EasyChair Preprint

№ 1029

# Keep Forwarding Path Freshest in VANET via Applying Reinforcement Learning

Xuefeng Ji, Wenquan Xu, Chuwen Zhang, Tong Yun, Gong Zhang, Xiaojun Wang, Yunsheng Wang and Bin Liu



May 27, 2019

# Keep Forwarding Path Freshest in VANET via Applying Reinforcement Learning

Xuefeng Ji*, Wenquan Xu*, Chuwen Zhang*, Tong Yun*, Gong Zhang‡, Xiaojun Wang§, Yunsheng Wang¶, Bin Liu*
*Tsinghua National Laboratory for Information Science and Technology (TNList), Tsinghua University
‡Huawei Future Network Theory Lab, Hong Kong §Dublin City University ¶Kettering University, MI USA

***Abstract*—In Vehicular Ad Hoc NETworks (VANET), dynamic topology changes of network and inconstant bandwidth make it hard to maintain an end-to-end path to complete long-time stable data transmission. Facing this challenge, researchers have proposed the hybrid routing approach, which tries to combine both the advantages of recalculating route when topology changes and looking up routing table as long as the network topology is relatively stable. However, the existing hybrid routing algorithms can easily cause the $blind\ path$ problem, that is a route entry in the routing table becomes invalid before it expires due to timeout, because the next hop is already unavailable before timeout. To address this issue, we propose a Reinforcement learning based Hybrid Routing algorithm (RHR) that can online track the available paths with their status and use packet-carry-on information as real-time feedback to guide routing. RHR keeps the forwarding path always the freshest and thus improves the system performance. Simulation results show that RHR achieves better performance in packet delivery ratio (PDR), roundtrip time (RTT) and overhead than other peers under different scenarios of network scale, request frequency and vehicle velocity.**

***Index Terms*—routing algorithm, reinforcement learning, VANET**

## I. Introduction

With the development of Internet of Things (IoT), the era of $Internet\ of\ Everything$ is coming. As a branch of IoT, Intelligent and Connected Vehicles (ICV) have attracted increasing attention in recent years. In this trend, how to maintain an uninterrupted end-to-end real-time communication among vehicles has become a challenging problem.

Efficient, accurate and reliable routing protocols play an important role in achieving the above goal. Among previous studies, researchers proposed the hybrid routing algorithms [1], [2] which combine topology-based method with location-based approach to improve the VANET performance. In order to minimize the route re-calculation frequency of vehicle nodes during end-to-end communication, existing hybrid routing algorithms preset an expiration time for every route entry once a new path is generated. Although the expiration time is adjustable at the system initialization, it is fixed once being set. GPSR [3] and AODV [4] are two extreme cases. The former presents the calculating-route while the latter is lookup-route. Calculating-route works on per-packet basis, which will obviously bring high computation overhead as the transmission rate goes high. As to the lookup-route, the current approach is to purely try one fixed path towards destination in forwarding table maintained by a timeout mechanism. It can easily happen that an entry in the route table is still within its lifetime, but actually the corresponding node is unreachable to destination because it either runs out of the communication range or goes failure. This explains how $blind\ path$ generates.

To address the above $blind\ path$ problem, in this paper, we propose a Reinforcement learning based Hybrid Routing algorithm (RHR) that utilizes reinforcement learning to online track the available paths with their status and use the packet-carry-on information in VANET as real-time feedback. RHR will potentially generate multiple paths adaptively and dynamically upon the current network conditions, while setting each path an appropriate random value as an initial parameter for reinforcement learning. During the communication, some data information from the back and forth packets will be abstracted as $seeds$ to guide and optimize the routing selection (changing the weight of some paths). In this way, RHR is able to select the current best path to forward packets. Particularly, we made the following contributions:

1) We have exposed the $blind\ path$ problem in the current hybrid routing algorithms and proposed a solution to always keep the latest valid next hop in forwarding table during each lookup process;

2) We design and implement RHR, a hybrid routing algorithm based on the combination of calculating-route and lookup-route, which uses the improved $low\text{-}cost$ Q-Learning algorithm, broadcast control and obstacle-avoiding strategies to make the communication between vehicles more stable and reliable;

3) We evaluate the performance of RHR and compare it with representative VANET routing protocols under different experimental conditions on the NS-3 [5] simulation platform. The experimental results indicate that RHR shows the best performance in all spectrums.

The rest of this paper is organized as follows. Section II raises and analyzes the $blind\ path$ problem in current hybrid routing protocols. Section III describes how to use Q-Learning on traditional hybrid routing together with some other improved strategies. Section IV conducts the simulation experiments. Section V surveys the related work. Finally, Section VI concludes the paper.

## II. Illustration of the Blind Path Problem

In this section, we will show the $blind\ path$ problem in existing hybrid routing protocols. As shown in Figure 1, node $S$ wants to send packets to $D$. At first, $S$ chooses the path $S$-$N_1$-$D$ at time $T_0$, because $D$ is in the covering scope of $N_1$ and this path is the shortest one and presents good quality.

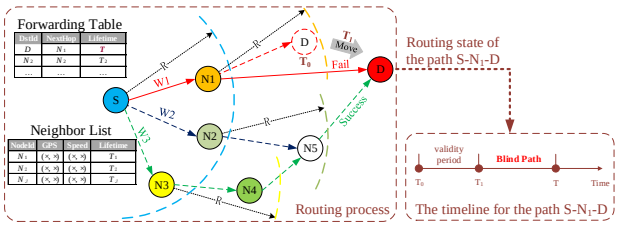



Fig. 1: Limitations of traditional routing methods

When $D$ moves away from $N_1$'s coverage at time $T_1$, the corresponding entry from $S$ to $D$ in $N_1$'s forwarding table should be changed to an updated new one immediately. But it is kept non-updated until the predefined period is expired. So the path $S$-$N_1$-$D$ becomes a *blind path* from time $T_1$ to $T$. Obviously, before a new path is re-calculated, the packets from $S$ to $D$ via $N_1$ will be lost.

In VANET, supposing each node has several neighbors, there is more than one path to the destination. Hence, we utilize reinforcement learning to let $S$ constantly track the available paths via exploring the packet-carry-on information. In our design, each path has a weight, representing the path quality (such as $W_1$, $W_2$ and $W_3$ in Figure 1 and higher value means better quality). As the example in Figure 1, $S$ uses calculating-route method to get three available paths to $D$: $S$-$N_1$-$D$, $S$-$N_2$-$N_5$-$D$ and $S$-$N_3$-$N_4$-$N_5$-$D$. When $S$ finds that the path $S$-$N_1$-$D$ will be no longer valid, it will check which one among the remaining paths is the best. Here, $S$ turns the path to $S$-$N_3$-$N_4$-$N_5$-$D$, as its path quality $W_3$ is higher than $W_2$, meaning it performs better than path $S$-$N_2$-$N_5$-$D$, though the latter path has fewer hops.

## III. Reinforcement Learning Based Hybrid Routing and Other Improvements

In this section, we will first introduce the Reinforcement learning based Hybrid Routing algorithm (RHR) in detail, which aims at solving the *blind path* problem that commonly exists in hybrid routing protocols presented in Section II. Then, we will introduce our another work on controlling broadcast overhead in Section III-B. Subsequently, we will describe the obstacle-avoiding strategy in Section III-C.

### A. Optimize routing using low-cost Q-Learning

In the field of robotics research [6], researchers often use reinforcement learning to train a robot to move and avoid obstacles, that is, to reward or punish their behaviors based on whether the chosen path is good. Moreover, the VANET environment has rich packet information which can be used as the data source to guide the behavior of vehicles through data-mining methods. Inspired by that, we utilize reinforcement learning to help make better routing decisions. Due to the movement of vehicles, the network topology changes continuously. Thus, in the vehicle intensive environment, relying only on one path makes it hard to keep the best transmission during vehicle movement. We try to explore multiple paths (if exist) simultaneously to a specific destination and run the reinforcement learning mechanism for each route in the forwarding table. It utilizes different kinds of data packets to $reward$ the routes that can benefit packet forwarding and $punish$ those ones with more broadcast overhead or failures. The forwarding table is dynamically maintained via data-mining on the back and forth data packets, with the purpose of finding the current best forwarding path. Only when there are not enough alternative paths, will the route re-calculation be triggered to replenish the forwarding table.

*1) Two-level Forwarding Table:* In our design, each route to a specific destination will maintain the optimal $K$ available nodes as optional next hops during the reinforcement learning process. That means there are multiple alternative paths toward the destination for any arrival packets (unless only one path exists). We design a two-level forwarding table to record this new routing structure as shown in Table I. It contains a special sub-table that can store multiple next-hop information (ID, weight and lifetime) for each routing entry.

*2) Reinforcement Learning Strategy:* Figure 2 shows the basic principle of reinforcement learning: if a certain action $a_t$ of an agent brings a positive reward $r_t$, then the strategy to generate this action will be strengthened. The duty of the agent is to seek the optimal strategy in each discrete state to maximize the expectation summation of the discount reward.

TABLE I: Two-level Forwarding Table

| Dst_ID | Next_Hop_ID | Weight | Lifetime |
|---|---|---|---|
| A | D | 100 | 1007 |
| | E | 85 | 1006 |
| | F | 64 | 1006 |
| B | H | 86 | 1007 |
| | J | 50 | 1005 |

TABLE II: Parameter settings of Q-Learning

| Parameter | Value |
|---|---|
| $R$ Value of $A$ Packets | -5 |
| $R$ Value of $B$ Packets | -10 |
| $R$ Value of $C$ Packets | 10 |
| $R$ Value of $D$ Packets | 5 |
| Learning Rate $\alpha$ | 0.2 |
| Discount Factor $\gamma$ | 0.8 |
| Exploration Probability $\varepsilon$ | 0.01 |

We observe that the traditional reinforcement learning problem usually has the following characteristics:

- Different actions incur different rewards;
- Reward has delay;
- The reward of an action relies on the current state.

These three features are very similar to the routing process in VANET. To begin with, different selections of next hop can be regarded as different states. Secondly, receiving different kinds of packets related to the current next hop corresponds to different actions. For these actions in different states, vehicle nodes will get feedback through the packet-carry-on information in VANET, which can guide them making better choices in the routing process.

We use the $low\text{-}cost$ Q-Learning theory [7] to model and analyze this mapping relation. The process of optimizing routing through Q-Learning is shown in Figure 3. Different actions will have diverse influence on the forwarding table. Routing weight that changes with these actions in different states will serve as the feedback to guide routing. Further, we reduce the overhead of this algorithm by simplifying Q-table and conditional learning strategy.

First, the core of Q-Learning is Q-table. The value $Q(s, a)$ in the Q-table measures how well the action $a$ is taken under the current state $s$. In the traditional Q-Learning method, there may be numerous states in real situations, so the scale of Q-table is hard to control. [8] proposed a scheme to use neural network to compress the scale of Q-table. But this approach is complex and incurs additional overhead. In our design, we adopt the idea of combining Q-table and forwarding table.

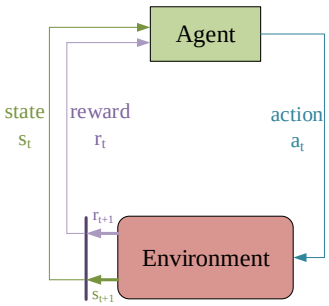



Fig. 2: Reinforcement Learning Schematic

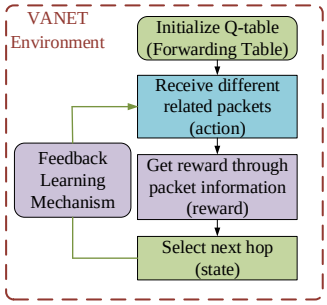



Fig. 3: Q-learning in Optimizing Routing

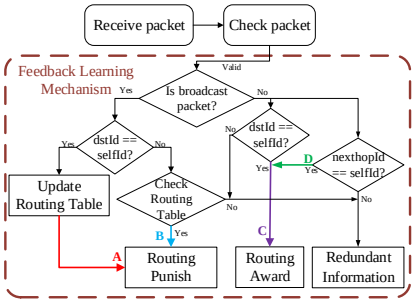



Fig. 4: Feedback Learning Mechanism

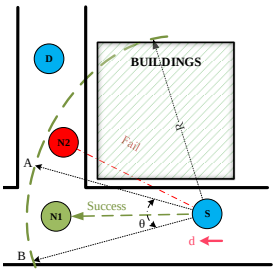



Fig. 5: Communication Sector

Compared with the multiple states in traditional method, the selections of next hop for current node in RHR are limited to $K$ nodes which refers to $K$ states. For ensuring the time-validity, vehicles only need to save the information of those selectable next-hop nodes within their lifetimes. Therefore, the size of Q-table is ideally controllable in VANET.

Second, it is worth noting that we use the conditional learning strategy to make the learning process *low-cost*. When the traffic pressure is beyond the threshold, we will use the probability $p$ calculated by $\frac{Threshold}{RequestsPerSecond}$ to sample data packets to reduce the overhead of reinforcement learning algorithm.

Subsequently, the main challenge of applying our algorithm to the routing process in VANET is how to establish a feedback mechanism according to the corresponding state.

*3) Feedback Learning Mechanism:* We optimize the routing by data-mining on the information of correctly calibrated data packets. While there are countless combinations of domains for the data packets, most of them are unnecessary for RHR. We only need the $DestinationID$ domain contained in the header of most routing protocols to increase the applicability of our algorithm. To sum up, we use $A$, $B$, $C$ and $D$ four kinds of packets as learning data, as shown in Figure 4.

If the current node receives a broadcast packet which goes toward itself, the reverse route to source via the last hop node will gain negative reward ($A$). Because this route is built up by the means of broadcasting, it is not an optimal route intuitively. On the contrary, if the received broadcast packet's destination is not the current node and there is a route to that destination in the current forwarding table, we will check whether the last hop of the received packet is on the way of that route. If so, it indicates that the route is not reliable and negative reward will be added to the path ($B$).

In the other case, if the current node receives a unicast packet which goes toward itself (destination or next hop), that reverse route through the last hop should be rewarded because it completes a successful data transfer ($C$ or $D$). And all the intermediate nodes on this path will be rewarded accordingly.

In conclusion, these relevant packets can be divided into two categories, one is positively related ($C$ and $D$) while the other is negatively related ($A$ and $B$). In the learning process, we use the Algorithm 1 to update the forwarding table.

To begin with, we use random values to initialize the Q-Value of top $K$ neighbor nodes closest to the destination in forwarding table (Random values must be kept in the same order as the top $K$ neighbors). Secondly, we use the $\varepsilon$-greedy strategy to select a next hop. The selection strategy is that each state has the probability of $\varepsilon$ to explore (select the next hop randomly), and the remaining $(1-\varepsilon)$ probability to develop (select the next hop with the largest utility value). Then the corresponding Q-Value in the table will be updated according to the related packet information. In Algorithm 1, $\alpha$ and $\gamma$ represent learning rate and discount factor respectively. Higher $\alpha$ makes the retention of previous training result less effective. And the larger the $\gamma$, the greater the effect of $max_a Q(s', a)$ we can get, which represents the maximum utility value of the corresponding forwarding node $s'$ record in the routing process. Through this method, the Q-table is updated to gradually converge to the optimal routing.

**Algorithm 1** Q-Learning process in RHR

1: Initialize Q-Value in forwarding table randomly
2: **for** each forwarding process **do**
3: Select the next hop according to the $\varepsilon$-greedy strategy
4: Obtain rewards ($R$) via the related packet information
5: $Q(s,a) \leftarrow (1-\alpha) \times Q(s,a) + \alpha \times [R + \gamma \times max_a Q(s', a)]$
6: **end for**

We use a specific example to describe the process of our algorithm. Suppose Table I is the forwarding table of a vehicle node at a certain moment. $Q(D, A)$ denotes the weight that node $D$ is selected as the next-hop to destination $A$ (measuring the value of this choice). When the node receives a positive related packet towards node $D$, it updates the utility value in the forwarding table based on the learning rate $\alpha$, discount factor $\gamma$ and the $reward$ value $R$ according to Equation 1 (The specific values of $\alpha$, $\gamma$ and $R$ will be described in Section IV).

$$Q(D,A) = (1-\alpha) \times Q(D,A) + \alpha \times [R + \gamma \times max\{Q(E,A), Q(F,A)\}] \quad (1)$$

Based on this feedback mechanism, the vehicle node can gradually maintain these stable routes and mark them with higher weights, meanwhile those poor routes will be phased out in the selection process.

### *B. Broadcasting control*

In VANET, the control of broadcast packets has always been a hot research issue [9], [10]. Without effective broadcast control, the route to destination will contain more hops, or even cause routing loops, which will decrease the communication quality of the network. Hence, in this section, we will introduce our broadcasting control strategy from two aspects.

*1) Protocol Packet:* In our design, each node needs to periodically broadcast the $Beacon$ packet to inform neighbor nodes of its location and speed. In order to effectively reduce the bandwidth consumption of $Beacon$ by controlling its number, we design an adaptive broadcast strategy that uses vehicle position prediction algorithm [11] to calculate the future positions of the neighbor nodes. The time interval of the broadcast is determined by the accuracy of prediction. If the motion of vehicle node is regular in a certain period of time and the predicted result is accurate, then the broadcast interval will be increased, otherwise we will reduce the interval.

*2) Data Packet:* When a vehicle node wants to send a data packet, if it cannot find a suitable entry in its forwarding table and has no destination's location information for calculation, it will broadcast that packet. We set the $TTL$ (Time to Live) of the broadcast packet to be slightly smaller than the normal data packet to reduce flooding. Furthermore, if the target is the neighbor of the forwarding node, the broadcast packet can be sent to it as a unicast packet. At the same time, if the vehicle node with a large number of neighbors rebroadcasts the received broadcast packet every time, it will cause heavy flooding. Therefore, when a node judges whether to perform rebroadcast, it is necessary to check the number of its neighbors. Only when this number is within the threshold, shall the broadcast continue. Otherwise, this node will stop rebroadcasting.

### C. Obstacle-avoiding strategy

The geographical environment of VANET is very complicated, and the communication between vehicles is often affected by the roadside buildings or other obstacles. In order to make the communication process more stable and reliable, we design an obstacle-avoiding strategy.

As shown in Figure 5, we present the concept of $communication\ sector$. When the vehicle selects next hop, the $communication\ sector$ of the node is constructed according to the direction of vehicle's motion $d$, the appropriate angle $\theta$ and the communication radius $R$. Only nodes in this area can be selected as the next hop. This setting allows the vehicle to communicate in the direction towards destination.

## IV. Simulation

In this section, we will verify the performance of RHR in different scenarios and compare it with the topology based routing protocol AODV [4], the position based routing protocol IGPSR [12] and the hybrid routing protocol RHR-simple which runs without reinforcement learning. Detailed configuration and experimental results are illustrated as below.

### A. Experimental scenarios

For simulating the actual traffic environment, we use OpenStreetMap [4] to generate an area map of approximately $1500m \times 900m$ from the real world, which contains main roads and a dozen of crossroads. Meanwhile, we use the SUMO [13], a road traffic simulation tool to generate traces of vehicle motion and use the NS-3 [5] network simulation tool to conduct experiments. As to the wireless configuration, we adopt the IEEE 802.11p with DCF standard in the MAC layer and set the log distance propagation loss model in the physical layer. According to the parameter analysis in section III-A3 and experimental debugging, the specific settings of Q-Learning in RHR during the simulation process are shown in Table II.

As for the configuration of application layer, we choose five pairs of vehicles to run the basic client-server application, in which each client periodically sends request packets (with 64-byte payload) to its corresponding server, and then the server will instantly reply data packets (with 1000-byte payload) to the client when receiving a request. Next, we conduct experiments under different scenarios of network scale, request frequency and vehicle velocity. Finally, we take Packet Delivery Ratio (PDR), Round Trip Time (RTT) and Normalized Routing Overhead (NRO) as metrics in the experiments.

- $PDR$: the ratio of successfully delivered packets to the destination over the total sent packets from the source.
- $RTT$: the experienced time from launching a request to receiving the corresponding reply.
- $NRO$: equals to the count of protocol packets (Rsf) divided by the total number of data packets (Psf), which is denoted as $\frac{Rsf}{Psf}$. This indicator reflects the efficiency of a protocol in a macro sense.

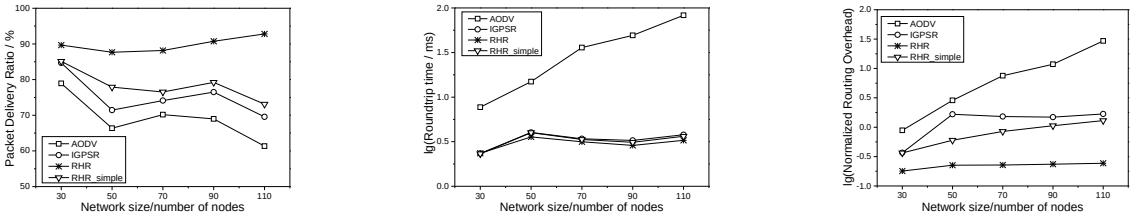
### B. Analysis of Experimental Results

In this section, we conduct experiments by changing the density of vehicle nodes in network (from 30 nodes to 110 nodes), the number of packets per second (from 20 pps to 100 pps) and the maximum moving speed of vehicles (from 10 m/s to 50 m/s). We take the logarithm of RTT and NRO to better reflect the difference of performance between different protocols. And we verify the effect of our optimization strategies. The experimental results are analyzed and compared based on the metrics in Section IV-A.

*1) Results on Network Size:* As shown in Figure 6(a), as the number of nodes in the network increases, RHR has the best PDR and remains relatively stable. This is because it well uses the packet-carry-on information on the fly in the link. Then, with the network scale and data traffic increase, the feedback and correction of the route also grow gradually. However, as the other three routing protocols cannot track the network state online, so their PDR are all lower than RHR.

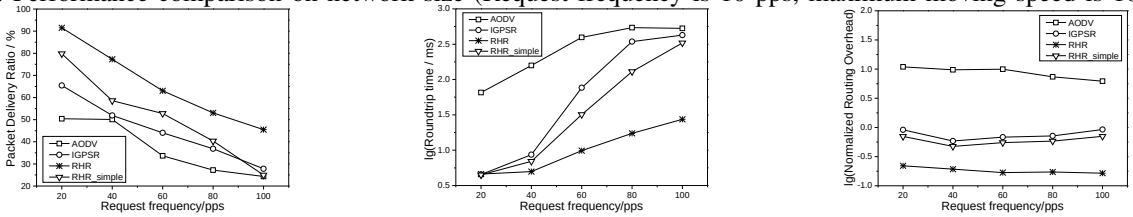
Figure 6(b) shows that the RTT of AODV is much higher than the others with the expansion of network size. This is because AODV is a topology based routing protocol. If the communication link fails, it will temporarily store the data packet in the buffer and send routing detection packet to repair the route. Not until the link is available again will those packets be sent out. Moreover, the topology of the VANET changes more frequently as the network size grows, which will lead to more link failures. It's worth mentioning that even though the differences of RTT among the other routing protocols are tiny, we can still see that RHR has the shortest delay because it selects the optimal route each time through Q-Learning.

Figure 6(c) shows that the overhead of AODV remains the highest and keeps rising as the network scale increases,



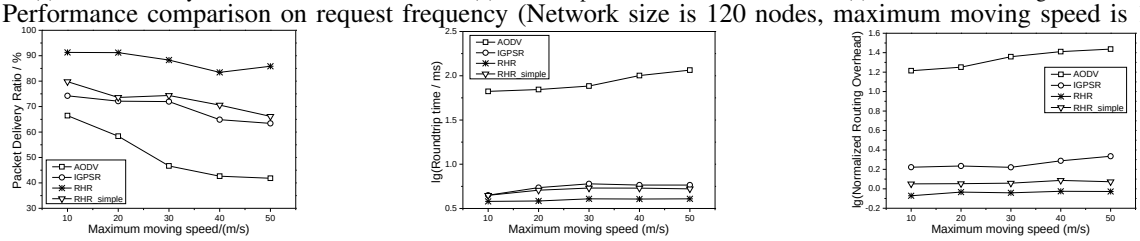

(a) Packet delivery ratio/% (b) Roundtrip time/ms (c) Normalized routing overhead

Fig. 6: Performance comparison on network size (Request frequency is 10 pps, maximum moving speed is 10 m/s).




(a) Packet delivery ratio/% (b) Roundtrip time/ms (c) Normalized routing overhead

Fig. 7: Performance comparison on request frequency (Network size is 120 nodes, maximum moving speed is 10 m/s).




(a) Packet delivery ratio/% (b) Roundtrip time/ms (c) Normalized routing overhead

Fig. 8: Performance comparison on moving speed (Network size is 120 nodes, request frequency is 10 pps).

because it needs more protocol packets to repair links frequently. IGPSR, RHR-simple and RHR have less overhead because they only use $Beacon$ as the protocol packet. RHR-simple can reduce a part of computation overhead through forwarding table, so its NRO is slightly smaller than IGPSR. Benefiting from the adaptive $Beacon$ strategy and broadcast control, RHR has the smallest overhead.

*2) Results on Request Frequency:* Figure 7(a) shows that as the request frequency increases, the communication pressure continues to rise, so the PDRs of all four protocols decline. However, because RHR adopts the Q-Learning, with packets increasing, more information is available for routing optimization, so it can still maintain the highest PDR. RHR-simple combines lookup and calculation to avoid some drawbacks of the single routing method, so its PDR is higher than IGPSR and AODV. Since IGPSR needs to calculate the routing for every packet, the computation overhead increases rapidly, and thus its PDR decreases gradually. AODV has the lowest PDR because increasing data traffic forces it to send more protocol packets to find the path, leading to more congestion.

As shown in Figure 7(b), AODV has the largest delay because it needs to resend the cached packets when the link is re-established. RHR-simple avoids per-packet calculation like IGPSR through hybrid routing, so the delay is relatively small. As the communication pressure increases, the computation overhead gradually grows. Hence, the delay of RHR-simple and IGPSR also increases. In the whole process, RHR always maintains the lowest delay due to the conditional learning strategy. When the request frequency reaches 100 pps, the RHR's RTT is much less than that of the other three protocols the gap even reaches up to 500 ms!

Figure 7(c) shows that AODV has a relatively large overhead because it has the maximum number of protocol packets. Both RHR-simple and IGPSR only use $Beacon$ packets to maintain topology information. And RHR-simple has a higher PDR than IGPSR, so it can transmit more data packets via employing relatively fewer protocol packets. Thus, its overhead is smaller than IGPSR. RHR has the minimal NRO due to the best PDR and minimum number of protocol packets.

*3) Results on Maximum Moving Speed:* It can be seen from Figures 8(a) and 8(b) that as the maximum moving speed of vehicles increases, the performance of RHR is stable, with the highest PDR and the lowest RTT in each scenario. However, the other three protocols' PDRs have different degrees of decline, and the RTTs are relatively large. This is because the dynamic of the network topology is also increasing with the speed of vehicle nodes. Then, the probability of the $blind\ path$ problem in routing protocols will rise when the topology changes drastically, resulting in packet loss. Beyond that, the route re-establishment mechanism of AODV leads to higher latency. Moreover, the number of failed links of IGPSR increases with higher motion speed of vehicle nodes, which reduces its performance, resulting in lower PDR and higher RTT than RHR_simple.

As to the overhead, RHR has the highest PDR because of its effective control of the number of $Beacons$. Figure 8(c) shows that, RHR always maintains the minimal NRO in each scenario.

*4) Results on Broadcast Control & Computation Overhead:* As shown in Figures 9 and 10, by applying our optimization

strategies, the number of broadcast packets and the calculation times in the network are greatly reduced.

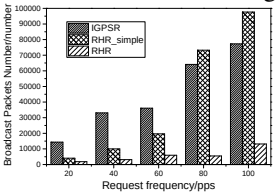



Fig. 9: The number of broadcast packets

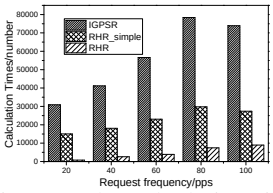



Fig. 10: The calculation times in routing process

## V. Related Work

The highly dynamic topology is the major cause of high latency as well as packet loss for VANET and makes the design of efficient routing protocols more challenging. Previous researchers have proposed various kinds of routing protocols to solve this problem. With the development of artificial intelligence, it attracts more attention to use the idea of reinforcement learning to optimize the routing process.

### A. Protocols of VANET

According to our survey, the routing protocol of VANET can be mainly divided into three categories: topology based, position based and hybrid routing protocols. AODV [4] and DSR [14] are both topology based routing protocols which need to establish the end-to-end path when a node requires sending data packets. But this kind of approach is difficult to achieve in the highly dynamic VANET. GPSR [3] is a typical position based routing protocol in which each node in network knows its own geographical location. As an extension of GPSR, [12] proposed IGPSR to set the priority for each next hop according to the information of vehicle nodes, and then it will select the node with the highest priority as next hop. However, per-packet routing computation may lead to large overhead. In order to cope with the various scenarios in VANET, some combined approaches have emerged. GEOADV [1] which combines both geographic and reactive routing method establishes the route by unicasting RREQ/RREP through locations of vehicle nodes. Beyond that, [2] proposed a new location-based hybrid routing protocol to particularly address the communication link broken issue between vehicles by efficiently using all the location information available.

### B. Q-Learning usage in routing

Paper [15] first proposed to use the Q-Learning algorithm in routing process. It utilizes a routing learning policy which keeps the balance between minimizing the number of hops a packet will take and the possibility of congestion along popular routes. Based on this research, [16] used the Dual Reinforcement Q-Routing algorithm to select the route that takes the shortest time by receiving packets from neighbors. In VANET, for overcoming the shortcomings of AODV, paper [17] used the Q-Learning algorithm to infer network state information and used unicast control packets to check the path availability to set up the dynamic route switching mechanism. Moreover, [18] proposed the PFQ-AODV algorithm that combines Q-Learning and AODV. It used the protocol packet to learn and used fuzzy logic to evaluate the quality of wireless link.

However, all previous methods do not take the $blind\ path$ problem into account. In our design, RHR not only combines both look-up and calculation but also uses Q-Learning and two-level forwarding table to optimize the routing process, effectively avoiding $blind\ path$, which performs better on packet delivery ratio, delay and overhead.

## VI. Conclusion

In order to solve the $blind\ path$ problem in existing VANET protocols, we designed and simulated a light-weight routing algorithm RHR based on reinforcement learning which combines calculating-route and lookup-route. In addition, we raised broadcast control and obstacle-avoiding strategies to optimize the performance. Finally we compared it with other representative routing protocols in VANET through simulation experiments in various scenarios. Experimental results show that RHR outperforms other approaches in PDR, RTT and NRO.


## Acknowledgements

This paper is supported by NSFC (68172213, 61432009) and Huawei Innovation Research Program (HIRP).


## References


[1] J. Skiles and I. Mahgoub, "A geographical hybrid solution for inter-vehicular communication in VANET," in *IWCMC, 2016 International*. IEEE, 2016, pp. 250–255.

[2] M. Al-Rabayah and R. Malaney, "A new scalable hybrid routing protocol for VANETs," *IEEE transactions on vehicular technology*, vol. 61, no. 6, pp. 2625–2635, 2012.

[3] B. Karp and H.-T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of ACM MOBICOM*, 2000.

[4] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," Tech. Rep., 2003.

[5] G. Carneiro, "NS-3," in *UTM Lab Meeting April*, vol. 20, 2010.

[6] W. D. Smart and L. P. Kaelbling, "Effective reinforcement learning for mobile robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation*, vol. 4. IEEE, 2002, pp. 3404–3410.

[7] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[8] Arthur Juliani, "Simple Reinforcement Learning with Tensorflow Part 0: Q-Learning with Tables and Neural Networks," 2016.

[9] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *ACM MOBICOM*, 2002, pp. 194–205.

[10] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless networks*, vol. 8, no. 2-3, pp. 153–167, 2002.

[11] Y. Li, Z. Wang, C. Zhang, H. Dai, W. Xu, X. Ji, and B. Liu, "Trajectory prediction algorithm for VANET routing," *Journal of Computer Research and Development*, vol. 54, no. 11, pp. 2421–2433, 2017.

[12] L. Hu, Z. Ding, and H. Shi, "An improved gpsr routing strategy in VANET," in *WiCOM, 2012 8th International Conference on*, 2012, pp. 1–4.

[13] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo–simulation of urban mobility," in *SIMUL 2011*, vol. 42, 2011.

[14] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile computing*, 1996, pp. 153–181.

[15] J. A. Boyan and M. L. Littman, "Packet routing in dynamically changing networks: A reinforcement learning approach," in *Advances in neural information processing systems*, 1994, pp. 671–678.

[16] S. Kumar and R. Miikkulainen, "Dual reinforcement Q-routing: An on-line adaptive routing algorithm," in *Proceedings of the artificial neural networks in engineering Conference*, 1997, pp. 231–238.

[17] C. Wu, K. Kumekawa, and T. Kato, "Distributed reinforcement learning approach for VANET," *IEICE transactions on communications*, vol. 93, no. 6, pp. 1431–1442, 2010.

[18] C. Wu, S. Ohzahata, and T. Kato, "Flexible, portable, and practicable solution for routing in VANETs: a fuzzy constraint Q-learning approach," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4251–4263, 2013.