QGrid: Q-learning Based Routing Protocol for Vehicular Ad Hoc Networks

Ruiling Li* Fan Li* Xin Li* Yu Wang†
* School of Computer Science, Beijing Institute of Technology, Beijing, 100081, China.
† Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC 28223, USA.

***Abstract*—In Vehicular Ad Hoc Networks (VANETs), moving vehicles are considered as mobile nodes in the network and they are connected to each other via wireless links when they are within the communication radius of each other. Efficient message delivery in VANETs is still a very challenging research issue. In this paper, a Q-learning based routing protocol (i.e., QGrid) is introduced to help to improve the message delivery from mobile vehicles to a specific location. QGrid considers both macroscopic and microscopic aspects when making the routing decision, while the traditional routing methods focus on computing meeting information between different vehicles. QGrid divides the region into different grids. The macroscopic aspect determines the optimal next-hop grid and the microscopic aspect determines the specific vehicle in the optimal next-hop grid to be selected as next-hop vehicle. QGrid computes the Q-values of different movements between neighboring grids for a given destination via Q-learning. Each vehicle stores Q-value table learned offline, then selects optimal next-hop grid by querying Q-value table. Inside the selected next-hop grid, we either greedily select the nearest neighboring vehicle to the destination or select the neighboring vehicle with highest probability of moving to the optimal next-hop grid predicted by the two-order Markov chain. The performance of QGrid is evaluated by using real life trajectory GPS data of Shanghai taxies. Simulation comparison among QGrid and other existing position-based routing protocols confirms the advantages of proposed QGrid routing protocol for VANETs.**

I. Introduction

Vehicular Ad Hoc Network (VANET) is a high-speed mobile wireless network containing a set of smart vehicles which could communicate with each other via wireless medium. Different from the traditional mobile ad hoc network, VANET has its unique characteristics which pose new challenges in the design of networking protocols and capacity analysis [1] [2], especially for routing protocols. High speed vehicles lead to highly dynamic and more easily disconnected network topology. Routing protocols in VANETs are classified into unicast, broadcast, multicast and geocast. QGrid routing proposed in this paper falls into both unicast and geocast categories. It delivers message to a specific geographic region and does not allow multiple copies of a message during the transmission.

The work of F. Li is partially supported by the National Natural Science Foundation of China under Grant No. 61370192, 61432015 and 60903151, and the Beijing Natural Science Foundation under Grant No. 4122070. The work of X. Li is partially supported by the National Natural Science Foundation of China under Grant No. 61300178 and National Program on Key Basic Research Project under Grant No. 2013CB329605. This work of Yu Wang is supported in part by the US National Science Foundation under Grant No. CNS-1319915 and CNS-1343355, and the National Natural Science Foundation of China under Grant No.61428203. F. Li is the corresponding author.

Unicast routing protocols in VANETs can be categorized into position-based and topology-based routing. The usage of digital maps, GPS equipments, and navigation system in modern vehicles inspires the study of position-based routing for VANETs. In this paper we design a new position-based routing protocol in unicast applications for VANETs. We assume that vehicles are equipped with GPS devices in order to obtain their current locations. Many position-based routing protocols in vehicular networks have been proposed in recent years [3]–[10]. For example, GPSR [7] makes greedy forwarding decisions only by using neighbor's information. It always looks for the nearest neighboring node to the destination as the next-hop node. However, due to the high speed of mobile vehicular node, the planarizartion of network graph becomes difficult. GPCR [8] realizes that the map of the city streets is planar naturally, so the planarization could be eliminated completely. Another position-based directional routing method, PDVR [9], selects the next-hop vehicle according to the vehicle driving direction. VANET is a special kind of Delay Tolerant Network (DTN). [11] is a novel contact prediction based routing scheme for DTNs. QGrid proposed in this paper is a unicast routing protocol based on geographic location information.

The forwarding strategy is the key component of vehicular routing. Many routing strategies based on the historical encounter information. The exploiting temporal dependency method [12] delivers message by mining the temporal correlation and the evolution of Inter Contact Times (ICTs) between each pair of vehicles by using the historical real life trajectory data. However, the temporal dependency information of nodes is based on the low-level movement information. Zoom [13] is an opportunistic forwarding algorithm which captures both microscopic mobility of pairwise contacts and macroscopic mobility of social relationships within VANETs. Current vehicle forwards message depending on the estimated delay between the neighboring vehicle and destination node. The vehicle which has the shortest estimated delay to the destination is selected as the next-hop node.

Our proposed QGrid considers both macroscopic aspect and microscopic aspect when making its routing decision which is similar to GeoMob [14]. GeoMob discretizes the map to sells and applies k-means clustering algorithm to cluster these cells to different regions. Different from GeoMob, which divides the city into regions by using clustering algorithm, our scheme

978-1-4799-7575-4/14/$31.00 ©2014 IEEE

QGrid divides city into different grids. QGrid computes the Q-values of different movements between neighboring grids for a given destination via Q-learning. Each vehicle stores Q-value table learned offline, then delivers message to the neighboring grid by querying Q-value table. Inside the grid, we either greedily select the nearest neighboring vehicle to the destination or select the neighboring vehicle with highest probability of moving to the optimal next-hop grid predicted by the two-order Markov chain. Thus, we name our proposed Q-learning and grid based routing protocol as *QGrid*.

The remainder of the paper is organized as follows. Section II provides a brief review of existing grid and Q-learning based routing protocols. Section III introduces our proposed QGrid routing protocol in details. Section IV presents simulation results over different position based methods and Section V concludes the paper.

II. Related Work

Grid-based routing is one of the traditional routing protocols based on geographic location. Our target is to deliver message from a mobile vehicle to a fixed destination with high probability. GVGrid [15] is a QoS routing protocol designed for VANETs. It assumes that every vehicle has a digital map of the city and knows its geographical position and direction through GPS equipment or other localization methods. Using this information, GVGrid selects a route by vehicles which are likely to move in similar speeds and toward similar directions. HarpiaGrid [16] is also a grid based geographic routing algorithm, which uses digital map to generate a shortest transmission grid route. Message transfer was restricted in the grid sequences. Our proposed QGrid divides the geographical area into uniform-size squares, called grids as GVGrid/HarpiaGrid does. Compared with GVGrid and HarpiaGrid, QGrid does not need digital map. We just only assume that every vehicle knows its geographical position from the GPS equipment or other localization methods.

Q-learning [17] [18] is a form of model-free reinforcement learning. The most prominent advantages of Q-learning is that it can acquire optimal control strategies from delay rewards, even when the agent has no prior knowledge of its actions on the environment. Therefore, it is widely used in robot, games, process control and scheduling, etc. In recent years, it has been gradually used in VANET. For example, QLAODV [19] is an improvement of AODV protocol which is based on Q-learning. It chooses a route by considering vehicle movement and available channel bandwidth. QLAODV is a distributed reinforcement learning routing protocol, which uses Q-Learning algorithm to infer VANET environment states information and uses unicast control packets to check the path availability in a real time manner. But the convergence speed of Q-learning algorithm is slow, especially when the number of learning states is very large. A fuzzy constraint Q-learning routing protocol for vehicular networks was proposed in [20], which is the improvement of QLAODV. It employs a fuzzy logic to evaluate link and uses Q-learning based approach to select a route which is more stable and efficient. In addition to application in VANET routing, Q-learning was used in the field of data aggregation and data collection. A delay control scheme named Catch-up [21] was proposed which is a distributed learning algorithm based on Q-learning. Catch-up adaptively chooses forwarding delays to make nearby reports have a better chance to meet each other.

Our proposed protocol is different from the protocols mentioned above in the following three ways. Firstly, we do not take vehicles as environment states in Q-learning; on the contrary, we regard the different grids as environment states. The biggest benefit of this approach is that it greatly reduces the learning states and increases the convergence speed of the method. In addition, learning is not affected by the change of number of vehicles in the network. Secondly, each vehicle greedily selects the optimal next-hop grid based on the Q-value table. The vehicle does not need to maintain the routing table, it can determine the optimal next-hop grid by querying the Q-value table no matter which grid it is in. Finally, the Q-value table is the long time learning result of the agent about the environment. Q-learning is a continuously iterative process; therefore action selection is based on a long-term benefit consideration, not just depending on the immediate benefits. The distribution of vehicles in different grid regions is relatively stable; therefore we can study the Q-value table offline. Our protocol can be divided into two perspectives, the macro is to find the optimal next-hop grid and the micro is to find the specific next-hop vehicle inside the selected grid.

III. QGrid: Q-learning and Grid Based Routing

In this section, we introduce our Q-learning based routing protocol for vehicular networks in details. Q-learning algorithm can acquire optimal control strategies from delayed rewards, even when the agent has no prior knowledge about the effect of its actions on the environment. Reinforcement learning algorithms are frequently used to solve optimization problems. In general, the reinforcement learning model consists of four parts.

- ***S*:** Discrete set of environment states.
- ***A*:** Discrete set of agent actions.
- $f_R(s_t, a_t)$**:** Reward function. The immediate reward received after taking action a_t in state s_t at time t, $s_t \in S$, $a_t \in A$.
- $f_S(s_t, a_t)$**:** Transfer function. State changes from s_t to another state when taking action a_t at time t, $s_t \in S$, $a_t \in A$.

Agent constantly interacts with the environment in order to learn control strategy. Agent observes that the current environment status is s_t, and then selects an action a_t. After taking the action a_t, agent gains the reward value $r_t = f_R(s_t, a_t)$, and system state changes to $s_{t+1} = f_S(s_t, a_t)$.

A. Q-learning Algorithm and Data Feature

However, in VANETs, there is no way to determine the reward until the message is delivered to the destination. Hence using the model-based approach is impossible. Therefore, we adopt Q-learning method which does not need a model of the

environment. $Q(s_t, a_t)$ is a real value corresponding to state-action pairs, called Q-value. Usually, in Q-learning algorithm there will be a learning factor α to describe the learning rate. It governs how quickly the Q-values can change with changing of the environment. In this paper, the learning factor is an empirical value as same as in QLAODV [19]. Let a' represent the next action corresponding to the next state. The core idea of Q-learning algorithm is to update Q-value constantly using Equation (1) [17] until convergence is achieved.

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t) + \alpha(f_R(s_t, a_t) + \gamma \max_{a'} Q(f_S(s_t, a_t), a')) \quad (1)$$

Discount factor γ is a very important parameter for the Q-learning algorithm. It is a constant value in the traditional Q-learning algorithm. But in order to better distinguish the pros and cons of different grids, we set γ to a dynamic parameter between different grids movement action. We define the value of γ using a piecewise function based on the number of vehicles in different girds. The goal of QGrid is to make the selected vehicles along the grids with high vehicle density. When the status and actions are discrete, reward function is bounded, and the pairs of (s, a) can be accessed with infinite times. Q-learning algorithm can converge to the Q function [17].

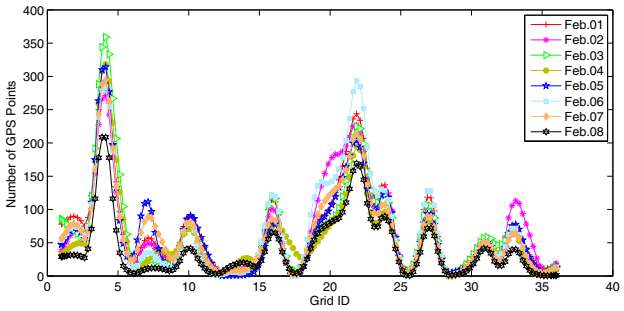


Fig. 1. Number of GPS points in each grid for different days.

Shanghai taxies data set [22] was collected from February 1 to March 3, 2007. Taxies periodically send GPS reports back to data center. The specific reports information includes: ID, longitude, latitude, timestamp, moving speed, heading direction and status. The study found that the vehicle's historical trajectory data have very strong regularity. Fig. 1 represents the GPS number variation in different grids from February 1 to February 8, 2007. The experiment area is about $1200m \times 1200m$ around Shanghai railway station, and the length of the grid is $200m$. The x axis represents grid ID, and the y axis represents the numbers of GPS points of each grid. As shown in the figure, the numbers of GPS points in each grid of different days are relatively stable, thus we can learn Q-value table from historical trajectory data offline and transfer message depend on them. Online convergence speed of Q-learning algorithm is slow especially when there are lots of states which reflect the advantage of offline learning. We think that the more vehicles in the grid, the more GPS point

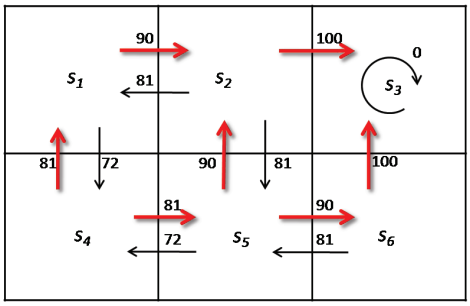


Fig. 2. A Q-learning example. s_4 is the source grid, and s_3 is the destination grid. The value above the arrow is the Q-value of this moving action. Red arrows demonstrate three possible paths from s_4 to s_3 based on Q-values.

records in the grid, because the GPS records information are uploaded depending on certain granularity.

In QGrid routing protocol, we divide the geographical area into uniform-size squares called *grids*. Each vehicle stores Q-value table of the area learned offline. QGrid is composed of two parts, which determines the optimal next-hop grid from the macroscopic aspect (offline) and identifies a specific vehicle in the selected grid from microscopic aspect (online).

B. Q-learning and Grid Based Routing

The aim of QGrid routing protocol is to send message from a grid in which the source vehicle is located to another fixed location with high probability. This aim has lots of similarities with the classic example in Q-learning algorithm displayed in Fig. 2, which is a simple case to illustrate the basic concept of Q-learning. In the figure, each grid square s represents a distinct state in S, and each arrow represents a distinct action moving between neighboring grids. The agent can move towards Up, Down, Left, and Right. $f_R(s_t, a_t)$ gives 100 points reward for actions entering the goal state, and zero points for other actions. In Fig. 2, s_3 is the destination grid, discount factor $\gamma = 0.9$, learning factor $\alpha = 1$. After a period of learning, Q-value table is computed. Q-value table includes Q-values of possible moving directions of the agent for each grid, which is presented by the values above the arrows in the figure. Thus vehicle can forward message by querying the Q-value table no matter which state it is in. It follows the direction which has the largest Q-values among four possible directions. For example, vehicles in s_4 will send message along the path $s_4 \rightarrow s_5 \rightarrow s_6 \rightarrow s_3$, $s_4 \rightarrow s_5 \rightarrow s_2 \rightarrow s_3$, or $s_4 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3$.

The difficulty we are facing is that how to use reinforcement learning model to characterize the vehicular networks. In QGrid, we define the different grids as state S. In this way, the number of learning states will be greatly reduced and Q-value table learned offline can be used during the whole lifetime. The entire vehicular ad hoc network is the environment. We assume that there is a virtual agent, and action each agent can take is to send packet from one grid to another grid. The agent infers the environment from the reward R. Reward R is used to represent whether the packet is delivered to destination grid or not. We assume the destination of the message is fixed during

the process of message delivery. When the destination is in the neighbor grid, the reward R will be 100 and in other cases, R will be 0. The discount factor γ is a dynamic parameter depending on the number of vehicles in the grid. It's easy to understand that the grid which vehicles often across is more likely to transmit message. Our purpose is to choose most reliable link to transfer message. We use a piecewise function to characterize the change of the discount factor. Let $num(s_i)$ denote the number of GPS records in grid s_i. The discount factor of QGrid is related with the number of GPS records in different grids. Therefore, different grids have different values of γ. Let $\overline{num}(s) = \frac{1}{n}\sum_{k=1}^{n} num(s_k)$, in which n represents the total number of grids. γ is defined as below:

$$\gamma = \begin{cases} min\{0.9, \beta \cdot \frac{num(s_i)}{\overline{num}(s)}\} & \text{if } num(s_i) \geq \overline{num}(s) \\ max\{0.3, \beta \cdot \frac{num(s_i)}{\overline{num}(s)}\} & \text{if } num(s_i) < \overline{num}(s) \end{cases}$$

This formula maps the value of γ to the range from 0.3 to 0.9 based on the different vehicle density in different grids. On the one hand we hope to distinguish different grids better (i.e., when $0 \leq \gamma \leq 1$), on the other hand we do not want the Q-value gained from the neighboring grids to be too large (i.e., when $\gamma = 1$), which means Q-value is heavily influenced by the maximum Q-value of the neighboring grid. At the same time, we do not want the Q-value to be too small (i.e., when $\gamma = 0$), which means the Q-values of neighboring grid has no effect on the Q-value calculation of current grid. Therefore we take a trade off to let $\gamma \in [0.3, 0.9]$. β is set to 0.6, because if the number of the GPS records in grid equals to average GPS records of all grids, we want the value of γ to be 0.6. Here we only use discount factor γ to reflect the influence of different vehicle density in other grids. At the start of communication, agent knows nothing about the entire environment. All elements of Q-value table are initialized to be zero. We utilize Q-learning method to gain the Q-value table offline and the Q-value table is pre-stored in each vehicle, then the message is transmitted based on the Q-value table.

In QGrid, vehicles do not need to maintain the routing table, they transmit message only depending on Q-value table learned offline, i.e., current vehicle selects next-hop grid who has the maximum Q-value as the target grid and select one vehicle in the target grid using *Vehicle Selection Strategy* which will be discussed in Section III-C. Q-value table contains the optimal next-hop grid corresponding with different destination grids. Because the discount factor γ in QGrid is related to the density, therefore QGrid may be regarded as routing algorithm based on density. But the difference between QGrid and traditional routing algorithm based on density is that QGrid considers macroscopic aspect and microscopic aspect when making the routing decision. The macroscopic aspect determines the optimal next-hop grid based on Q-values learned offline, and the microscopic aspect locally determines the specific vehicle in the optimal grid to be selected as next-hop vehicle online. Thus QGrid take the advantages of both online and offline methods.

Let v_i represent current node, s_j represent the next-hop grid, $member(s_j)$ represent the set of vehicles in grid s_j, $neighbor(v_i)$ represent the set of neighboring vehicles of v_i, $dist(v_i, v_j)$ represent the distance between v_i and v_j. There are three cases when choosing the next-hop vehicle:

1) If there exist v_i's neighboring vehicles in the optimal next-hop grid, current vehicle will forward message to the selected next-hop vehicle in the neighboring vehicles based on *Vehicle Selection Strategy*. *Vehicle Selection Strategy* determines how to choose one vehicle inside the selected grid to be the next-hop vehicle, which will be discussed in Section III-C.
2) If there is no neighboring vehicle of v_i inside the next-hop grid (i.e., s_j), current vehicle v_i will select the nearest neighboring vehicle of v_i to destination v_d.
3) If current vehicle v_i cannot find a neighboring vehicle which has closer distance to the destination v_d than v_i itself, v_i will hold the message and waiting for next forwarding opportunity arrival.

The details of the QGrid algorithm is described in **Algorithm** 1:

Algorithm 1 QGrid: Q-learning and Grid based Routing Protocol

Current vehicle v_i in Grid s_i has a copy of message destinated to a specific location $\boldsymbol{D}$, which is inside Grid s_d. Every vehicle has a copy of Q-value table learned offline by using Equation (1).

```
Select next-hop Grid s_j from querying Q-value table whose destination grid is s_d and has the maximum Q-value.
if member(s_j) ⋂ neighbor(v_i) ≠ Φ then
    Select next-hop vehicle inside Grid s_j using Vehicle Selection Strategy.
else
    for all vehicle v_k ∈ neighbor(v_i) do
        if minimum dist(v_k, D) < dist(v_i, D) then
            v_i sends message to the neighbor v_k which has the minimum distance to destination location D.
        else
            v_i continues holding the message and waiting for next forwarding opportunity arrival.
        end if
    end for
end if
```

C. Vehicle Selection Strategy inside Optimal Grid

After querying Q-value table, we know which neighboring grid is the optimal next-hop grid, but inside the grid, which vehicle should be selected as the next-hop relay is also worth considering. In this section, we will discuss the strategy to select a specific vehicle, i.e., *Vehicle Selection Strategy*. Two strategies will be introduced, i.e., *Greedy selection strategy* and *Markov selection strategy*. The idea of *Greedy selection strategy* is very similar to HarpiaGrid. Current vehicle v_i with

message prefers to select the nearest neighboring vehicle v_k to the destination v_d inside the optimal next-hop grid s_j as the relay vehicle.

Markov selection strategy uses two order Markov chain to predict the vehicle's next location grid. At first, we need to extract the grid sequence of each vehicle based on the historical GPS trajectory data. Due to the large granularity of real life GPS trajectory data, the extraction may cause partial grids loss, although the data interpolation is utilized. The purpose of this step is to discretize the successive trajectory information. After discretization, action sequence of vehicles can be described as $\{s_{k_1}, s_{k_2}, \cdots, s_{k_n}\}$, which is a set of sample values of random process $\{G_i | G_i \in S\}_{i=1}^{n}$. The probability of using m order Markov chain to predict vehicle's next position grid is expressed as follows:

$$Pr(G_n = s_{k_n} | G_1 = s_{k_1}, G_2 = s_{k_2}, \cdots, G_{n-1} = s_{k_{n-1}}) = Pr(G_n = s_{k_n} | G_{n-m} = s_{k_{n-m}}, \cdots, G_{n-1} = s_{k_{n-1}})$$

It suggests that the current state situation is only related to the past m states. Our problem can be expressed as for a given set of past states of random process $\{G_j | G_j \in S\}_{j=1}^{m}$, how to predict the state of G_{m+h}.

$$Pr(G_{m+h} = s_{k_{m+h}} | G_1 = s_{k_1}, G_2 = s_{k_2}, \cdots, G_m = s_{k_m})$$

Here, m stands for the past m states and h stands for the h steps forwards from current state. In our proposed protocol, $m = 2$ and $h = 1$. In other words, we use the past two grid position to predict the next grid position. We apply two order Markov chain because higher order Markov chain leads to high computational complexity. The most important thing is that the increasing of complexity may not improve the precision of the prediction obviously. We calculate one step transition matrix of the second order Markov chain based on the historical trajectory data of the vehicles. The current vehicle v_i in Grid s_i carries message M to destination Grid s_d. The optimal next-hop grid of s_i is s_j which is determined by Q-value table. There are two cases when choosing optimal next-hop vehicle.

1) If there is only one vehicle in the communication radius of v_i in Grid s_j. There is no doubt that we will choose this vehicle as the relay.
2) If there are more than one vehicles (i.e., q $(q > 1)$) in the communication radius of v_i inside Grid s_j. Assume $v_j \in (member(s_j) \bigcap neighbor(v_i))$, $j = 1, 2, \cdots q$. We will choose relay vehicle based on transition matrix. The optimal next-hop grid selected is s_j, then we find the optimal next-hop grid of s_j is s_l depending on Q-value table again. Assume the previous grid of v_j before s_j is s_p, $j = 1, 2, \cdots q$. We will select the vehicle which has the highest probability among $Pr_{v_j}(s_l | s_j s_p)$ $(j = 1, 2, \cdots q)$ as the optimal next-hop vehicle.

This *Markov selection strategy* is described in **Algorithm 2**.

Vehicle's movement is limited by many conditions, such as roads, light, obstacle, and the other vehicles on the road, but there are not so many restrictions for message transferring. Vehicles must move along the road, but the message does not need. In our proposed QGrid, vehicles can transfer message to eight directions of neighboring grids, as shown in the Fig. 3. The blue circle represents the communication radius. No matter which grid the vehicle is in, it can always find the optimal next-hop grid by querying the Q-value table. The size of Q-value table is related to the number of grids. The larger grids number is, the more items in Q-value table, and vice versa.

It worths to mention that vehicles are considered as the learning states in QLAODV and data aggregation delay control scheme Catch-up. However, constantly moving vehicles in the VANETs lead to the fast change of the learning status, which is a big disadvantage of the routing protocol applied in highly dynamic network topology of VANETs. Our proposed QGrid algorithm consider grids as learning states, and only needs to learn historical data periodically offline when choosing the next-hop grid, which saves a lot of system resources and time.

Algorithm 2 Markov Selection Strategy

1: **if** $member(s_j) \bigcap neighbor(v_i)$ has only one element **then**
2: v_i forwards M to this vehicle
3: **else**
4: Determine the next-hop grid s_l of current Grid s_j by querying Q-value table.
5: **for all** $v_j \in (member(s_j) \bigcap neighbor(v_i))$ **do**
6: Assume the previous grid of v_j before Grid s_j is Grid s_p.
7: Use two order Markov chain to calculate conditional probability $Pr_{v_j}(s_l | s_j s_p)$.
8: **end for**
9: Select the vehicle with the maximum conditional probability $Pr_{v_j}(s_l | s_j s_p)$ as the next-hop vehicle.
10: **end if**

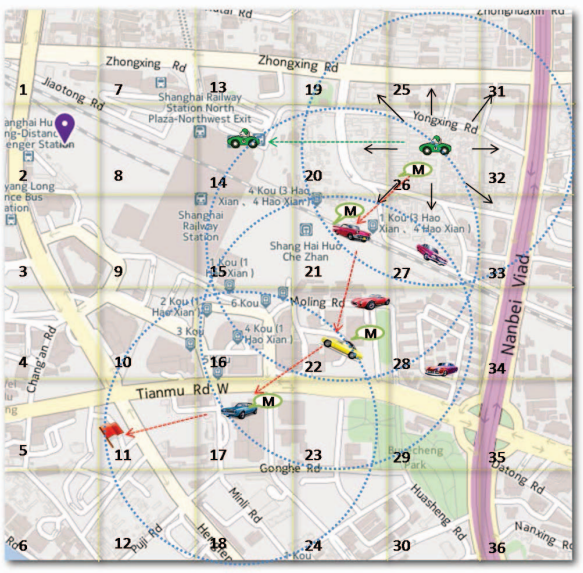


Fig. 3. An example of message transferring of QGrid routing protocol.

Inside the next-hop grid, QGrid locally determines the next-hop vehicle. Thus, QGrid take the advantages of both online and offline methods.

IV. Simulations

In this section, we conduct simulations with real life taxi trajectory data to evaluate our proposed protocol and compare it with other existing traditional position-based routing protocol.

A. Compared Routing Protocol

In the simulation, we name QGrid using *Greedy selection strategy* or *Markov selection strategy* to determine the next-hop vehicle inside next-hop grid as **QGrid_G** and **QGrid_M**, respectively. We compare the performance of QGrid_G and QGrid_M with other existing position based routing protocols listed as below.

- **GPSR**: GPSR is a geographic based routing algorithm, which establishes routing with greedy policy. When v_i needs to send message to v_d, v_i selects the nearest neighboring node to v_d as the next-hop node, then v_i sends packet to it.
- **HarpiaGrid**: HarpiaGrid assumes each vehicle in VANET has a digital map, and vehicle sends message by discovering the shortest route.

B. Routing Metrics

In all experiments, we compare each algorithm using the following routing metrics.

- **Delivery ratio:** The average percentage of successfully delivered messages from the sources to the destinations.
- **Hop count:** The average number of hops during each successful delivery from the sources to the destinations.
- **Delay:** The average time duration of successfully delivered messages from the sources to the destinations.
- **Number of forwarding:** The average number of message forwardings in the network during the whole simulation period.

C. Data Processing

Shanghai taxies data set has introduced in III-A. Due to the communication cost for data transmission, reports are sent at a time granularity. The granularity of reports is one minute for taxies with passengers and about 15 seconds for empty. Considering that Spring Festival of China and Valentine's Day are in the February of 2007, therefore, we choose trajectories from February 1 to February 8 as the training set and February 9 as the test set. The area size we selected is about $1200m \times 1200m$ around railway station in Shanghai, as shown in Fig. 3.

Real trajectory data is very sparse, and it usually needs interpolation. In Fig. 4, the green points represent the real life trajectories of one taxi whose ID is 18290 before interpolation. Interpolation is an important research direction in VANETs; it also involves the map matching problem. In our paper, we pay more attention to the design of the routing algorithm. Therefore, we use the simple method Dijkstra algorithm for data interpolation. The point on the path will be inserted into the vehicle's GPS trajectory. The timestamp was added to the new points by using average speed. The pink points represent the new inserted GPS data after interpolation. You can see that the vehicle's trajectory data increases clearly. Removing duplicate data and error status data is necessary in Shanghai taxies data set.

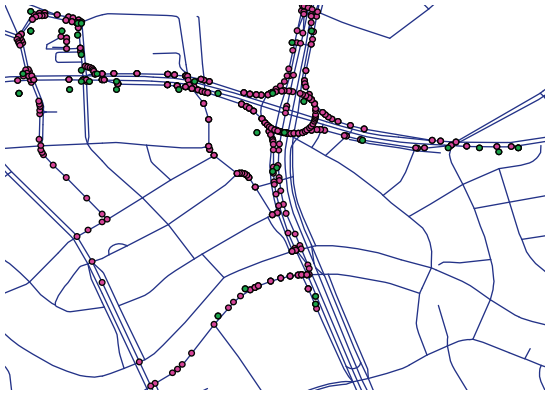


Fig. 4. The trajectories of one taxi with ID 18290 after interpolation within one hour. The green/pink points in the figure represent trajectory before/after interpolation.

Different vehicles upload their GPS data at different time. Even though they are inside each other's communication range at same time, they may not upload their GPS data at the same moment, which leads to actual neighboring vehicles disconnected in the GPS data set. Thus we use time slot to represent the real life GPS data uploaded time. If two vehicles are inside each other's communication range during this period of time, we consider these two vehicles meet during this period. The selection of the time slot affects the network topology dramatically, thus influence the routing performance. We use $\triangle T$ to represent time slot, and the time field will be replaced by $t = time/\triangle T$.

QGrid routing protocol consists of two steps, the first step is to determine the optimal next-hop grid, and the second step is to determine the specific vehicle in the optimal grid to be selected as the next-hop vehicle. In the simulation, the length of the grid is $2 \times radius/\sqrt{10}$ as same as in HarpiaGrid. The length of the grid is very important. On the one hand, if the length of grid is too small, the number of the status (i.e., the number of grids) to study will increase sharply, which not only leads to slow convergence speed but also increases hop count to the destination. Besides, it will be very difficult to find the candidate vehicles in the next-hop grid. On the other hand, if the length of grid is too large, although convergence is faster, the neighboring vehicles may be in the same grid with the vehicle who carries the message. It is easy to determine the optimal next-hop grid by querying Q-value table, but it is most likely that there is no neighboring vehicle in the optimal next-hop grid because of the limitation of communication radius. The communication radius is set to $100m$. We do not want that the destination location is in the communication radius of the source vehicle who generates the

TABLE I
PARAMETERS USED IN SIMULATIONS

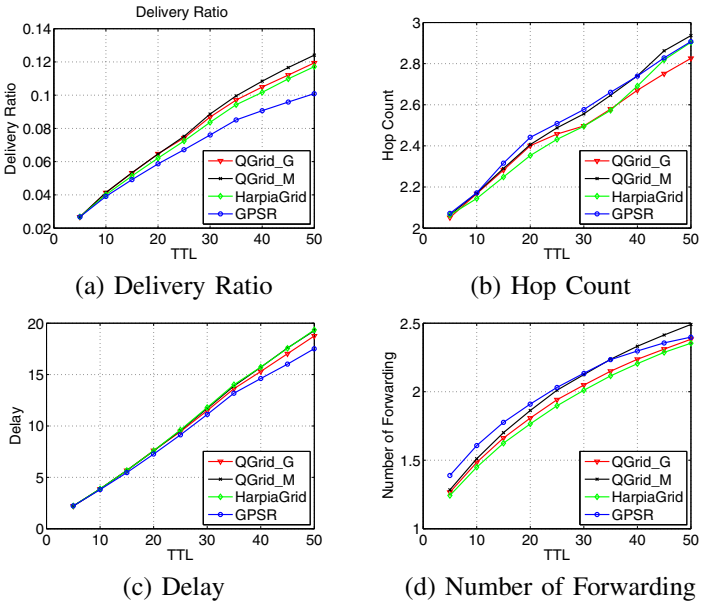
Parameter	Value or Range
α	0.8
β	0.6
γ	$[0.3, 0.9]$
experiment area	$1200m \times 1200m$
$dist(v_s, \boldsymbol{D})$	$\geq 600m$
message generating rate for each vehicle	10 / second
communication radius	100m
grid length	$2 \times radius/\sqrt{10}$
reward R	$0, 100$
the experimental data date	Feb 1, 2007 to Feb 8, 2007
the experimental data time	Feb 9, 2007
time slot ΔT	$1s, 20s, 30s$
TTL	$10, 20, 30, 40$

message. Therefore, the minimum distance between the source vehicle and destination location is set to $600m$ considering the experimental area of $1200m \times 1200m$. The source vehicle and destination location of the message is randomly generated. Source vehicle generates 10 messages every second.

D. Simulation Results

In the following simulations, we evaluate the performance of QGrid and other existing position based routing protocols using real life trace data of Shanghai taxies. The experiment parameters used are shown in table I. Because QGrid_M needs to use two order Markov chain to predict the next possible grid, if the time slot $\triangle T$ is too large, during this time slot, vehicle may move across several grids, which may result in inaccurate location prediction. Therefore, at first we compare QGrid_G and QGrid_M with other protocols when $\triangle T = 1s$. Small time slots can accurately reflect the performance of different algorithms, but at the same time the delivery ratio of simulation is lower than it really is because of the different GPS trajectory uploaded time. Therefore, we also compare QGrid_G with other algorithms when $\triangle T = 20s$ and $\triangle T = 30s$.

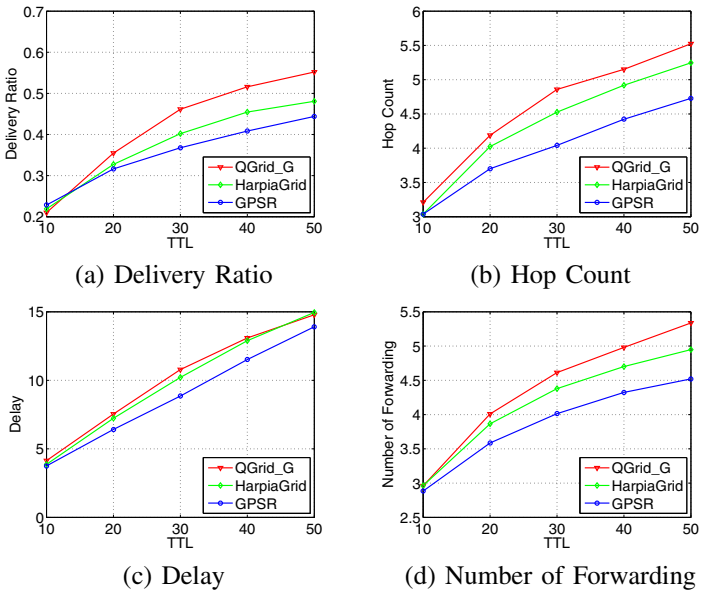
we set time slot $\triangle T = 1s$ in Fig. 5. Fig. 5(a) shows that the delivery ratio rises gradually along with the increase of TTL. That is because more messages will be successively delivered when TTL increases. It can be seen from the figure that the delivery ratio of QGrid_G and QGrid_M is higher than HarpiaGrid and GPSR, which is mainly because GPSR selects the most nearest neighboring nodes to destinations as the relay nodes, but it does not have the whole picture of the network. QGrid_G and QGrid_M select next-hop grid in the long term perspective. The delivery ratio of QGrid_M is higher than QGrid_G because it selects next-hop vehicle depending on two order Markov chain predicted based on the historical trajectory data. Fig. 5(b) shows that the hop counts of GPSR and QGrid_M are higher than QGrid_G and HarpiaGrid, but the differences are very small. Fig. 5(c) shows that the delays of QGrid_G, QGrid_M and HarpiaGrid are higher than GPSR. The reason is that GPSR always selects the most nearest neighbor nodes to destinations as the relay nodes, although the successfully transferred message number is relatively small,



(a) Delivery Ratio (b) Hop Count

(c) Delay (d) Number of Forwarding

Fig. 5. Simulation results of all different routing protocols when $\Delta = 1s$.



(a) Delivery Ratio (b) Hop Count

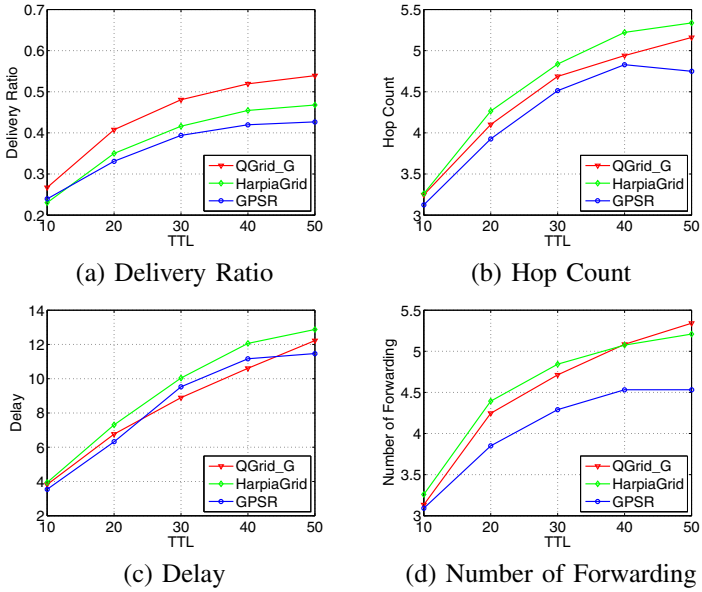
(c) Delay (d) Number of Forwarding

Fig. 6. Simulation results of all different routing protocols when $\Delta = 20s$.

this strategy has brought low latency. Fig. 5(d) shows that the number of forwardings of QGrid_G and HarpiaGrid are lower than GPSR and QGrid_M.

Due to the different GPS trajectory uploaded time, vehicles may miss a lot of neighboring nodes. when $\triangle T = 1s$, the delivery ratio is much lower than it really is. Therefore, we increase vehicle's neighbors by increasing time slot. Fig. 6 and Fig. 7 represent the simulation results when $\triangle T = 20s$ and $\triangle T = 30s$, respectively. Note that when $\triangle T$ is long, the prediction of QGrid_M by Markov chain may be inaccurate thus Qgrid_M is not included in these two sets of simulation.

From Fig. 6(a), we can clearly find that the delivery ratios increase compared with Fig. 5(a). The delivery ratios rise



(a) Delivery Ratio (b) Hop Count

(c) Delay (d) Number of Forwarding

Fig. 7. Simulation results of all different routing protocols when $\Delta = 30s$.

gradually along with the increasing of time slot $\triangle T$, that is because the number of neighboring vehicles will increase when the time slot $\triangle T$ becomes longer. Fig. 6(a) clearly demonstrates that QGrid_G has higher delivery ratio tha that of HarpiaGrid and GPSR. The cost of higher delivery ratio is more overhead, as can be seen from Fig. 6(b-d), the hop count, delay, and the number of forwardings of QGrid_G is higher than GPSR and HarpiaGrid. The reason is that QGrid_G delivers message along the possible successful path other than the shortest path, which will increase the hop count, delay, and number of forwardings. The goal of QGrid algorithm is to improve the delivery ratio in VANETs environment, therefore, the results can be accepted to a certain degree. Fig. 7 are the simulation results when $\triangle T = 30s$, which are very similar to Fig. 6.

V. Conclusion

In this paper, a routing protocol called QGrid is proposed which is based on reinforcement learning to improve the delivery ratio of message transferring. QGrid considers macroscopic aspect and microscopic aspect when making the routing decision. The macroscopic aspect determines the optimal next-hop grid by querying Q-value table learned offline. The microscopic aspect determines the specific vehicle in the optimal next-hop grid to be selected as next-hop vehicle. We either greedily select the nearest neighboring vehicle to the destination or select the neighboring vehicle with highest probability of moving to the optimal next-hop grid predicted by the two-order Markov chain. Thus, QGrid takes advantages of both offline and online methods. The performance of QGrid is evaluated by using real life trajectory GPS data of Shanghai taxies. The simulation comparison among QGrid and existing position based routing protocols confirms the effectiveness and efficiency of proposed QGrid for VANETs.

References

[1] Y. Huang, X. Guan, Z. Cai, and T. Ohtsuki, "Multicast capacity analysis for social-proximity urban bus-assisted vanets," in *Communications (ICC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6138–6142.

[2] Y. Huang, M. Chen, Z. Cai, X. Guan, , and T. OHTSUKI, "Capacity analysis for urban vehicular ad hoc networks with graph-theory based construction," in *IEEE Global Telecommunications Conference*, 2014, pp. 136–145.

[3] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *Vehicular Technology Magazine, IEEE*, vol. 2, no. 2, pp. 12–22, 2007.

[4] Y.-W. Lin, Y.-S. Chen, and S.-L. Lee, "Routing protocols in vehicular ad hoc networks a survey and future perspectives." *Journal of Information Science & Engineering*, vol. 26, no. 3, 2010.

[5] F. Li, L. Zhao, X. Fan, and Y. Wang, "Hybrid position-based and dtn forwarding for vehicular sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, 2012.

[6] L. Zhao, F. Li, and Y. Wang, "Hybrid position-based and dtn forwarding in vehicular ad hoc networks," in *Vehicular Technology Conference (VTC Fall), 2012 IEEE*. IEEE, 2012, pp. 1–5.

[7] B. Karp and H.-T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM, 2000, pp. 243–254.

[8] C. Lochert, M. Mauve, H. Füßler, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 1, pp. 69–72, 2005.

[9] D. Tian, K. Shafiee, and V. C. Leung, "Position-based directional vehicular routing," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*. IEEE, 2009, pp. 1–6.

[10] Y. Huang, M. Chen, Z. Cai, X. Guan, and T. OHTSUKI, "Intersection-based forwarding protocol for vehicular ad hoc networks," *Telecommunication Systems Journal*, p. In Press, 2014.

[11] L. Zhang, X. Wang, J. Lu, M. Ren, Z. Duan, and Z. Cai, "A novel contact prediction based routing scheme for dtns," *Transactions on Emerging Telecommunications Technologies*, p. In Press, 2014.

[12] H. Zhu, S. Chang, M. Li, K. Naik, and S. Shen, "Exploiting temporal dependency for opportunistic forwarding in urban vehicular networks," in *INFOCOM, 2011 Proceedings IEEE*. IEEE, 2011, pp. 2192–2200.

[13] H. Zhu, M. Dong, S. Chang, Y. Zhu, M. Li, and X. Sherman Shen, "Zoom: scaling the mobility for fast opportunistic forwarding in vehicular networks," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 2832–2840.

[14] L. Zhang, B. Yu, and J. Pan, "Geomob: A mobility-aware geocast scheme in metropolitans via taxicabs and buses," in *IEEE INFOCOM*, 2014.

[15] W. Sun, H. Yamaguchi, K. Yukimasa, and S. Kusumoto, "Gvgrid: A qos routing protocol for vehicular ad hoc networks," in *Quality of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on*. IEEE, 2006, pp. 130–139.

[16] K.-H. Chen, C.-R. Dow, S.-C. Chen, Y.-S. Lee, and S.-F. Hwang, "Harpiagrid: A geography-aware grid-based routing protocol for vehicular ad hoc networks." *Journal of Information Science & Engineering*, vol. 26, no. 3, 2010.

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

[18] T. M. Mitchell, "Machine learning. 1997," *Burr Ridge, IL: McGraw Hill*, vol. 45, 1997.

[19] W. Celimuge and K. Kumekawa, "Distributed reinforcement learning approach for vehicular ad hoc networks," *IEICE transactions on communications*, vol. 93, no. 6, pp. 1431–1442, 2010.

[20] C. Wu, S. Ohzahata, and T. Kato, "Routing in vanets: A fuzzy constraint q-learning approach," in *Global Communications Conference (GLOBECOM), 2012 IEEE*. IEEE, 2012, pp. 195–200.

[21] B. Yu, C.-Z. Xu, and M. Guo, "Adaptive forwarding delay control for vanet data aggregation," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 1, pp. 11–18, 2012.

[22] http://wirelesslab.sjtu.edu.cn, Wireless and Sensor networks Lab (WnSN),Shanghai Jiao Tong University.