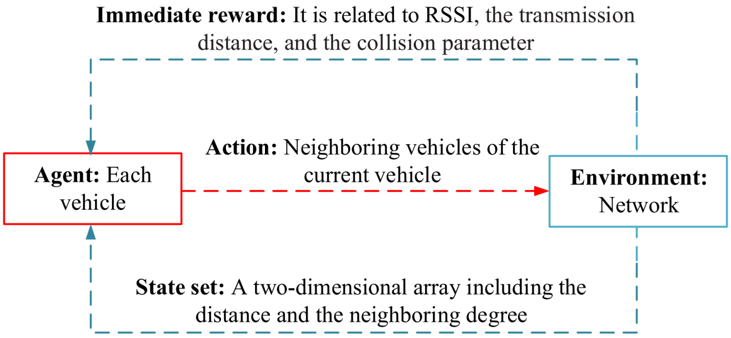
Immediate reward: It is related to RSSI, the transmission distance, and the collision parameter
Action: Neighboring vehicles of the current vehicle
Agent: Each vehicle
Environment: Network
State set: A two-dimensional array including the distance and the neighboring degree

Figure 10. Learning structure in QAGR.

6.3. CEPF

An et al. in [56] have presented a context-aware edge-based packet forwarding approach (CEPF) for VANETs. It supports both unicast and broadcast communication types. This method includes two main steps: selecting edge nodes and learning the best route for the last two-hop communication using reinforcement learning. Vehicles selected as the edge nodes are responsible for transmitting, processing, and storing data. The advantage of edge nodes in CEPF is that the number of forwarder nodes is reduced in the routing process. This enhances resource efficiency and improves the performance of this routing scheme. Edge nodes are selected in a decentralized fuzzy logic-based manner. This fuzzy system has the three input parameters, including vehicle speed, the number of nodes moving in the same direction, and link quality to estimate the fitness value of each vehicle for choosing the edge vehicle. After receiving each hello message, each vehicle computes its fitness value using the fuzzy mechanism and compares it with the fitness values of other neighboring vehicles. If the node has more fitness than the neighboring vehicles, the vehicle will be the edge node. In the reinforcement learning-based route discovery operation, the learning environment is the whole network, and each packet is the agent. The state space includes all vehicles in the network and the action set contains the choice of a single-hop neighbor as the next-hop node. Q-table consists of Q-values that are shared through hello messages with other neighbors. Note that in the Q-value updating process, the link status parameter is considered to create stable routes. It is computed with regard to the hello reception rate. The learning structure of CEPF is represented in Figure 11.

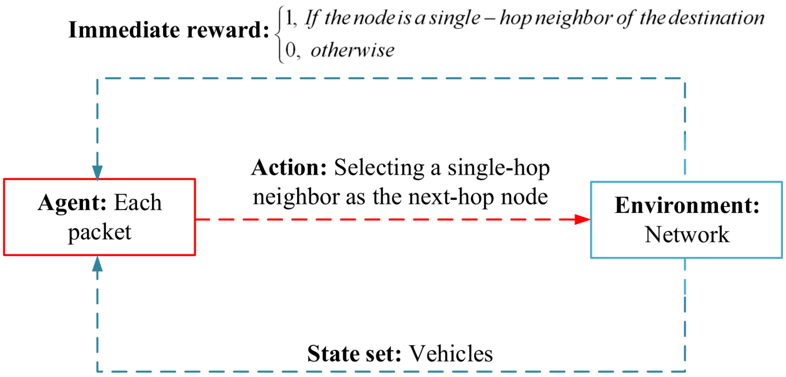
Immediate reward: $\begin{cases} 1, & \textit{If the node is a single-hop neighbor of the destination} \\ 0, & \textit{otherwise} \end{cases}$
Action: Selecting a single-hop neighbor as the next-hop node
Agent: Each packet
Environment: Network
State set: Vehicles

Figure 11. Learning structure in CEPF.

6.4. PFQ-AODV

Wo et al. in [57] have offered a fuzzy Q-learning technique called PFQ-AODV for VANETs. The purpose of this method is to obtain the most suitable path between network nodes. In general, PFQ-AODV utilizes a fuzzy system to calculate connection links and utilizes a Q-learning algorithm to learn the most suitable paths. In PFQ-AODV, each node regularly transfers a hello message including bandwidth information with its neighboring nodes. After receiving this message, vehicles store the information about their single-hop and two-hop neighboring nodes in a neighboring table. In PFQ-AODV, nodes are independent of a positioning system. This has improved bandwidth consumption in this method. PFQ-AODV has designed a fuzzy logic-based link evaluation technique to analyze the link status between a vehicle and its neighbors. This link evaluation is applied to refresh the Q-value in PFQ-AODV. In this routing algorithm, each packet acts as the agent, and the network illustrates the learning environment. The state set involves all network nodes, and the action set consists of all single-hop neighbors that can be selected by the agent in the current state. Figure 12 displays the learning structure in PFQ-AODV. When creating a path, the source vehicle generates a route request (RREQ) packet and broadcasts it to its neighbors. Each node that receives RREQ for the first time re-disseminates this message. The operation continues until RREQ reaches the desired vehicle. In PFQ-AODV, after receiving any hello or RREQ, each node refreshes Q-table. The Q-value is used to evaluate nodes for participating in the data transmission process. After receiving RREQ, the desired vehicle searches its Q-table and choose a vehicle with the highest Q-value, and transfers the route reply (RREP) message to this node. The operation continues until RREP reaches the source vehicle. Ultimately, the created route can be applied for transmitting data packets to the destination node.

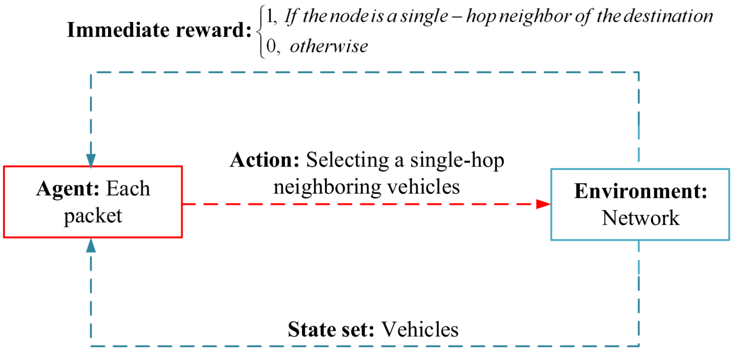
Immediate reward:
$\begin{cases} 1, & \textit{If the node is a single-hop neighbor of the destination} \\ 0, & \textit{otherwise} \end{cases}$
Action: Selecting a single-hop neighboring vehicles
Agent: Each packet
Environment: Network
State set: Vehicles

Figure 12. Learning structure in PFQ-AODV.

6.5. QGrid

Li et al. in [58] have suggested the Q-learning and grid-based (QGrid) routing method for VANETs. This method works on both levels, macroscopic and microscopic. At the macroscopic level, the network is partitioned into smaller grids. Then, QGrid explores the most suitable next grid towards the destination direction. Unlike other grid-based methods, this approach does not select a fixed grid head node for each grid. It chooses the relay vehicle in each grid using the strategy provided at the microscopic level. When selecting the optimal next grid, QGrid uses a Q-learning algorithm. In this process, a virtual agent is considered in the network. Also, the state space contains all grids in the network. As a result, QGrid decreases the state space and accelerates the convergence rate. Also, the set of actions indicates the transmission of a message to the neighboring grid. The learning structure of QGrid is displyed in Figure 13. In this learning model, the historical traffic data is obtained from GPS and Q-table is created based on this data. Q-value illustrates the possibilities of a vehicle to enter the next grids. In this scheme, the agent creates Q-table using an off-line manner. Thus, this table is fixed throughout the simulation. This table is loaded into the memory of each vehicle before deploying the network. At the microscopic level, the transmitter node should select a relay vehicle at the optimum grid. The relay selection process is based on two different strategies: (1) In the greedy forwarding methodology, the neighboring vehicle closest to the destination is selected as the relay node. (2) In the Markov prediction method, the relay vehicle is selected by the two-order Markov chain with regard to the possibility of moving toward the optimal grid.

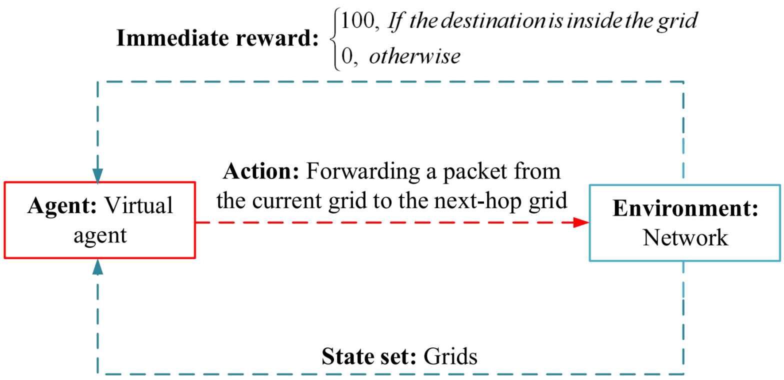
Immediate reward: $\begin{cases} 100, & \textit{If the destination is inside the grid} \\ 0, & \textit{otherwise} \end{cases}$
Action: Forwarding a packet from the current grid to the next-hop grid
Agent: Virtual agent
Environment: Network
State set: Grids

Figure 13. Learning structure in QGrid.

6.6. RRPV

Jafarzadeh et al. in [59] have proposed a reinforcement routing protocol (RRPV) in VANETs. This method integrates model-based reinforcement learning and fuzzy logic. RRPV utilizes a multi-agent model-based reinforcement learning technique called DynaQ, which has a high convergence speed. Model learning and reinforcement learning are two main components in DynaQ architecture. This routing approach applies the fuzzy logic-based system to build the learning model, and the Markov decision framework is also used for the second component. This fuzzy logic-based system has two input parameters, namely link stability and connection quality. Link stability indicates the connection time, which is determined based on the Euclidean distance, relative velocity, and movement direction. Furthermore, the connection quality is also evaluated based on the packet reception ratio. After determining the link quality using the fuzzy system, its results are considered as the state transition probability from the former state to the latter state in MDP. In the reinforcement learning process, any vehicle that has a data packet for sending is considered as an agent. In this case, each vehicle (agent) has two states, including *F* (i.e., the node has a packet for sending) and *D* (i.e., the node has delivered its packet to a neighbor). The action set represents the transmission of a hello message from the current vehicle to the neighboring vehicle. Moreover, the reward function is determined with regard to the link quality and the Euclidean distance between the two neighboring vehicles. The learning structure of RRPV is represented in Figure 14. MDP generates Q-values, which are used to choose relay nodes. Each vehicle stores a routing table having its Q-values and the V-values of neighboring vehicles. These values are periodically updated, meaning that the routing paths are gradually changed.

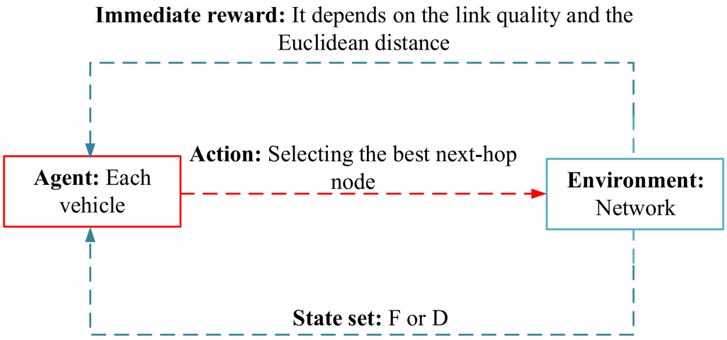
Immediate reward: It depends on the link quality and the Euclidean distance
Action: Selecting the best next-hop node
Agent: Each vehicle
Environment: Network
State set: F or D

Figure 14. Learning structure in RRPV.

6.7. QTAR

Wu et al. in [60] have suggested the RSU-assisted Q-learning-based traffic-aware routing method (QTAR) for VANETs. This routing scheme is a combination of the geographic routing and Q-learning. QTAR uses two Q-learning-based routing algorithms for sending data packets between vehicles at the road level and for sending data packets between RSUs at the intersection level. These two algorithms are called V2V Q-learning-based greedy geographical forwarding and R2R Q-learning-based greedy geographical forwarding. In the routing process, vehicles broadcast $Hello_{V2V}$ messages, which include their speed and position. In addition, RSUs exchange $Hello_{R2R}$ messages with each other. In V2V Q-learning-based routing method at road segments, each packet illustrate the learning agent, and the network expresses the learning environment. The state space contains neighboring vehicles. In this learning process, the set of actions consists of all neighboring vehicles, which can be chosen as the latter vehicle by the former vehicle. In this algorithm, the reward function is obtained from link quality, link expiration time, and delay. This learning structure is shown in Figure 15a. As a result, each vehicle maintains a V2V Q-table to send data packets at road segments. This Q-table is refreshed after receiving any $Hello_{V2V}$. In the R2R Q-learning-base routing algorithm, each packet illustrates the learning agent, and the set of states demonstrates neighboring RSUs. Furthermore, the action space involves all neighboring intersections. Each RSU maintains two Q-tables: V2V Q-table for sending data packets at road segments and R2R Q-table for sending data packets at intersections. Furthermore, Figure 15b displays the R2R Q-learning structure. Note that these Q-tables will be updated after receiving $Hello_{R2R}$ messages. Finally, if the QTAR method is involved with the local optimal problem, the store-carry-forward technique will be used to decrease the packet loss rate.

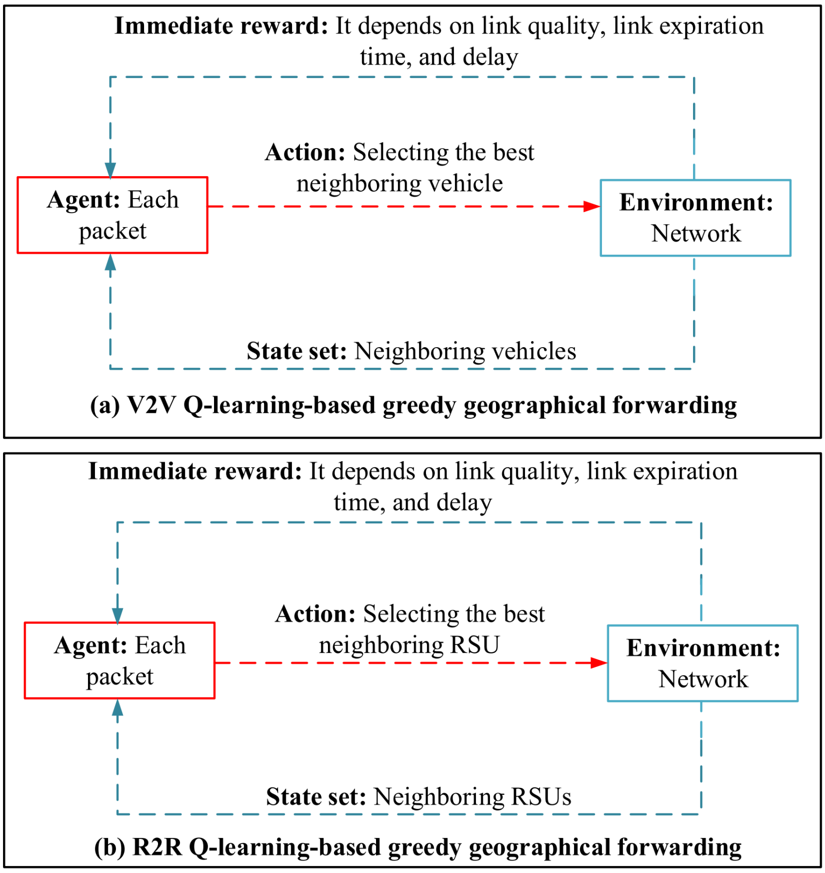
Immediate reward: It depends on link quality, link expiration time, and delay
Action: Selecting the best neighboring vehicle
Agent: Each packet
Environment: Network
State set: Neighboring vehicles
(a) V2V Q-learning-based greedy geographical forwarding
Immediate reward: It depends on link quality, link expiration time, and delay
Action: Selecting the best neighboring RSU
Agent: Each packet
Environment: Network
State set: Neighboring RSUs
(b) R2R Q-learning-based greedy geographical forwarding

Figure 15. Learning structure in QTAR.

6.8. Q-LBR

Roh et al. in [61] have introduced the Q-learning-based load balancing routing (Q-LBR) scheme for VANETs. This method uses unmanned aerial vehicles (UAVs) in the routing operation. In this method, it is assumed that the UAVs can monitor the congestion level on the ground network. It has also designed a Q-learning-based congestion control system to balance the load on the network. In this method, UAVs estimate the network load based on the queue status of each vehicle in the ground network. This technique has a low routing overhead. Q-LBR consists of four main steps: route discovery and maintenance, network load estimation, Q-learning-based load balancing, and routing decision. In the route discovery and maintenance phase, Q-LBR, like the source-based multi-path routing protocol, disseminates RREQs to discover paths. Finally, the destination node responds to this request by sending back the route reply (RREP) message for optimal and suboptimal paths. In this process, UAVs also receive RREQ messages from ground vehicles. Therefore, the created paths may also include UAVs as relay nodes. The network load estimation includes two steps: identifying the ground network congestion level and identifying the UAV congestion level. Each vehicle estimates the ground network congestion identifier (GNCI) parameter based on its queue load and sends it to the corresponding UAV through a hello message. As a result, the UAV can estimate the average congestion level of the ground network. Next, each UAV calculates the UAV relay congestion identification (URCI) parameter based on its buffer queue. GNCI and URCI are considered as the state space in the Q-learning-based load balancing methodology, and UAV acts as an learning agent. According to the current state, UAV must select its action from the action space, which includes the upper UAV routing policy area ($URPA_{upper}$) and the lower UAV routing policy area ($URPA_{lower}$). The reward value is obtained from GNCI and URCI. This learning structure is displayed in Figure 16. Finally, the URPA parameter is sent to the ground nodes. In the routing decision phase, when the source vehicle obtains several RREP messages, it must choose the best route for the data transmission process from those created paths. These routes include ground routes and the routes having UAV nodes. If a route includes UAVs as relay nodes and has less routing cost than the ground routes, and the congestion level of the UAV nodes is less than a congestion threshold, the source node selects this path for sending data packets. Otherwise, the ground route will be selected to send data packets.

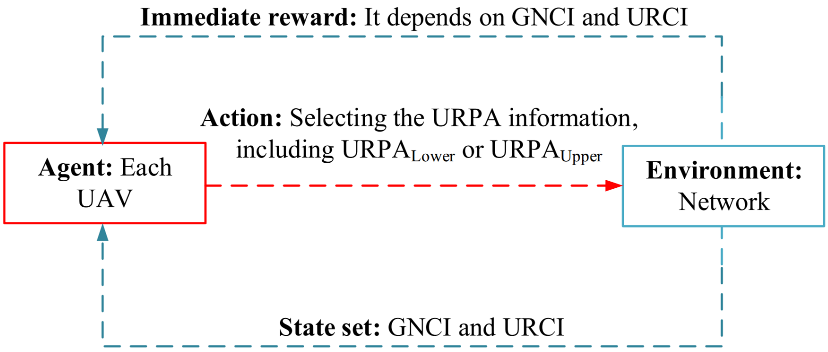
Immediate reward: It depends on GNCI and URCI
Action: Selecting the URPA information, including $\mathrm{URPA_{Lower}}$ or $\mathrm{URPA_{Upper}}$
Agent: Each UAV
Environment: Network
State set: GNCI and URCI

Figure 16. Learning structure in Q-LBR.

6.9. ECTS

Li et al. in [62] have suggested the efficient charging information transmission strategy (ECTS) in hybrid vehicular networks. Large-scale hybrid vehicular networks include a large number of electric vehicles, which must exchange a large volume of charging information. ECTS is a scalable and efficient routing framework that can greatly reduce communication overhead. In ECTS, an accurate mathematical model is designed to estimate the probability of local connectivity in the two-line road segments. This model improves the performance of the routing process in ECTS. Then, an efficient routing algorithm is presented to transfer the charging information between the server and the electric vehicle in the VANET environment. In ECTS, each electric vehicle can achieve its location and the server position using a digital map and GPS. In addition, an immobile node such as RSU is located at each intersection. It is responsible for storing the routing information and receiving the density and location of vehicles at each road segment. This routing approach tries to choose the most suitable intersection-based path with maximum connectivity. This issue is solved using a Q-learning algorithm. When transferring data between the source vehicle and the server, the first step is to determine source and destination intersections. These intersections are determined with regard to movement direction and curvilinear distance between the desired node and the intersection. Now, the source vehicle transfers its charging data to the source intersection. Then, the source intersection runs a Q-learning-based route discovery operation. In this operation, the state space consists of all intersections in the network, and the set of actions includes the selection of the latter intersection. Figure 17 shows the learning structure in ECTS. When selecting the road segment with high connectivity, the discount parameter is dynamically adjusted to deliver charging data efficiently and maintain a tradeoff between the future and immediate rewards. In ECTS, the size of the state and action sets is limited. Therefore, this algorithm has an acceptable convergence speed.

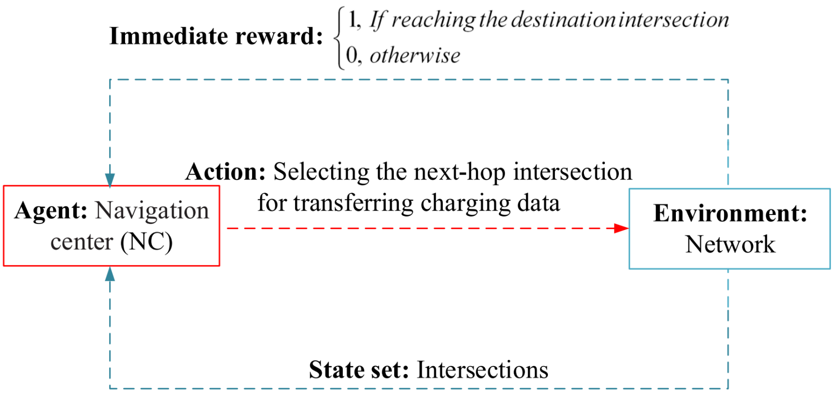
Immediate reward: $\begin{cases} 1, & \textit{If reaching the destination intersection} \\ 0, & \textit{otherwise} \end{cases}$
Action: Selecting the next-hop intersection for transferring charging data
Agent: Navigation center (NC)
Environment: Network
State set: Intersections

Figure 17. Learning structure in ECTS.

6.10. RLRC

Bi et al. in [63] have presented a RL-based routing protocol for clustered network (RLRC), which consists of two main steps: K-Harmonic Means (KHM) clustering algorithm and RL-based routing algorithm. In RLRC, each vehicle employs a digital map and a positioning system to find out its speed, position and direction at any moment. Each electric vehicle shares this information through hello messages with adjacent vehicles. This information is applied in the clustering operation. The KHM-based clustering algorithm categorizes electric vehicles into k clusters. In the clustering process, the cluster head nodes (CHs) are determined with regard to bandwidth and residual energy. In the data transmission process, each cluster member node transfers its data directly to the corresponding CH. Then, the CH forwards the data to its neighboring CH through a RL-based routing algorithm. In the routing operation, the Sarsa-Lambda learning technique has been used to choose the most suitable path between CHs. In this issue, the whole network expresses the learning environment, and each node is the agent. The state set includes neighboring CHs, and the action set represents the choice of a next-hop CH. Also, the reward value is computed with regard to the link status parameter, which is obtained based on the bandwidth factor and the inverse link duration (ILD). Figure 18 shows the learning structure in RLRC. In this approach, the Q-value is renewed by hello messages and the route request (RREQ) messages. CHs periodically broadcast hello messages to their neighboring nodes. This message contains the highest Q-value obtained for sending data through this neighboring node to the destination.

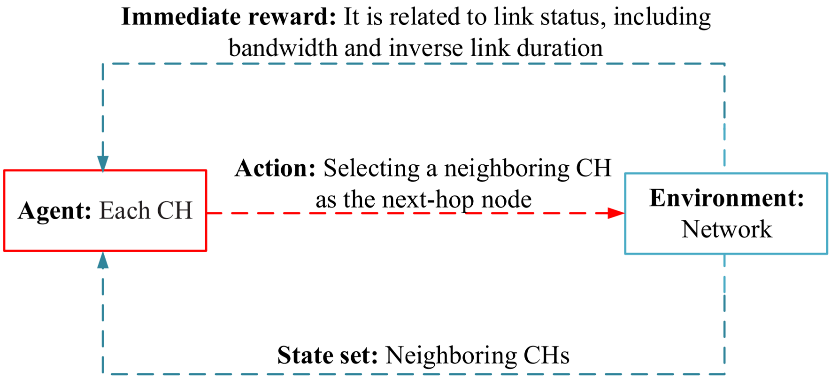
Immediate reward: It is related to link status, including bandwidth and inverse link duration
Action: Selecting a neighboring CH as the next-hop node
Agent: Each CH
Environment: Network
State set: Neighboring CHs

Figure 18. Learning structure in RLRC.

6.11. GLS

Zhao et al. [64] have presented the greedy routing with link stability (GLS) to the software-defined vehicular networks. This method divides the vehicular network into several sub-areas so that each intersection is located at the center of each sub-area. The controller performs the area selection process (AS) using a proactive manner. In GLS, the central controller checks traffic conditions in each area and initializes the entries of the routing table using a fuzzy system. This fuzzy system consists of three input parameters, namely mixed distribution (MD), one-way connectivity (OC), and valid distance (VD). Note that two scales, namely the density of vehicles in a road segment and the vehicle distribution in that area are used for calculating the MD parameter. The OC parameter is also determined by the density and movement direction of vehicles and indicates the successful packet delivery rate from an area to the adjacent area. Finally, the VD parameter guarantees that packages are sent to areas toward the destination and are close to it. According to this fuzzy system, a transfer priority is determined for each road segment and is recorded into the routing table. Then, a reinforcement learning scheme is used to select an area. This helps GLS to better adapt to the dynamics of the network environment. In this learning model, the controller expresses the agent, and each area in the network indicates a state. Moreover, the set of actions is to select adjacent areas for transferring data packets. Figure 19 shows the learning structure in GLS. This routing scheme presents a new technique for updating Q-table (also known as the routing table). In the multi-hop data transmission process, each packet stores all areas passed between the source node and the desired area. After ending this process, a report message about this route is uploaded into the controller. Then, it uses this information for updating Q-table. Upon receiving this message, the controller uses a tree-based positive updating mechanism to update successful paths in the routing table and improve their corresponding Q-values by giving a positive reward. This positive reward is calculated based on MD, OC, and VD parameters. On the other hand, if the routes are broken when sending the data, the controller uses a loss-aware negative updating mechanism to penalize the paths that have increased packet loss by giving a negative reward. This negative reward is calculated based on MD, OC and VD, and the size of the buffer capacity. After determining the connections between the adjacent areas using fuzzy logic and reinforcement learning, the central controller is responsible for deciding on the route based on the routing table. When the source node needs a new path between itself and the destination node, it transmits a route request message to the controller. After receiving this request, the AS operation is started by searching the routing table to determine the suitable route for transferring data (a sequence of intersections with maximum Q-value). After determining the optimal path, the controller sends this path to the requesting node through a message. The source node records this path in the packet header and sends it to the intermediate node, which is selected using the relay vehicle selection (RS) process. The intermediate node extracts the route information from the packet header and determines the next area for sending the data. In the relay vehicle selection

(RS) process, each vehicle uses a greedy forwarding technique to choose a relay vehicle with the maximum link stability from its neighboring vehicles close to the destination. For this purpose, the transmitter vehicle uses a fuzzy system to calculate the link stability for each neighboring vehicle. This fuzzy mechanism involves three inputs, including distances, movement direction, and speed. This information is obtained by broadcasting hello message in the network.

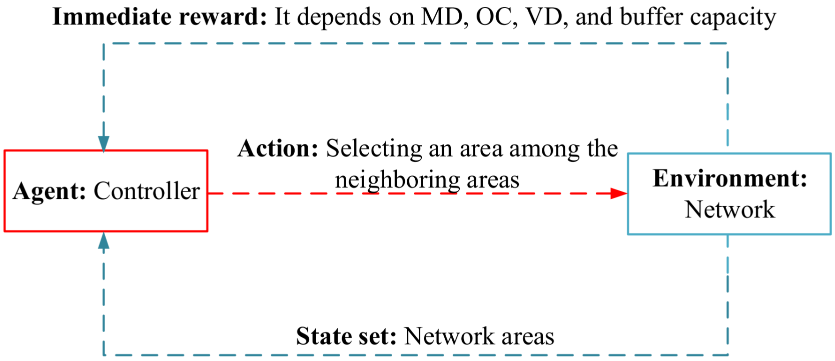
Immediate reward: It depends on MD, OC, VD, and buffer capacity
Agent: Controller
Action: Selecting an area among the neighboring areas
Environment: Network
State set: Network areas

Figure 19. Learning structure in GLS.

6.12. RSAR

Zhang et al. in [65] have offered the reliable self-adaptive routing scheme (RSAR) for VANETs. In this method, it is assumed that vehicles are distributed at the one-way highway according to a log-normal distribution. The first step in RSAR is to present a connection lifetime model with regard to the features of the vehicle movement such as speed, acceleration, direction, and distance. This model is applied for computing the link reliability, which is used in the learning model. In the Q-learning-based routing scheme, each vehicle expresses a learning agent. The state set consists of all vehicles except the vehicle considered as the agent. The action space also indicates a beacon message that is sent from the current vehicle to the next vehicle. Figure 20 shows the learning structure in RASR. In this optimization issue, each vehicle stores a Q-table, and the size of this table depends on the number of single-hop neighboring nodes and the number of destination nodes. Q-value is periodically freshened by broadcasting beacon messages. Each beacon message includes information about the vehicle, including its speed, location, and Q-value. In this method, the learning process is performed using a decentralized technique, which increases the convergence rate. Furthermore, the scalability of this method is also improved. In this approach, the learning parameters are computed with regard to the number of hops, bandwidth, and link reliability. RSAR has two parts: path development and route maintenance. In the path development operation, the source vehicle searches its Q-table to obtain the most suitable relay vehicle with the highest Q-value. If it finds such a vehicle, it forwards this data packet to this relay vehicle. Otherwise, the source vehicle begins the path discovery operation by broadcasting a route request on the network. RREQ stores the list of all the intermediate nodes passed to reach the destination node. Next, the destination vehicle produces a RREP message and transfers it to the source vehicle. When a vehicle receives RREP, it refreshes its Q-table. After forming the path and updating the Q-value in neighboring tables, the route maintenance process is started by exchanging beacon messages periodically to ensure the validity of the created routes. This process is responsible for dynamically maintaining the Q-table and solving the network segmentation problem. To update Q-table effectively, the beacon message must be exchanged at a certain time interval. If the time duration to reach the destination through a certain path is longer than this beacon time interval, the path is not updated and is known as an invalid path and is deleted from Q-table. When the network is segmented because vehicles are moving in the network, RSAR utilizes the store-and-forward strategy and re-start the route discovery process to build a new path.

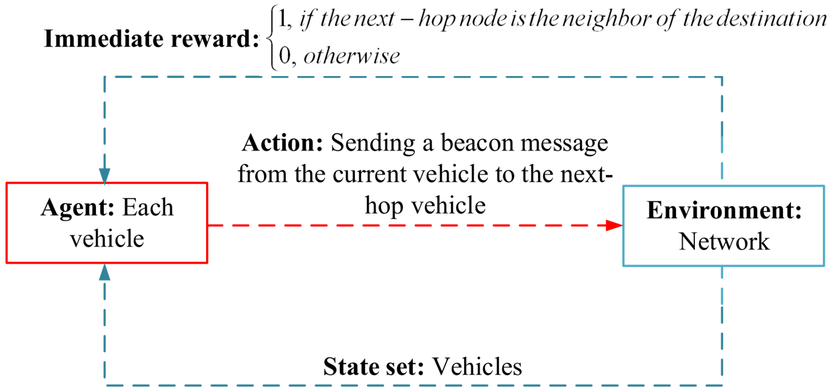
Immediate reward: $\begin{cases} 1, \; if\; the\; next-hop\; node\; is\; the\; neighbor\; of\; the\; destination \\ 0, \; otherwise \end{cases}$
Action: Sending a beacon message from the current vehicle to the next-hop vehicle
Agent: Each vehicle
Environment: Network
State set: Vehicles

Figure 20. Learning structure in RSAR.

6.13. Wu et al. Method

Wu et al. in [66] have suggested a cluster-based routing protocol for VANETs. In this approach, it is assumed that each vehicle exchanges its information such as location and speed through hello messages with neighboring vehicles. In the first step, a decentralized clustering approach based on fuzzy logic is designed to choose CHs. In this fuzzy system, there are three inputs, including mobility factor (MF), leadership factor (LF), and signal quality factor (SQF). The mobility factor is obtained from the velocity of vehicles in the network. The leadership factor is evaluated with regard to the density of nodes moving in the same direction of the current vehicle and the signal quality factor is also computed based on the hello reception rate. These parameters are extracted from the hello message. Upon receiving hello message, each node gets the fitness value of itself and its own single-hop neighbors using the fuzzy system. Then, a node with the highest fitness is chosen as a CH. It announces itself as the CH node through the hello message. In this clustering process, the communication overhead for joining/leaving clusters is zero because this scheme does not require the joining/leaving messages for maintaining the information of cluster member nodes. Moreover, cluster head nodes can be directly connected with other neighboring CH nodes. In the next step, a coalitional game theory-based model implements the clustering process to select a better path. In this model, players indicates single-hop neighbors, and each coalition value expresses an agreement between vehicles to use the cluster-based transmission process. The coalition value is determined by the average collision probability. Initially, each RSU initializes payoff and sends it to CHs. Ultimately, the payoff is evenly allocated by each CH to its single-hop neighbors. In this process, if a vehicle is close to RSU, it receives the payoff directly from the RSU. Otherwise, it must choose a CH vehicle that is closest to the RSU. Note that vehicles tend to send their data through CHs because the payoff is only distributed by CHs and RSU. This reduces the number of transmitter nodes and improves the network throughput. This payoff evaluates routing paths in the RL-based routing operation. In the RL-based route selection process, the network nodes are considered as the learning agent that learns the network environment by exchanging the hello message. The action is the choice of the latter vehicle. This learning structure is depicted in Figure 21. Each vehicle stores a Q-table that is refreshed by hello messages. Note that the Q-value is related to the number of hops, the payoff value obtained from the game theory, and the signal quality factor (SQF). Moreover, fixed values are considered for the learning rate and the discount factor.

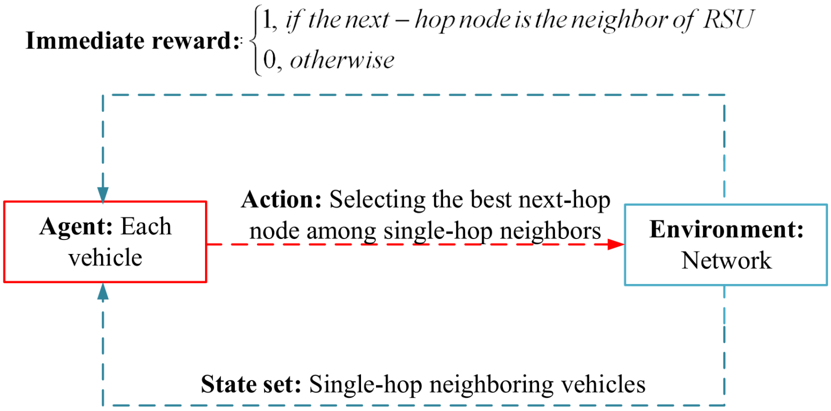
Immediate reward:
1, if the next – hop node is the neighbor of RSU
0, otherwise
Agent: Each vehicle
Action: Selecting the best next-hop node among single-hop neighbors
Environment: Network
State set: Single-hop neighboring vehicles

Figure 21. Learning structure in Wu et al. scheme.

6.14. RHR

Ji et al. in [67] have proposed the reinforcement learning-based hybrid routing algorithm (RHR) for VANETs. It is a multipath routing method, which combines proactive and reactive routing techniques. Multipath routing increases fault tolerance in RHR because if a route fails for any reason, the next path will be replaced. Moreover, RHR can solve the blind path problem. This problem means that a route will be invalid before ending its route expiration time. Lind path increases packet loss rate in the network. RHR can solve this problem because it utilizes multiple paths between two nodes. This method uses a reinforcement learning algorithm to make a decision on the path. Additionally, there are *k* nodes as the next-hop node (if exist) for each route to a particular destination in the routing table. This means that RHR can simultaneously discover multiple routes between source and destination. It maintains information such as the route weight and the route lifetime for each entry in the routing table. The route weight indicates the priority of a path for transferring data and is determined by Q-learning algorithm. In RHR, the Q-learning technique implements the routing operation. In this scheme, the next-hop node selection modes are considered as a set of states. Furthermore, the action set corresponds to the reception of different packets related to the next-hop vehicle. Figure 22 illustrates the learning structure of RHR. Taking different actions has a different impact on the routing table and changes the weight of each path according to this selected action. This weight is used as feedback to determine how much the path is suitable for transferring data. In RHR, selecting the next-hop node is limited to *k* nodes to manage the size of Q-table. In this table, Q-value is refreshed according to the information of the exchanged packets to gradually converge to the best path, so that paths with higher weight will be recorded in the routing table and weak paths will be eliminated. RHR designs the broadcast control technique to prevent routing loops. For example, it adjusts the time interval for broadcasting beacon messages dynamically. According to this adaptive broadcast technique, this time interval is determined by a vehicle position prediction strategy. This strategy uses the speed information of vehicles and reduces bandwidth consumption when broadcasting beacon messages. On the other hand, a vehicle broadcasts its data packets when it has no path to the desired node. To control the broadcast process, it determines a time to live (TTL) for this packet to manage the flooding problem. If the desired vehicle is the neighbor of the transmitter node, the packet is sent in a unicast manner. Also, if the number of neighbors is very high, the vehicle will stop the broadcast process to reduce the flooding problem.

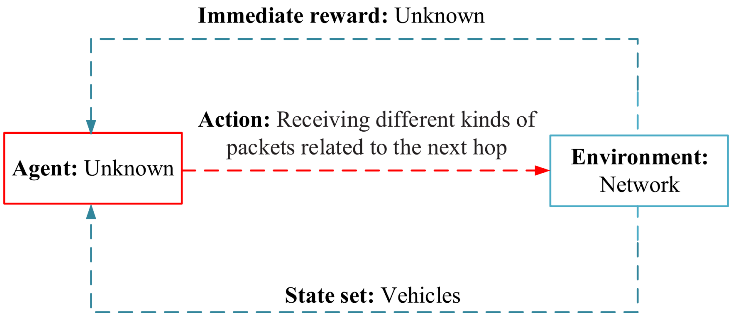
Immediate reward: Unknown
Action: Receiving different kinds of packets related to the next hop
Agent: Unknown
Environment: Network
State set: Vehicles

Figure 22. Learning structure learning in RHR.

6.15. SeScR

Nahar and Das in [68] have presented the SDN-enabled spectral clustering-based routing scheme using deep learning (SeScR) in VANETs. It utilizes SDN for central management. There are three main components, namely controller, RSU, and vehicle in this scheme. The controller is responsible for making an intelligence system and improving routing decisions. RSUs are fixed network units managed by the controller. Vehicles are connected to RSUs and send topological information to them by exchanging the beacon message. Then, RSUs forward this information to SDN to be used in the routing process. In the clustering process, vehicles are divided into several groups to minimize routing overhead by controlling the broadcast packet. SeScR uses the spectral clustering technique that creates stable clusters using eigenvalues of the Laplacian matrix. For choosing CHs, each vehicle computes a parameter called cluster head eligibility score (CES) based on relative velocity and Euclidean distance. Then, vehicles share their CES with each other using a beacon message. After receiving these messages, a vehicle with a maximum CES will be CH. This node is responsible for coordinating and calculating the best route to the destination. To calculate the best route, SeScR utilizes the deep deterministic policy gradient (DDPG) technique, which has a successful performance in large-scale networks with large state space. DDPG is used to determine whether a vehicle is suitable to be selected as the relay node in the data transmission process. This technique works in accordance with the actor-critic architecture. The actor network is responsible for suggesting an action for the current state and the critic network evaluates the action and forecasts its positive or negative effect. The learning process begins by exchanging beacon messages on the network. These messages include speed, location and other information. For optimizing the routing process using DDPG, vehicles play an agent role, and the state set involves the location, speed, and direction of the vehicles. In addition, an action decides for sending packets. Figure 23 shows the learning structure in SeScR. According to this method, each agent performs an action at the current state to obtain a new state. Based on the road, packets can be sent to the front, rear, left, and right neighboring vehicles. Additionally, this packet may be eliminated because of failure to access the destination node. The learning process is repeated until the packet reaches the desired vehicle.

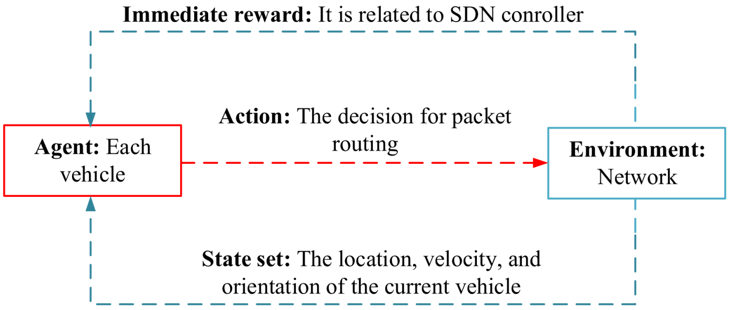
Immediate reward: It is related to SDN conroller
Action: The decision for packet routing
Agent: Each vehicle
Environment: Network
State set: The location, velocity, and orientation of the current vehicle

Figure 23. Learning structure in SeScR.

6.16. IRQ

Khan et al. in [69] have suggested an intersection-based routing technique using Q-learning (IRQ) for VANET. In this approach, a novel structure is proposed to broadcast traffic information in the network. According to this structure, the central server and other network nodes (i.e., vehicles and RSUs) can access the updated traffic information. They search for routes beads on both refreshed and historical traffic information and do not rely only on the old information. To create such a structure, two beacon and traffic messages are used. Beacon messages are regularly exchanged among all network nodes while traffic messages transfer traffic condition from RSUs to the central server. According to this traffic structure, IRQ obtains two global and local perspectives in the network. The global view shows the general traffic status in the road sections and is used by the central server to build a routing solution based on Q learning. This solution builds the most suitable routes at different network intersections. In this learning operation, the central server acts as an agent and uses its traffic table that contains the traffic status of different roads, to search the network environment for obtaining optimal paths between intersections. Also, the network acts as a learning environment, cooperates with the central server (i.e., the agent), responds to the actions performed by the agent and calculates the latter state and the reward corresponding to the performed action. The state set consists of all intersections, and the action set contains the road sections connected to the current intersection. In this method, the reward value is defined with regard to the density of vehicles, connection time, and delay. See Figure 24. In the learning model, the discount factor is a dynamic value obtained from the density of the road sections and the distance to the destination, and the learning rate has a fixed value based on experience. In this operation, the central server prevents congestion in the routes in accordance with a suitable solution to reduce collision and packet loss. This solution applies information about the traffic situation, i.e., delay and density of vehicles. Whenever the network server finds a congested path based on the congestion detection solution, it reduces the reward value of this path and prevents the selection of the path in future. Ultimately, IRQ employs a greedy routing approach for selecting paths in road sections. This approach is dependent on the local view. This forwarding solution is used for two parts such as vehicle-to-vehicle (V2V) routing and vehicle-to-RSU (V2I) routing. In V2V routing, each node picks out the closest vehicle to the target and forwards the data packet to the node. If the local optimum problem occurs in the routing process, the current vehicle gets a rank for its neighbors with regard to the Euclidean distance between itself and the next node, the one-hop delay, and the connection time, and transmits the packet to a vehicle with the best rank. In the V2I forwarding strategy, RSU at the intersection forwards the data packet to the corresponding road section using a greedy approach. In this process, if there is no vehicle to reach the target point, the RSU keeps this packet until a suitable next-hop node is found.

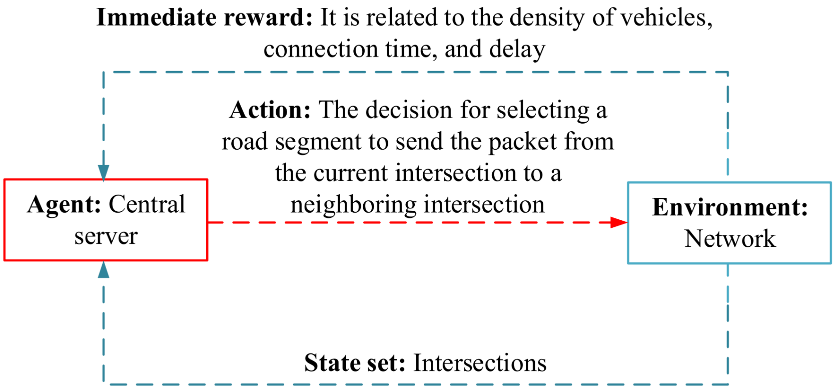
Immediate reward: It is related to the density of vehicles, connection time, and delay
Action: The decision for selecting a road segment to send the packet from the current intersection to a neighboring intersection
Agent: Central server
Environment: Network
State set: Intersections

Figure 24. Learning structure in IRQ.

6.17. QFHR

Rahmani et al. in [70] have offered a Q-learning and fuzzy-based hierarchical routing solution (QFHR) for VANETs. It includes three parts, namely traffic pattern recognition, Intersection-to-Intersection (I2I) routing, and routing at road sections. In the first part, each network node (vehicle or RSU) considers a neighbor table to keep the traffic conditions of road sections. This table is obtained from hello messages transmitted by the network nodes in the road sections. This message is disseminated on the network at a fixed time interval. In addition, RSUs also maintain a traffic table to be aware of the traffic pattern of road sections connected to their intersections at any moment. The task of RSUs is to implement a routing method based on Q-learning to explore different paths between network intersections. In this method, the intersections form the state set. This decreases its size and boosts the convergence rate. In this operation, each message acts as the agent, and the whole network expresses the learning environment. The action set also represents the four road sections connected to each intersection, i.e., the north road, the south road, the east road, and the west road. The reward function considers the density of vehicles, connection quality (obtained from the connection time and the hello reception rate), the congestion level, the road traveling time. See Figure 25. This solution gets a dynamic discount factor according to the density of vehicles and the distance to the destination while the proposed method considers a constant learning rate. Ultimately, vehicles in each road section use a greedy technique to choose the most suitable path in each road section. If this algorithm fails due to the local optimum problem, a fuzzy solution performs route recovery and selects the next node. This algorithm analyzes the chances of neighboring vehicles based on congestion level, connection quality and distance to the destination. This analysis is used for selecting relay node. In the last step, each RSU uses a greedy approach to transmit data packets to the corresponding road section. If the RSU cannot find any vehicle, then the RSU will hold the packet until a suitable next-hop node is found.

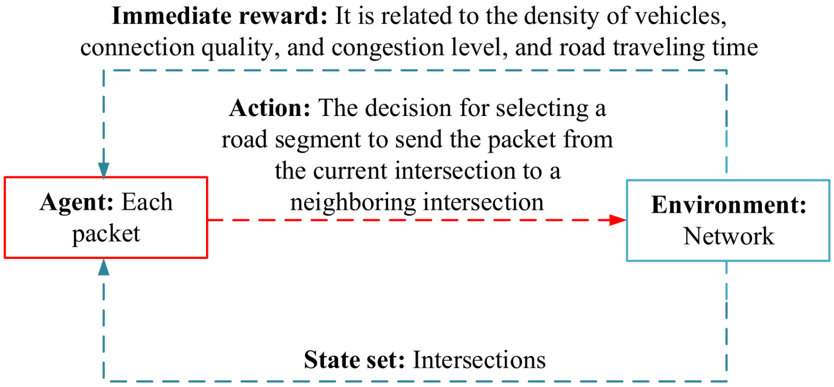
Immediate reward: It is related to the density of vehicles, connection quality, and congestion level, and road traveling time
Action: The decision for selecting a road segment to send the packet from the current intersection to a neighboring intersection
Agent: Each packet
Environment: Network
State set: Intersections

Figure 25. Learning structure in QFHR.

7. Discussion

Here, according to the routing approaches studied in Section 6, it can be deduced that the most common RL technique applied in most routing protocols is Q-learning. For example, based on Table 8, it can be found that IV2XQ, QAGR, CEPF, PFQ-AODV, QGrid, QTAR, Q-LBR, ECTS, GLS, RASAR, Wu et al., RHR, IRQ, QFHR have utilized the Q-learning algorithm to design their routing model. The reason for this issue is the simplicity and low computational complexity of Q-learning. The performance of this RL technique is related to the size of the state and action sets. If these sets are small, Q-learning is an appropriate option for modeling the routing process because it can obtain an optimal response in a short time. However, when there is a complicated learning environment, meaning that it has large state and action sets, the learning capability of the agent is greatly dropped, and the agent may explore the most suitable response in a long time or may never reach an optimal response. This means that the Q-learning algorithm has a low convergence speed in this condition. Researchers have suggested various ideas to solve this problem. Table 8 presents the most important parameters of the reviewed methods in summary. According to this table, it can be deduced that IV2XQ, QGrid, ECTS, GLS, IRQ, and QFHR have used an intersection-based routing technique. According to this idea, the intersections in the network environment are considered as the state space and the central controller is responsible for obtaining the most suitable path between the intersections. This idea makes a smaller state set and improves the convergence speed of the RL-based routing approaches. However, in these schemes, the central controller calculates the global path to reach the destination. They usually use historical information to discover the routes. As a result, these methods may be incompatible with real-time events such as accidents or gridlocks on roads in the network. Another idea considered in some routing methods such as RLRC, Wu et al., and SeScR is to use clustering techniques on the network. In this case, CH nodes perform the routing operation. As a result, the state and action sets in the routing algorithms are limited and their convergence speed is improved. Moreover, CEPF presents a novel idea for selecting edge nodes and sending data through these nodes. This idea improves the speed of route formation and reduces the number of transmitter nodes in the network. Another solution that some researchers have used is to utilize UAVs to improve the routing operations in VANET. For example, QAGR has used UAVs to calculate routing direction, which filters the neighboring ground nodes and makes a smaller state space, which will help QAGR to improve its convergence speed. Furthermore, in QTAR, the researchers have used two Q-learning-based routing techniques for finding global paths between RSUs at intersections and exploring local paths between vehicles on each road segment.

Also, according to Table 9, it can be deduced that the studied methods in this review support various communications in the network and are suitable for specific applications. For example, IV2XQ, QTAR, ECTS, Wu et al., SeScR, IRQ, and QFHR support V2V and V2I communications and are more suitable for urban environments. QAGR and Q-LBR methods use two communication types, V2V and vehicle-to-UAV (V2U) communication and require an aerial network to support the routing process. Therefore, these protocols can be deployed when this aerial infrastructure is available. Building this aerial network is costly and unsuitable for any applications. It is also difficult to manage this network and communication between different nodes, which makes the routing process very complex. In addition, some routing methods such as CEPF, PFQ-AODV, QGrid, RRPV, RLRC, GLS, RSAR, and RHR support only V2V communications. These algorithms are suitable for routing on highways and urban areas.

Table 10 presents a comparison between the routing approaches with regard to different equipment, like positioning systems and digital map. This equipment is used to calculate the location of vehicles. However, it is difficult to obtain the position of vehicles in some areas, such as tunnels. Also, the performance of a routing approach depends heavily on the positioning device. If this device cannot accurately calculate the location of vehicles, the performance of the routing method will not be desirable. Another point is that positioning systems are highly costly in terms of bandwidth consumption. Therefore, some routing methods such as PFQ-AODV and Q-LBR have attempted to design the routing process independent of this equipment. This idea is attractive and can be considered for designing routing methods in the future.

Table 8. Most important features of RL-based routing methods.

Scheme	RL Algorithm	Agent	State Space	Action Space	Reward Function	Learning Rate	Discount Factor
IV2XQ [54]	Multidimensional Q-learning	Central server	Intersections	Selecting road sections	φ, If reaching the destination intersection; 0, Otherwise	Fixed	Related to the vehicle density and distance to the destination
QAGR [55]	Q-learning	Vehicle	An $K \times M$ array including distance between a vehicle and its neighbors and neighboring degree.	Selecting the next-hop node (neighboring node)	Related to the received signal strength indication (RSSI), the transmission distance, and the collision between vehicles	Fixed	Fixed
CEPF [56]	Q-learning	Packet	Vehicles	Selecting the next-hop node (neighboring node)	1, If the node is a single-hop neighbor of the destination; 0, Else	Fixed	Fixed
PFQ-AODV [57]	Fuzzy constraint Q-learning	Packet	The set of single-hop and two-hop neighboring nodes	The set of single-hop nodes	1, If the node is a single-hop neighbor of the destination; 0, Otherwise	Fixed	Fixed
QGrid [58]	Q-learning	Virtual agent	Grids	Forwarding the message from one grid to the best next-hop grid.	100, If the destination is inside the grid; 0, Else	Fixed	Related to the number of vehicles in each grid

Table 8. *Cont.*

Scheme	RL Algorithm	Agent	State Space	Action Space	Reward Function	Learning Rate	Discount Factor
RRPV [59]	DynaQ	Vehicles	Forwarding (F) and delivering (D)	Selecting the best next-hop node	Related to the link quality and the Euclidean distance	Fixed	Fixed
QTAR [60]	Q-learning	Packets	The neighboring vehicles in the V2V Q-learning algorithm and the neighboring RSUs in the R2R Q-leaning algorithm	The set of neighboring vehicles in the V2V Q-learning algorithm and the set of neighboring RSUs in the R2R Q-learning algorithm	Related to link quality, link expiration time, and delay	Fixed	Fixed
Q-LBR [61]	Q-learning	UAV	Ground network congestion identifier (GNCI) and the UAV relay congestion identification (URCI)	Selecting the UAV routing policy area (URPA), including $URPA_{Lower}$ and $URPA_{Upper}$	Related to GNCI and URCI	Fixed	Fixed
ECTS [62]	Q-learning	Navigation center (NC)	Intersections	Selecting the data transmission route between neighboring intersections	1, If reaching the destination intersection; 0, Otherwise	Fixed	Related to the connectivity probability
RLRC [63]	SARSA-Lambda	CHs	The set of all cluster-head nodes in the network	Selecting an cluster-head node in the network	Related to link status, including bandwidth and inverse link duration	Fixed	Related to hop count
GLS [64]	Fuzzy-Q-learning	Controller	The set of all areas in the network	Selecting an area among the neighboring areas	Related to MD, OC, VD, and buffer capacity	Related to route length	Fixed

Table 8. *Cont.*

Scheme	RL Algorithm	Agent	State Space	Action Space	Reward Function	Learning Rate	Discount Factor
RSAR [65]	Q-learning	Each vehicle	The set of all nodes in the network	Transferring a beacon packet	1, If the next hop vehicle is a single-hop neighbor of the destination; 0, Otherwise	Related to number of hops and band-width.	Related to link reliability
Wu et al. [66]	Q-learning	Each vehicle	The set of all single-hop neighbors	Selecting the best next-hop node among single hop neighbors	1, If the next hop vehicle is a single-hop neighbor of RSU; 0, Otherwise	Fixed	Fixed
RHR [67]	Q-learning	Unknown	Vehicles	Receiving different kinds of packets related to the current next hop	Unknown	Fixed	Fixed
SeScR [68]	DDPG	Each vehicle	The location, velocity, and orientation of the vehicle	The decision for packet routing	Related to SDN controller	Fixed	Fixed
IRQ [69]	Q-learning	Central server	All intersections in the network	Road sections connected to the current intersection	Related to the density of vehicles, connection time, and delay	Fixed	Related to the density of the road sections and the distance to the destination
QFHR [70]	Q-learning	Each packet	All intersections in the network	Four road sections connected to each intersection	Related to the density of vehicles, connection quality, the congestion level, the road traveling time	Fixed	Related to the density of the road sections and the distance to the destination

Table 9. Comparison of the routing methods in terms of various communication types.

Scheme	V2V	V2I	V2U
IV2XQ [54]	✓	✓	×
QAGR [55]	✓	×	✓
CEPF [56]	✓	×	×
PFQ-AODV [57]	✓	×	×
QGrid [58]	✓	×	×
RRPV [59]	✓	×	×
QTAR [60]	✓	✓	×
Q-LBR [61]	✓	×	✓
ECTS [62]	✓	✓	×
RLRC [63]	✓	×	×
GLS [64]	✓	×	×
RSAR [65]	✓	×	×
Wu et al. [66]	✓	✓	×
RHR [67]	✓	×	×
SeScR [68]	✓	✓	×
IRQ [69]	✓	✓	×
QFHR [70]	✓	✓	×

Table 10. Comparison of various routing methods in terms of positioning systems.

Scheme	Positioning System	Digital Map
IV2XQ [54]	✓	✓
QAGR [55]	✓	×
CEPF [56]	✓	✓
PFQ-AODV [57]	×	×
QGrid [58]	✓	×
RRPV [59]	✓	✓
QTAR [60]	✓	×
Q-LBR [61]	×	×
ECTS [62]	✓	✓
RLRC [63]	✓	✓
GLS [64]	✓	✓
RSAR [65]	✓	×
Wu et al. [66]	✓	✓
RHR [67]	✓	×
SeScR [68]	✓	×
IRQ [69]	✓	✓
QFHR [70]	✓	✓

Additionally, Table 11 compares various routing methods based on their need to control messages. These control messages are exchanged between nodes to evaluate communication links and share spatial and velocity information of vehicles with other nodes in the network. However, control messages increase communication overhead, bandwidth consumption and other resources, network congestion, and delay in the routing process. Therefore, many researchers are trying to lower control messages exchanged on the network and control these broadcast messages. For example, IV2XQ and QGrid methods do not use any control messages to design the Q-Learning-based routing model. These schemes only broadcast beacon messages between vehicles to find the best route in each road segment. As a result, these methods have a suitable bandwidth consumption, low delay, and low communication overhead. Moreover, RHR has presented an adaptive broadcast technique to manage bandwidth consumption. According to this technique, the broadcast interval of the beacon message is dynamically determined using a broadcast control strategy.

Table 11. Comparison of various routing methods in terms of control messages.

Scheme	Control Message	Dissemination Process
IV2XQ [54]	Beacon	Broadcast (for single-hop neighbors locally)
QAGR [55]	$Hello_{V2U}$	Unicast
	$Hello_{V2V}$	Broadcast (for single-hop neighbors locally)
CEPF [56]	Hello	Broadcast (for single-hop and two-hop neighbors locally)
PFQ-AODV [57]	Hello	Broadcast (for single-hop and two-hop neighbors locally)
	RREQ	Broadcast (flooding)
	RREP	Unicast
QGrid [58]	Hello	Broadcast (for single-hop neighbors locally)
RRPV [59]	Hello	Broadcast (for single-hop neighbors locally)
QTAR [60]	$Hello_{V2V}$	Broadcast to neighboring vehicles locally
	$Hello_{R2R}$	Broadcast to neighboring RSUs locally
Q-LBR [61]	Hello	Broadcast to UAVs and sharing ground network congestion
	RREQ	Broadcast (Flooding)
	RREP	Unicast
ECTS [62]	Beacon	Broadcast (for single-hop neighbors locally)
RLRC [63]	Hello	Broadcast (for single-hop and two-hop neighbors locally)
	RREQ	Broadcast (Flooding)
	RREP	Unicast
GLS [64]	Hello	Broadcast (for single-hop and two-hop neighbors locally)
RSAR [65]	Hello	Broadcast (for single-hop neighbors locally)
	RREQ	Broadcast (Flooding)
	RREP	Unicast
Wu et al. [66]	Hello	Broadcast (for single-hop neighbors locally)
RHR [67]	Beacon	Broadcast (for single-hop neighbors locally)
SeScR [68]	Beacon	Broadcast (for single-hop neighbors locally)
IRQ [69]	Beacon	Broadcast (for single-hop neighbors locally)
	Traffic message	Unicast
QFHR [70]	Hello	Broadcast (for single-hop neighbors locally)

Table 12 compares various routing methods in terms of learning framework. According to this table, it can be found that the routing approaches such as IV2XQ, QAGR, CEPF, PFQ-AODV, QGrid, QTAR, ECTS, RLRC, GLS, RSAR, Wu et al., RHR, SeScR, IRQ, QFHR apply single-agent reinforcement learning in their routing process. These routing schemes are simpler than the multi-agent approaches and have less computational complexity. However, the learning capability and convergence rate of the agent in single-agent routing schemes is less those that in multi-agent approaches. According to Table 12, it can be deduced that only RRPV uses multi-agent technique. In this protocol, several agents (vehicles) interact with each other and share their experiences. This has increased the learning capability of this routing scheme, especially for large-scale networks. However, RRPV has more computational complexity than other single-agent approaches.

Table 12. Comparison of RL-based routing methods in terms of learning framework.

Scheme	Single-Agent Learning	Multi-Agent Learning
IV2XQ [54]	✓	×
QAGR [55]	✓	×
CEPF [56]	✓	×
PFQ-AODV [57]	✓	×
QGrid [58]	✓	×
RRPV [59]	×	✓
QTAR [60]	✓	×
Q-LBR [61]	✓	×
ECTS [62]	✓	×
RLRC [63]	✓	×
GLS [64]	✓	×
RSAR [65]	✓	×
Wu et al. [66]	✓	×
RHR [67]	✓	×
SeScR [68]	✓	×
IRQ [69]	✓	×
QFHR [70]	✓	×

Moreover, Table 13 compares the routing approaches in terms of learning model. According to this table, it can be found that most routing protocols such as IV2XQ, QAGR, CEPF, PFQ-AODV, QGrid, QTAR, ECTS, RLRC, GLS, RSAR, Wu et al., RHR, SeScR, IRQ, QFHR use a free-model reinforcement learning framework. In these methods, the agent estimates the value function according to the obtained experience and does not build any model from the learning environment. These routing schemes have less computational complexity compared to model-based routing methods. However, they need more experience compared to model-based techniques and are not flexible. Our studies show that RRPV is only model-based routing method. This method uses fuzzy logic to build the network model. Due to creating the network model, RRPV can estimate the value function with less interaction with the environment. Also, this routing method is more flexible against sudden changes in the environment. However, model-based techniques have higher computational complexity and require a lot of computational resources, and have poor performance for large-scale networks.

Table 13. Comparison of RL-based routing methods in terms of learning model.

Scheme	Model Based Learning	Free-Mode Learning
IV2XQ [54]	×	✓
QAGR [55]	×	✓
CEPF [56]	×	✓
PFQ-AODV [57]	×	✓
QGrid [58]	×	✓
RRPV [59]	✓	×
QTAR [60]	×	✓
Q-LBR [61]	×	✓
ECTS [62]	×	✓
RLRC [63]	×	✓
GLS [64]	×	✓
RSAR [65]	×	✓
Wu et al. [66]	×	✓
RHR [67]	×	✓
SeScR [68]	×	✓
IRQ [69]	×	✓
QFHR [70]	×	✓

Table 14 compares routing approaches with regard to learning algorithms. According to this table, IV2XQ, QAGR, CEPF, PFQ-AODV, QGrid, QTAR, Q-LBR, RRPV, ECTS, RLRC, GLS, RSAR, Wu et al., RHR, IRQ, and QFHR use traditional RL techniques. These methods have a good performance and find the optimal response with an acceptable convergence speed if the state and action sets are small. However, large state and action sets lead to slow convergence and more time to find an optimal response. SeScR utilizes a deep reinforcement learning technique to improve the learning rate. This method can solve complex computational operations in complex (large-scale) environments, such as VANETs, and speeds up the agent ability to optimize this policy. These routing protocols have a suitable performance for large networks, and their learning speed is good. There are few routing methods that use deep reinforcement learning techniques to improve the routing process. Due to rapid progress of technology and the emergence of new networks such as Internet of Vehicles (IoV), it is essential to design routing techniques that use deep reinforcement learning because these methods are successful in terms of convergence speed in larger state and action spaces.

In addition, Table 15 compares various routing approaches with regard to the learning process. In this table, in IV2XQ, QGrid, Q-LBR, ECTS, GLS, and IRQ, the reinforcement learning algorithm is executed by a central agent (for example a central server or a UAV) in the network. Then, the agent sends information about this learning process to the network nodes to begin the data transfer operation. This can reduce communication and computational overhead in the network because these methods do not need to exchange control messages between nodes. However, they may be involved with the single point of failure problem, meaning that, if the central agent cannot properly execute the RL-based routing algorithm, the network will be disrupted. Also, the central agent cannot adapt itself to the topology changes in a real-time manner. Therefore, when occurring events like accidents, these algorithms cannot predict real-time traffic conditions and adapt to sudden changes in the network. Also, based on Table 15, it can be found that in some methods like QAGR, CEPF, PFQ-AODV, RRPV, QTAR, RLRC, RSAR, Wu et al., RHR, SeScR, and QFHR, RL-based routing algorithms are locally executed in the network. However, in this case, computational cost and communication overhead are increased compared to centralized routing approaches because the network topology information is obtained locally by exchanging beacon messages between vehicles in the network. However, these methods are scalable, are more consistent with the dynamic network environment.

Table 14. Comparison of RL-based routing methods in terms of learning algorithm.

Scheme	Traditional Reinforcement Learning	Deep Reinforcement Learning
IV2XQ [54]	✓	×
QAGR [55]	✓	×
CEPF [56]	✓	×
PFQ-AODV [57]	✓	×
QGrid [58]	✓	×
RRPV [59]	✓	×
QTAR [60]	✓	×
Q-LBR [61]	✓	×
ECTS [62]	✓	×
RLRC [63]	✓	×
GLS [64]	✓	×
RSAR [65]	✓	×
Wu et al. [66]	✓	×
RHR [67]	✓	×
SeScR [68]	×	✓
IRQ [69]	✓	×
QFHR [70]	✓	×

Table 15. Comparison of RL-based routing schemes according to learning process.

Scheme	Centralized Learning	Distributed Learning
IV2XQ [54]	✓	×
QAGR [55]	×	✓
CEPF [56]	×	✓
PFQ-AODV [57]	×	✓
QGrid [58]	✓	×
RRPV [59]	×	✓
QTAR [60]	×	✓
Q-LBR [61]	✓	×
ECTS [62]	✓	×
RLRC [63]	×	✓
GLS [64]	✓	×
RSAR [65]	×	✓
Wu et al. [66]	×	✓
RHR [67]	×	✓
SeScR [68]	×	✓
IRQ [69]	✓	×
QFHR [70]	×	✓

Finally, Table 16 has categorized the RL-based routing schemes with regard to routing techniques. In this table, it can be found that IV2XQ, QGrid, GLS, QAGR, CEPF, RRPV, QTAR, RASR, IRQ, and QFHR are position-based routing approaches. These routing methods do not require information about the entire network and use local information to send data packets. As a result, they have low communication overhead and efficiently consume bandwidth and energy resources. In this type of routing technique, RRPV, QTAR, and RSAR are known as the DTN routing methods. This means that they use the store-carry-forward technique to transfer data packets to the target node. However, this technique has a good routing overhead; but it boosts delay in the data transfer operation. On the other hand, IV2XQ, GLS, QAGR, CEPF, IRQ, QFHR are known as the non-DTN routing protocols. This means that they use a greedy forwarding technique for data transfer. These methods have a good performance in dense networks. Moreover, they are scalable, have low routing overhead, and consume low memory and bandwidth. The most important challenge in these routing protocols is to accurately obtain location information of nodes because if the position of the nodes is not available and/or is not accurately calculated, the performance of these protocols will not be accurate. Note that QGrid uses two routing techniques, namely the greedy strategy (non-DTN routing scheme) and the Markov prediction method (DTN routing scheme) to discover routes in the road segments. On the other hand, RLRC, Wu et al., and SeScR use clustering techniques in their routing process, so that CH node manages the cluster and inter-cluster communication. These routing methods can greatly reduce routing messages exchanged in the network and prevent network congestion. However, the challenges of this type of routing protocol are CH selection and cluster management, especially for dynamic networks such as VANET. Furthermore, Q-LBR, ECTS, PFQ-AODV, and RHR are known as topology-based routing protocols, so that Q-LBR, ECTS, and PFQ-AODV are reactive routing methods. These methods are successful in terms of memory consumption, bandwidth, and routing overhead. However, they are faced with challenges such as high delay in the route discovery process, flooding control messages, and congestion on the network. On the other hand, RHR integrates proactive and reactive schemes and utilizes their benefits such as controlling routing overhead and lowering delay in the routing operation. It is appropriate for large-scale networks. Note that some routing protocols integrate position-based routing and topology-based routing. For example, the RL-based routing processes in some geographic routing protocols like IV2XQ, QGrid, GLS, and IRQ are executed in a proactive manner. Additionally, the routing process in QAGR, CEPF, RSAR, and QFHR integrate geographic and reactive routing methods. Finally, RLRC is a cluster-based reactive routing scheme. This means that the routing process between CH nodes is performed using a reactive manner.

Table 16. Comparison of RL-based routing methods in terms of routing techniques.

Scheme	Position-Based Routing		Cluster-Based Routing	Topology-Based Routing		
	DTN	Non-DTN		Proactive	Reactive	Hybrid
IV2XQ [54]	×	✓	×	✓	×	×
QAGR [55]	×	✓	×	×	✓	×
CEPF [56]	×	✓	×	×	✓	×
PFQ-AODV [57]	×	×	×	×	✓	×
QGrid [58]	✓	✓	×	✓	×	×
RRPV [59]	✓	×	×	×	×	×
QTAR [60]	✓	×	×	×	×	×
Q-LBR [61]	×	×	×	×	✓	×
ECTS [62]	×	×	×	×	✓	×
RLRC [63]	×	×	✓	×	✓	×
GLS [64]	×	✓	×	✓	×	×
RSAR [65]	✓	×	×	×	✓	×
Wu et al. [66]	×	×	✓	×	×	×
RHR [67]	×	×	×	✓	✓	✓
SeScR [68]	×	×	✓	×	×	×
IRQ [69]	×	✓	×	✓	×	×
QFHR [70]	×	✓	×	×	✓	×

8. Challenges and Open Issues

Designing routing schemes based on reinforcement learning is a serious research topic that must be regarded by researchers in the future. There are various challenges in designing and improving these routing methods in VANETs. Here, we express some of the most important challenges in this area.

- **Designing the broadcast control mechanisms:** Exchanging control messages periodically and flooding routing messages increase the use of bandwidth and lead to high overhead. Thus, designing broadcast control mechanisms is an essential need, which must be considered when presenting RL-based routing schemes. For example, researchers can dynamically adjust the broadcast time interval of control messages or filter some nodes with regard to factors such as link quality and movement information to control the flooding process of routing messages.
- **Limiting the state space:** Large state and action sets significantly lower the convergence rate and increase delay in the routing process. Therefore, researchers must focus on strategies to restrict state and action spaces such as filtering some states in accordance with specific criteria and employing clustering approaches.
- **Q-table:** Q-values obtained from a reinforcement learning algorithm are recorded in Q-table. The size of this table depends on the size of the state and action sets. Therefore, large state and action sets will sharply grow the dimensions of Q-table. This greatly increases the need for memory to maintain Q-table. Also, updating this table needs high latency, computational costs, and overhead. Thus, the management of Q-table size should be considered by in the future, for example, limiting the state space mentioned above can be considered a potential solution. Also, focusing on DRL-based solutions are useful.
- **Tradeoff between exploration and exploitation:** In the RL-based routing process, it is very important to dynamically adjust learning parameters to make a balance between exploration and exploitation. It can be considered by researchers in the future.

- **Multi-objective RL-based routing methods:** Most RL-based routing protocols focus on a quality of service (QoS) requirement. However, focusing on multi-objective routing protocols is a research subject that can be considered by researchers in the future. For example, designing the reward function based on several objectives such as delay and link quality.
- **Designing a predictive RL-based routing protocol:** This is a serious challenge, which must consider in the future. These protocols should be able to predict the latter positions of vehicles on the network to make routing decisions more accurately.
- **Testbed:** Most researchers use simulation tools such as network simulator version 2 (NS2), NS3, and MATLAB to analyze the routing approaches. However, these tools are not accurate and cannot evaluate the performance of these methods correctly. They should be simulated in real environments to evaluate their performance carefully. However, this is extremely expensive.
- **Scalability:** RL-based routing approaches should be suitable for different sizes of networks. However, when the network is large, it experiences a long delay in the data transfer operation. This weakens the performance of routing protocols. Therefore, designing scalable RL-based routing methods is a serious challenge that should review by researchers.

9. Conclusions

In this review paper, we studied a number of RL-based routing methods. Initially, we have outlined an introduction to Rl, Markov decision framework, and their characteristics. Then, we have categorized and briefly introduced RL-based routing approaches with regard to learning framework, learning model, learning algorithm, learning process, and the routing algorithm. In this paper, we attempted to help researchers find a new view of reinforcement learning applications to design routing schemes. This help researchers to properly and accurately understand how these routing methods are designed in VANETs and recognize existing challenges to try for removing them. In the future, we will study deep reinforcement learning and review its effect on the routing methods in VANETs. Today, these new approaches have been applied in various areas and presented promising results. We believe that these new approaches can be applied for designing routing aproaches in VANETs to enhance their performance and efficiency.

Author Contributions: Conceptualization, M.H. and A.M.R.; methodology, M.H. and J.L.; validation, A.M.R.; investigation, J.L. and A.M.R.; resources, M.H., A.M.R. and J.L.; writing—original draft preparation, J.L.; supervision, M.H.; project administration, M.H. and A.M.R. All authors have read and agreed to the published version of the manuscript.

Funding: The result was created through solving the student project ”Security analysis and developing lightweight ciphers and protocols” using objective-oriented support for specific university research from the University of Finance and Administration, Prague, Czech Republic.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Authors thank Michal Merta, and Zdeněk Truhlář for their help with the research connected with the topic of the article.

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

References

1. Boussoufa-Lahlah, S.; Semchedine, F.; Bouallouche-Medjkoune, L. Geographic routing protocols for Vehicular Ad hoc NETworks (VANETs): A survey. *Veh. Commun.* **2018**, *11*, 20–31. [CrossRef]
2. Rasheed, A.; Gillani, S.; Ajmal, S.; Qayyum, A. Vehicular ad hoc network (VANET): A survey, challenges, and applications. In *Vehicular Ad-Hoc Networks for Smart Cities*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 39–51. [CrossRef]

3. Campista, M.E.M.; Rubinstein, R.M.G. *Advanced Routing Protocols for Wireless Networks*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
4. Hartenstein, H.; Laberteaux, L. A tutorial survey on vehicular ad hoc networks. *IEEE Commun. Mag.* **2008**, *46*, 164–171. [CrossRef]
5. Nazib, R.A.; Moh, S. Routing protocols for unmanned aerial vehicle-aided vehicular ad hoc networks: A survey. *IEEE Access* **2020**, *8*, 77535–77560. [CrossRef]
6. Abdel-Halim, I.T.; Fahmy, H.M.A. Prediction-based protocols for vehicular Ad Hoc Networks: Survey and taxonomy. *Comput. Netw.* **2018**, *130*, 34–50. [CrossRef]
7. Khezri, E.; Zeinali, E. A review on highway routing protocols in vehicular ad hoc networks. *SN Comput. Sci.* **2021**, *2*, 1–22. [CrossRef]
8. Wlodarczak, P. *Machine Learning and Its Applications*; CRC Press: Boca Raton, FL, USA, 2019.
9. Mohammed, M.; Khan, M.B.; Bashier, E.B.M. *Machine Learning: Algorithms and Applications*; CRC Press: Boca Raton, FL, USA, 2016.
10. Mazyavkina, N.; Sviridov, S.; Ivanov, S.; Burnaev, E. Reinforcement learning for combinatorial optimization: A survey. *Comput. Oper. Res.* **2021**, *134*, 105400. [CrossRef]
11. Saravanan, M.; Ganeshkumar, P. Routing using reinforcement learning in vehicular ad hoc networks. *Comput. Intell.* **2020**, *36*, 682–697. [CrossRef]
12. Sun, Y.; Lin, Y.; Tang, Y. A reinforcement learning-based routing protocol in VANETs. *Commun. Signal Process. Syst.* **2019**, *463*, 2493–2500. [CrossRef]
13. Nazib, R.A.; Moh, S. Reinforcement learning-based routing protocols for vehicular ad hoc networks: A comparative survey. *IEEE Access* **2021**, *9*, 27552–27587. [CrossRef]
14. Mekrache, A.; Bradai, A.; Moulay, E.; Dawaliby, S. Deep reinforcement learning techniques for vehicular networks: Recent advances and future trends towards 6G. *Veh. Commun.* **2021**, *33*, 100398. [CrossRef]
15. Mchergui, A.; Moulahi, T.; Zeadally, S. Survey on Artificial Intelligence (AI) techniques for Vehicular Ad-hoc Networks (VANETs). *Veh. Commun.* **2021**, *34*, 100403. [CrossRef]
16. Frikha, M.S.; Gammar, S.M.; Lahmadi, A.; Andrey, L. Reinforcement and deep reinforcement learning for wireless Internet of Things: A survey. *Comput. Commun.* **2021**, *178*, 98–113. [CrossRef]
17. Althamary, I.; Huang, C.W.; Lin, P. A survey on multi-agent reinforcement learning methods for vehicular networks. In Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 1154–1159. [CrossRef]
18. Lansky, J.; Ali, S.; Rahmani, A.M.; Yousefpoor, M.S.; Yousefpoor, E.; Khan, F.; Hosseinzadeh, M. Reinforcement Learning-Based Routing Protocols in Flying Ad Hoc Networks (FANET): A Review. *Mathematics* **2022**, *10*, 3017. [CrossRef]
19. Coronato, A.; Naeem, M.; De Pietro, G.; Paragliola, G. Reinforcement learning for intelligent healthcare applications: A survey. *Artif. Intell. Med.* **2020**, *109*, 101964. [CrossRef] [PubMed]
20. Al-Rawi, H.A.; Ng, M.A.; Yau, K.L.A. Application of reinforcement learning to routing in distributed wireless networks: A review. *Artif. Intell. Rev.* **2015**, *43*, 381–416. [CrossRef]
21. Gronauer, S.; Diepold, K. Multi-agent deep reinforcement learning: A survey. *Artif. Intell. Rev.* **2022**, *55*, 895–943. [CrossRef]
22. Padakandla, S. A survey of reinforcement learning algorithms for dynamically varying environments. *ACM Comput. Surv. (CSUR)* **2021**, *54*, 1–25. [CrossRef]
23. Rezwan, S.; Choi, W. A survey on applications of reinforcement learning in flying ad-hoc networks. *Electronics* **2021**, *10*, 449. [CrossRef]
24. Sharma, S.; Kaul, A.; Ahmed, S.; Sharma, S. A detailed tutorial survey on VANETs: Emerging architectures, applications, security issues, and solutions. *Int. J. Commun. Syst.* **2021**, *34*, e4905. [CrossRef]
25. Wang, X.; Mao, S.; Gong, M.X. An overview of 3GPP cellular vehicle-to-everything standards. *GetMobile: Mob. Comput. Commun.* **2017**, *21*, 19–25. [CrossRef]
26. Al-shareeda, M.A.; Alazzawi, M.A.; Anbar, M.; Manickam, S.; Al-Ani, A.K. A Comprehensive Survey on Vehicular Ad Hoc Networks (VANETs). In Proceedings of the 2021 International Conference on Advanced Computer Applications (ACA), Maysan, Iraq, 25–26 July 2021; pp. 156–160. [CrossRef]
27. Karunathilake, T.; Förster, A. A Survey on Mobile Road Side Units in VANETs. *Vehicles* **2022**, *4*, 482–500. [CrossRef]
28. Ayyub, M.; Oracevic, A.; Hussain, R.; Khan, A.A.; Zhang, Z. A comprehensive survey on clustering in vehicular networks: Current solutions and future challenges. *Ad Hoc Netw.* **2022**, *124*, 102729. [CrossRef]
29. Chatterjee, T.; Karmakar, R.; Kaddoum, G.; Chattopadhyay, S.; Chakraborty, S. A survey of VANET/V2X routing from the perspective of non-learning-and learning-based approaches. *IEEE Access* **2022**, *10*, 23022–23050. [CrossRef]
30. Belamri, F.; Boulfekhar, S.; Aissani, D. A survey on QoS routing protocols in Vehicular Ad Hoc Network (VANET). *Telecommun. Syst.* **2021**, *78*, 117–153. [CrossRef]
31. Shahwani, H.; Shah, S.A.; Ashraf, M.; Akram, M.; Jeong, J.P.; Shin, J. A comprehensive survey on data dissemination in Vehicular Ad Hoc Networks. *Veh. Commun.* **2021**, *34*, 100420. [CrossRef]
32. Yousefpoor, M.S.; Barati, H. DSKMS: A dynamic smart key management system based on fuzzy logic in wireless sensor networks. *Wirel. Netw.* **2020**, *26*, 2515–2535. [CrossRef]

33. Yousefpoor, M.S.; Barati, H. Dynamic key management algorithms in wireless sensor networks: A survey. *Comput. Commun.* **2019**, *134*, 52–69. [CrossRef]
34. Yousefpoor, E.; Barati, H.; Barati, A. A hierarchical secure data aggregation method using the dragonfly algorithm in wireless sensor networks. *Peer- Netw. Appl.* **2021**, *14*, 1917–1942. [CrossRef]
35. Busoniu, L.; Babuska, R.; De Schutter, B. A comprehensive survey of multiagent reinforcement learning. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2008**, *38*, 156–172. [CrossRef]
36. Nguyen, T.; Nguyen, N.; Nahavandi, S. Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. *IEEE Trans. Cybern.* **2020**, *50*, 3826–3839. [CrossRef]
37. Kalakanti, A.K.; Verma, S.; Paul, T.; Yoshida, T. RL SolVeR pro: Reinforcement learning for solving vehicle routing problem. In Proceedings of the 2019 1st International Conference on Artificial Intelligence and Data Sciences (AiDAS), Ipoh, Malaysia, 19 September 2019; pp. 94–99.
38. Vinayakumar, R.; Soman, K.; Poornachandran, P. Applying deep learning approaches for network traffic prediction. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; pp. 2353–2358.
39. Drummond, N.; Niv, Y. Model-based decision making and model-free learning. *Curr. Biol.* **2020**, *30*, R860–R865. [CrossRef]
40. Agostinelli, F.; Hocquet, G.; Singh, S.; Baldi, P. From reinforcement learning to deep reinforcement learning: An overview. In *Braverman Readings in Machine Learning. Key Ideas from Inception to Current State*; Springer: Cham, Switzerland, 2018; pp. 298–328. [CrossRef]
41. Sewak, M.; Sahay, S.K.; Rathore, H. Policy-Approximation Based Deep Reinforcement Learning Techniques: An Overview. In *Information and Communication Technology for Competitive Strategies (ICTCS 2020)*; Springer: Singapore, 2022; pp. 493–507. [CrossRef]
42. Chen, Y.R.; Rezapour, A.; Tzeng, W.G.; Tsai, S.C. RL-routing: An SDN routing algorithm based on deep reinforcement learning. *IEEE Trans. Netw. Sci. Eng.* **2020**, *7*, 3185–3199. [CrossRef]
43. Luong, N.C.; Hoang, D.T.; Gong, S.; Niyato, D.; Wang, P.; Liang, Y.C.; Kim, D.I. Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3133–3174. [CrossRef]
44. Benamar, N.; Singh, K.D.; Benamar, M.; El Ouadghiri, D.; Bonnin, J.M. Routing protocols in vehicular delay tolerant networks: A comprehensive survey. *Comput. Commun.* **2014**, *48*, 141–158. [CrossRef]
45. Mangrulkar, R.; Atique, M. Routing protocol for delay tolerant network: A survey and comparison. In Proceedings of the 2010 International Conference on Communication Control and Computing Technologies, Nagercoil, Tamil Nadu, India, 7–9 October 2010; pp. 210–215. [CrossRef]
46. Wu, C.; Yoshinaga, T.; Bayar, D.; Ji, Y. Learning for adaptive anycast in vehicular delay tolerant networks. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 1379–1388. [CrossRef]
47. He, J.; Cai, L.; Pan, J.; Cheng, P. Delay analysis and routing for two-dimensional VANETs using carry-and-forward mechanism. *IEEE Trans. Mob. Comput.* **2017**, *16*, 1830–1841. [CrossRef]
48. Karthikeyan, L.; Deepalakshmi, V. Comparative study on non-delay tolerant routing protocols in vehicular networks. *Procedia Comput. Sci.* **2015**, *50*, 252–257. [CrossRef]
49. Sharef, B.T.; Alsaqour, R.A.; Ismail, M. Vehicular communication ad hoc routing protocols: A survey. *J. Netw. Comput. Appl.* **2014**, *40*, 363–396. [CrossRef]
50. Saleem, Y.; Yau, K.L.A.; Mohamad, H.; Ramli, N.; Rehmani, M.; Ni, Q. Clustering and reinforcement-learning-based routing for cognitive radio networks. *IEEE Wirel. Commun.* **2017**, *24*, 146–151. [CrossRef]
51. Wheeb, A.H.; Nordin, R.; Samah, A.; Alsharif, M.H.; Khan, M.A. Topology-based routing protocols and mobility models for flying ad hoc networks: A contemporary review and future research directions. *Drones* **2021**, *6*, 9. [CrossRef]
52. Ajaz, F.; Naseem, M.; Ahamad, G.; Khan, Q.R.; Sharma, S.; Abbasi, E. Routing protocols for internet of vehicles: A review. In *AI and Machine Learning Paradigms for Health Monitoring System*; Springer: Singapore, 2021; pp. 95–103. [CrossRef]
53. Di Maio, A.; Palattella, M.; Engel, T. Performance Analysis of MANET Routing Protocols in Urban VANETs. *Ad-Hoc Mob. Wirel. Netw.* **2019**, *11803*, 432–451. [CrossRef]
54. Luo, L.; Sheng, L.; Yu, H.; Sun, G. Intersection-Based V2X Routing via Reinforcement Learning in Vehicular Ad Hoc Networks. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 5446–5459. [CrossRef]
55. Jiang, S.; Huang, Z.; Ji, Y. Adaptive UAV-assisted geographic routing with q-learning in VANET. *IEEE Commun. Lett.* **2020**, *25*, 1358–1362. [CrossRef]
56. An, C.; Wu, C.; Yoshinaga, T.; Chen, X.; Ji, Y. A context-aware edge-based VANET communication scheme for ITS. *Sensors* **2018**, *18*, 2022. [CrossRef]
57. Wu, C.; Ohzahata, S.; Kato, T. Flexible, portable, and practicable solution for routing in VANETs: A fuzzy constraint Q-learning approach. *IEEE Trans. Veh. Technol.* **2013**, *62*, 4251–4263. [CrossRef]
58. Li, F.; Song, X.; Chen, H.; Li, X.; Wang, Y. Hierarchical routing for vehicular ad hoc networks via reinforcement learning. *IEEE Trans. Veh. Technol.* **2018**, *68*, 1852–1865. [CrossRef]
59. Jafarzadeh, O.; Dehghan, M.; Sargolzaey, H.; Esnaashari, M.M. A Model-Based Reinforcement Learning Protocol for Routing in Vehicular Ad hoc Network. *Wirel. Pers. Commun.* **2022**, *123*, 975–1001. [CrossRef]

60. Wu, J.; Fang, M.; Li, H.; Li, X. RSU-assisted traffic-aware routing based on reinforcement learning for urban vanets. *IEEE Access* **2020**, *8*, 5733–5748. [CrossRef]
61. Roh, B.S.; Han, M.H.; Ham, J.H.; Kim, K.I. Q-LBR: Q-learning based load balancing routing for UAV-assisted VANET. *Sensors* **2020**, *20*, 5685. [CrossRef]
62. Li, G.; Gong, C.; Zhao, L.; Wu, J.; Boukhatem, L. An efficient reinforcement learning based charging data delivery scheme in VANET-enhanced smart grid. In Proceedings of the 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), Busan, Republic of Korea, 19–22 February 2020; pp. 263–270. [CrossRef]
63. Bi, X.; Gao, D.; Yang, M. A reinforcement learning-based routing protocol for clustered EV-VANET. In Proceedings of the 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 12–14 June 2020; pp. 1769–1773. [CrossRef]
64. Zhao, L.; Bi, Z.; Lin, M.; Hawbani, A.; Shi, J.; Guan, Y. An intelligent fuzzy-based routing scheme for software-defined vehicular networks. *Comput. Netw.* **2021**, *187*, 107837. [CrossRef]
65. Zhang, D.; Zhang, T.; Liu, X. Novel self-adaptive routing service algorithm for application in VANET. *Appl. Intell.* **2019**, *49*, 1866–1879. [CrossRef]
66. Wu, C.; Yoshinaga, T.; Ji, Y.; Zhang, Y. Computational intelligence inspired data delivery for vehicle-to-roadside communications. *IEEE Trans. Veh. Technol.* **2018**, *67*, 12038–12048. [CrossRef]
67. Ji, X.; Xu, W.; Zhang, C.; Yun, T.; Zhang, G.; Wang, X.; Wang, Y.; Liu, B. Keep forwarding path freshest in VANET via applying reinforcement learning. In Proceedings of the 2019 IEEE First International Workshop on Network Meets Intelligent Computations (NMIC), Dallas, TX, USA, 7–9 July 2019; pp. 13–18. [CrossRef]
68. Nahar, A.; Das, D. SeScR: SDN-Enabled Spectral Clustering-Based Optimized Routing Using Deep Learning in VANET Environment. In Proceedings of the 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 24–27 November 2020; pp. 1–9. [CrossRef]
69. Khan, M.U.; Hosseinzadeh, M.; Mosavi, A. An Intersection-Based Routing Scheme Using Q-Learning in Vehicular Ad Hoc Networks for Traffic Management in the Intelligent Transportation System. *Mathematics* **2022**, *10*, 3731. [CrossRef]
70. Rahmani, A.M.; Naqvi, R.A.; Yousefpoor, E.; Yousefpoor, M.S.; Ahmed, O.H.; Hosseinzadeh, M.; Siddique, K. A Q-Learning and Fuzzy Logic-Based Hierarchical Routing Scheme in the Intelligent Transportation System for Smart Cities. *Mathematics* **2022**, *10*, 4192. [CrossRef]