Hierarchical Routing for Vehicular Ad Hoc Networks via Reinforcement Learning

Fan Li, *Member, IEEE*, Xiaoyu Song, Huijie Chen, Xin Li, *Member, IEEE*, and Yu Wang, *Fellow, IEEE*

***Abstract*—Vehicular *ad hoc* network is a collection of vehicles and associated road-side infrastructure, which is able to provide mobile wireless communication services. This highly dynamic topology structure is still open to many routing and message forwarding challenges. This paper addresses the issue of message delivery from vehicle to a fixed destination, by hopping over neighboring vehicles. We propose a reinforcement-learning-based hierarchical protocol called QGrid to improve the message deliver ratio with minimum possible delay and hops. The protocol works at two levels. First, it divides the geographical area into smaller grids and finds the next optimal grid toward the destination. Second, it discovers a vehicle inside or moving toward the next optimal grid for message relaying. There is no need of routing tables as the protocol builds a Q-value table based on the traffic flow in neighbor grids, which is then used for the grid selection. The vehicle selection process can employ different strategies, like, greedy selection of nearest neighbor, or solution based on the two-order Markov chain prediction of neighbor movement. This combination makes QGrid an offline and online solution. QGrid is further improved giving higher priority to vehicles with fixed routes and better communication capabilities, like buses, when making the vehicle selection. We have carried out extensive simulation evaluation by using real-world vehicular traces to measure the performance of our proposed schemes. The simulation comparisons among QGrid with/without bus aid, and existing position-based routing protocols, show the great improvement in the delivery percentage by our proposed routing protocol.**

***Index Terms*—Vehicular ad hoc network, routing, position-based routing, Q-learning.**

I. Introduction

VEHICULAR Ad Hoc Networks (VANETs) consist of mobile vehicles on the road and provides communication services among nearby vehicles or with roadside infrastructure. VANETs support a wide range of applications [1]–[6], which cover emergency alerts, road safety, landmark & destination location, prevention of collision & blind crossing, and information services, etc. In-car on-demand entertainment has emerged as another possibility with smart cars. Regardless of application, the essential problem is to efficiently deliver the data message. VANETs have unique characteristics which are quite different from traditional Mobile Ad Hoc Networks (MANETs), such as high speed, strong ability to calculate and unlimited power[7]. At the same time, the real urban environment has some special features which block signal transmission, such as buildings, overpasses, and fixed road. Moreover, the traffic patterns and other vehicular traffic regulations, make the topological changes extremely challenging for communication. The design of routing protocols for VANETs should overcome such challenges [7], [8].

The high speed of vehicles leads to highly dynamic network topology and frequent communication link disconnections. Delay Tolerant Networks (DTNs) specialize in intermittent connectivity, and some DTN-based routing methods could be used in VANETs [9], [10]. However, these DTN routing algorithms are fundamentally designed for small-scale networks of mobile devices carried by people. They cannot achieve an acceptable performance in large-scale vehicular networks. Position-based routing uses geographic information of the nodes when making the routing decision, as it assumes that each node knows its geographic location [7], [11]–[15]. The advantages of position based VANET routing are scalability, appropriate for high node mobility patterns and no need of route discovery and management. For example, GPSR [11] forwards the messages greedily based on neighbor's information, where it selects the neighbor node with minimal distance to destination as the next-hop. GRA [12] uses partial information and constructs routing table to solve the routing issues for wireless ad hoc networks. Li *et al.* [14] combine position-based routing with store-carry-forward mechanism, where the deriving direction of the vehicle is considered while forwarding or holding decisions are made. The key component of vehicular routing is the design of forwarding strategy, where vehicular characteristics can utilized. For example, PVCast [16] proposes the concept of packet-value to quantify the data preferences and makes dissemination decision based on the packet-value. The temporal dependency method [17] uses the contact-level mobility to decide the forwarding decision. Messages are delivered to nodes with minimal end-to-end delay, by utilizing the Inter Contact Times (ICTs) between each pair of vehicles obtained from the real life historical trajectory data. Probabilistic Control Centrality (pCoCe) [18] utilizes

Manuscript received May 16, 2018; revised September 28, 2018; accepted November 19, 2018. Date of publication December 17, 2018; date of current version February 12, 2019. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 61772077, 61370192, 61300178, 61772074, and 61572347; in part by the National Key R&D Program of China under Grant 2017FB0803300; in part by the US National Science Foundation under Grants CNS-1319915 and CNS-1343355; and in part by the U.S. Department of Transportation Center for Advanced Multi-Modal Mobility Solutions and Education under Grant 69A3351747133. The review of this paper was coordinated by Prof. P. Lorenz. *(Corresponding author: Fan Li.)*

F. Li, X. Song, H. Chen, and X. Li are with the School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: fli@bit.edu.cn; songxiaoyu@bit.edu.cn; chenhuijie@bit.edu.cn; xinli@bit.edu.cn).

Y. Wang is with the Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC 28223, USA (e-mail: yu.wang@uncc.edu).

Digital Object Identifier 10.1109/TVT.2018.2887282

the social-level mobility to decide the forwarding decision. The relay vehicles are selected according to the probabilistic control centrality, which accounts for the number of directed and diverse paths emanating from each individual vehicle. Zoom [19] is a protocol which use hybrid mobility of contact-level and social-level when making the forwarding decision. It captures the pairwise contacts between vehicles in microscopic aspect and the social relationships within VANETs in macroscopic aspect to calculate the shortest estimated delay. However, the vehicle which has the shortest estimated delay may have the worst communication quality, without taking into account the capabilities of the communication equipment. The interference between each pair of nodes at the MAC and routing layer also affects the performance of routing protocol, which is studied in [20]. It proposes a metric based on the maximization of the average Signal-to-Interference Ratio (SIR) level of the connection between source and destination, and then integrates it with an on-demand routing scheme and results in significant performance improvements.

The major objective of this work is to develop a novel position-based hierarchical routing protocol for vehicular as hoc networks. it should be capable to deliver messages from a mobile vehicle to a fixed destination with high probability. Although the vehicles move quickly in VANETs, in the macroscopic view, the vehicles exhibit some significant patterns. For example, the destinations of the taxis starting from the railway stations and airports have a high probability to go to city center are. Similarly, the buses always follow fixed routes. Our proposed QGrid considers both macroscopic aspect and microscopic aspect when making its routing decision. We divide the geographic region into fixed grids, extract the historical traffic flow from the real-world GPS trace data of vehicles, and compute the Q-table via Q-learning according to the volume of traffic flow. The values of Q-table indicates the possibilities of a vehicle entering different next-hop grids. We select the next-hop grid which has the largest Q-values.

After selecting the next direction, the current vehicle can choose the relay vehicle inside the next-hop grid based on two methods: 1) a greedy method which selects the nearest neighbor vehicle to the destination or 2) a Markov prediction method which selects the next vehicle based on the probability of moving to the optimal next-hop grid predicted by the two-order Markov chain. Moreover, we further improve the performance by giving higher priority to buses when making the vehicle selection. Each bus follows are timetable and moves along a fixed route which are consistent over long periods of time. We have carried out extensive simulations by using real vehicular traces to evaluate the performance of our proposed scheme. The results have shown a significant increase in the delivery ratio against other position based and greedy protocols.

The organization of rest of the paper is as follows. Section II provides a comprehensive review and analysis of current VANETs routing proposal. We propose our QGrid routing protocol in detail in Section III. Section IV introduces an enhanced version of QGrid with the consideration of fixed route buses. Simulation results over different position based methods are demonstrated in Section V-E, followed by Section VI which concludes the paper.

II. Related Work

Hierarchical Routing provides self-organization capabilities which makes the deployment of large scale network possible. In hierarchical architecture, the environment is divided into different groups and a subset of nodes are selected as the heads in charge of data transmission [7], [21], [22]. GVGrid [23] proposes an on-demand routing protocol, which determines the route from a fixed source to mobile vehicles in a specific destination region or area. The destination regions are pre-defined uniform sized grid squares. The objective is to provide a position-based routing protocol that constructs and maintains a stable route. However, it requires that every vehicle is GPS capable and has a digital map to geo-locate its position and direction. TTBR [24] proposes a two-level trajectory-based routing protocol. When a message needs to be transmitted, the protocol sends a packet to discover a cell path. Then the protocol transmits the message along the cell path found in advance and selects next hop vehicle in specific cell using local map, which increases the time and space overhead. HHLS [25] combines Greedy Perimeter Stateless Routing (GPSR [11]) and Hierarchical Location Service (HLS [26]) to enhance the network performances while reducing the location overhead. In CBR [27], uses clustering mechanism on top of square grids. The cluster heads are selected on each grid which relay the information based on their geographic information This protocol does not consider velocity and direction, which are significant factors in VANETs. HarpiaGrid [28] is also a geographic based hierarchical routing algorithm, which generates a grid sequence with shortest transmission distance according to the digital map. Our proposed QGrid divides the geographical area into uniform-size squares, then the messages are delivered along the grid sequences. The difference between QGrid and other two existing methods, i.e., HarpiaGrid and GVGrid, is that we do not need the digital map, instead, we assume that the geographical position of every vehicle can be obtained either from some localization methods or GPS equipment.

Reinforcement Learning (RL) is used to maximize the performance, by allowing machines and software agents to automatically determine the ideal behavior within a specific context. This can be done using reward mechanism (reinforcement signal) for the agent to learn its behavior [29]–[31]. By enabling mobile node to observe and gather information from their dynamic environments, RL has shown its advantages in making efficient routing decisions [32], [33]. QLAODV [34] is a distributed RL routing protocol, which infers VANET environment state information by using a Q-Learning algorithm and checks the path availability by using unicast control packets in a real time. The convergence speed of Q-learning algorithm can be adversely effected, if the learning states are too many. Q-learning is a form of model-free RL, which is a popular RL approach. A major advantage of Q-learning is the ability to obtain optimal control strategy from delay rewards, even in the absence of past knowledge of actions taken. In addition, Q-learning is also used in the field of data collection and aggregation.

A-STAR [35] uses fixed bus routes to build a sequence of anchor points along the route, which can be optimized for data forwarding and connectivity. However, the communication path is dependent on the path of bus which are primarily not designed for data traffic and may not optimally cover the whole geographic area. MIBR [36] is another reactive position based protocol, which heavily relies on the location and path of buses in the area. Buses mimic a mobile infrastructure for data forwarding. MIBR selects the road segment towards destination, based on the minimum number of bus-hops on that segment. CBS [37], [38] analyzes the real traces of 2,515 buses in Beijing and discovers a strong community structure in the bus lines of these buses, then it proposes a community-based backbone and a routing scheme over the backbone. However, the method is only suitable for sole vehicle type, namely bus, and relies on specific city transportation planning. GeoMob [39] employs two level of mobility patterns of the whole network and the individual vehicles (taxis and buses). The taxis have different mobility patterns due to drivers' behaviors and the buses follow fixed routes. The message is transferred between taxi and bus without bias in GeoMob.

Different from the protocols mentioned above, our proposed protocol has the following features. 1) We separate the geographical area into uniform-sized grids, then calculate the optimal grid sequence from each grid to the destination by applying reinforcement learning. Unlike the existing methods, we regard different grids as environment states of Q-learning instead of the vehicles. In this way, the learning states are greatly reduced, and convergence speed is increased. In addition, the Q-value table is the result of long term learning of environment by the agent. It is a continuous and iterative process, hence the actions are selected for long-term benefits, rather than short-term or immediate advantages. Furthermore, since the distributions of vehicles in different grid regions are relatively stable, thus we can study the Q-value table offline. 2) Unlike the existing methods, we do not select a fixed cluster head to be in charge of the inter-grid message transfer. We choose the relay vehicle inside the grid through a greedy method or a Markov prediction method. 3) We further improve the performance by giving higher priority to buses when making the vehicle selection.

III. QGrid: Q-Learning and Grid Based Routing

QGrid is a combination of Q-learning and grid based routing. Q-learning is a type of model-free reinforcement learning method, which provides and optimal control strategy from designed rewards even when the agent do not have the prior knowledge about the actions on the environment. Grid based routing enables splitting of geographic area into small grids, where an optimal sequence of grids can be determined (from source to destination) through Q-learning.

In this section, we first analyze the GPS data from real life traces to establish that the historical vehicular data has strong regularity. following that we discuss our QGrid in detail.

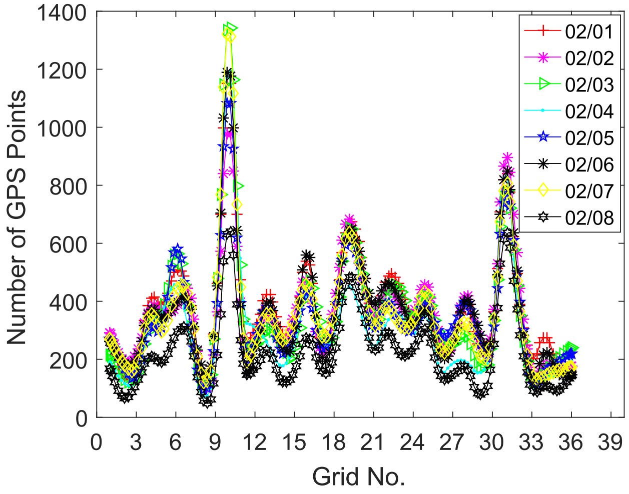


Fig. 1. GPS location count for taxis inside each grid from Feb. 1 to Feb. 8, 2007.

A. Data Features

SUVnet-Trace data [40] contains GPS data of 2,299 taxis and 2,500 buses of 103 routes in Shanghai from February 1 to March 3, 2007. Taxis periodically send GPS reports back to data center every one minute with customers or 15 seconds without customers and buses report once every minute. The information in their reports contains: ID, latitude/longitude, time stamp, movement speed, and status. We have performed statistical analysis to extract features from the data between February 1 to February 8, 2007 within the area of 3000 m × 3000 m around Shanghai railway station. This area contains GPS data from 2,066 taxis and 1,107 buses of 43 routes, and it is the traffic center of Shanghai. We divide the area into uniform grids with length of 500 m, which is related to the communication range. Fig. 1 shows the number of GPS points of taxis inside each grid. The horizontal axis is the grid number, and the vertical axis is the number of GPS points inside each grid. The figure demonstrates that the numbers of GPS points inside each grid are relatively stable in different days, which means that the vehicle's historical trajectory data has very strong regularity, thus we can learn the feature of historical trajectory offline and use it online later. Fig. 1 only shows the GPS data of taxis. Adding location information from buses does not change the regularity of data, as the routes of buses are the same every day. Hence, utilizing this feature to transfer the messages may be more efficient. Note that later in the simulation section, we use this real life dataset to conduct our simulations.

B. Q-Learning

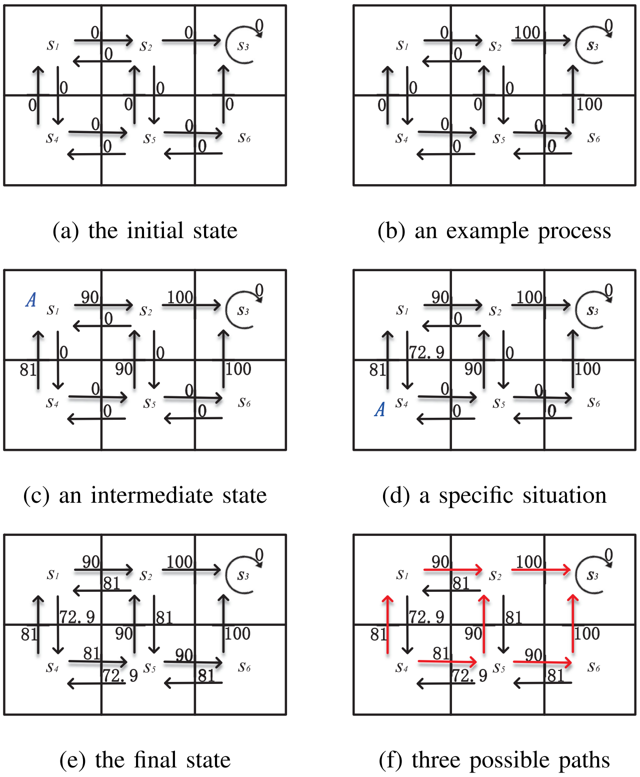
In general, the reinforcement learning model comprises of a set of agent actions (denoted as A), a set of environment states (denoted as S), a reward function, and a transfer function. The immediate reward $f_R(s_t, a_t)$ is received after taking action a_t in state s_t at time t. Transfer function $f_S(s_t, a_t)$ means that state s_t changes to another state when taking action a_t at time t, $s_t \in S$, $a_t \in A$. An agent constantly interacts with the environment in order to learn a control strategy. The agent observes that the current environment status is s_t, and then selects an action a_t. After taking this action, it gains the reward value $r_t = f_R(s_t, a_t)$, and the system state changes to $s_{t+1} = f_S(s_t, a_t)$.

However, it is impossible to determine the reward, unless the destination receives the message in a vehicular network. Thus, the model-based approach is impractical. Q-learning [29] is a model-free reinforcement learning method. It is usually used to find the optimal action-selection strategy for any finite Markov Decision Process (MDP) even when the agent does not have prior knowledge about the effect of its actions on the environment. Therefore, we adopt Q-learning method in this paper. $Q(s_t, a_t)$ is a real value corresponding to state-action pairs, called Q-value. Usually, in Q-learning algorithm, the learning rate α determines to what extent the newly acquired information will override the old information. A factor of 0 makes the agent not learn anything, while a factor of 1 makes the agent consider only the most recent information. In our work, α is set to the same empirical value as in QLAODV [34]. Let a' represent the next action corresponding to the next state. The main idea of Q-learning algorithm is to update Q-value iteratively by using Equation (1) until convergence is achieved. The optimal policy can be constructed by simply selecting the action with the highest value in each state.

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

In Equation (1), discount factor γ is a very important parameter for the Q-learning algorithm which determines the importance of future rewards. It is a constant value in the traditional Q-learning algorithm, however, in order to better measure the pros and cons of different grids, we use a piecewise function to set γ dynamically based on the number of vehicles in different girds. The goal of QGrid is to make the selected vehicles travel along the grid sequence with high vehicle density. Because the states and actions are discrete, reward function is bounded, and the pairs of (s, a) can be accessed with infinite times. Q-learning function can converge in limited time according to the Q function.

The design goal of QGrid routing is to transfer message from a source grid to a destination with high probability, which exhibits similarity with the classical example in Q-learning algorithm. Fig. 2 shows a simple example to demonstrate how Q-learning works. In the figure, each grid s_i represents a distinct state in S. The arrow represents a distinct action moving between neighbor grids, i.e., Up, Down, Left, and Right. The value above each arrow shows the Q-value of this moving direction. In this example, $f_R(s_t, a_t)$ gives 100 points of reward if this action enters the goal state, and nothing for other actions. For illustration, we set discount factor as a constant value $\gamma = 0.9$, and learning factor $\alpha = 1$ in this example. However, the γ is set dynamically based on the number of vehicles in different girds to make the messages transferred along the grid sequence with high vehicle density. Q-value table includes Q-values of all possible moving directions of the agent for each grid, which is presented by the arrows in the figure. It is computed offline after some period of learning.



(a) the initial state (b) an example process

(c) an intermediate state (d) a specific situation

(e) the final state (f) three possible paths

Fig. 2. Q-learning process illustration. (a) shows the initial state. (b) shows a general process of how to calculate the Q-value. (c) and (d) show an example that agent A locates in s_1 and move into s_4, how the Q-value is calculated. Assume s_4 is the source grid and s_3 contains the destination. (e) shows the final Q-values when Q-learning is completed. (f) shows the three possible paths marked with red which have the highest Q-values among all possible directions.

If s_3 is set as the grid where the destination locates, then the neighbor grids s_2 and s_6 can reach s_3 through one hop. According to Equation (1), $Q(s_2, a_{s_2\rightarrow s_3}) = 100$ and $Q(s_6, a_{s_2\rightarrow s_3}) = 100$ as shown in Fig. 2(b). We calculate and update the Q-values through continuous iteration by using Equation (1). In the beginning, the Q-values are set as zeros, which are shown in Fig. 2(a). An agent is randomly placed in one grid, which can move to any neighbor grids. In this process, $Q(s_t, a_t)$ is updated by using Equation (1). Assume that the agent A is located in s_1 at some point as shown in Fig. 2(c). The agent can move to s_4 or s_2, so $Q(s_1, a_{s_1\rightarrow s_4}) = 0 + 0.9\max\{81, 0\} = 72.9$, and $Q(s_1, a_{s_1\rightarrow s_2}) = 0 + 0.9\max\{90,90,100\} = 90$ according to Equation (1). When the agent arrives the specific grid, the relevant Q-value is updated. Fig. 2(d) shows the updated Q-values after agent moving into s_4.

The agent randomly moves between grids, and the Q-values are updated iteratively by using Equation (1) until convergence is achieved which means the Q-values do not change. In the practical, multi agent can be used to speed up the convergence. Fig. 2(e) shows a final stable status. The vehicle forwards messages by querying the Q-value table and follows the direction with the highest Q-values among four possible directions. Given the source grid s_4 and destination grid s_3, three possible paths can be selected to send message, namely, $s_4 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3$, $s_4 \rightarrow s_5 \rightarrow s_2 \rightarrow s_3$, or $s_4 \rightarrow s_5 \rightarrow s_6 \rightarrow s_3$, which are marked in red lines in Fig. 2(f).

C. QGrid Routing

The challenge we face is the use of reinforcement learning model to characterize the vehicular ad hoc network. In QGrid routing method, we first divide the geographical area into uniform-size squares called *grids*. Unlike the existing methods, we regard different grids as environment states S of Q-learning instead of the vehicles, thus the number of learning states will be greatly reduced. Q-value tables can be learned offline and used during the whole time due to the stability of the number of GPS points in each grid. We assume that the system has a virtual agent, and the action each agent can take is to send a message from one grid to its neighbor grid. The agent infers the environment from the reward r_t. We define the reward r_t to represent whether the message is delivered to the grid with destination or not. When taking this action, if the destination is inside the grid, the reward r_t is 100, and 0 otherwise, i.e.,

$$r_t = \begin{cases} 100 & \text{if the destination is inside the grid;} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The discount factor γ is a dynamic parameter which depends on the number of vehicles in each grid. It is easy to understand that the grid through which more vehicles pas is more likely to find an appropriate vehicle to transmit messages inside grid itself. Our objective is to choose the most reliable links to transfer messages, thus we adopt a piecewise function to characterize the changes of the discount factor. Let $\#(s_i)$ denote the number of vehicular 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 $\overline{\#}(s) = \frac{1}{N}\sum_{k=1}^{N}\#(s_k)$, in which N represents the total number of grids. γ is defined as

$$\gamma = \begin{cases} \min\left\{0.9, \beta \cdot \frac{\#(s_i)}{\overline{\#}(s)}\right\} & \text{if } \#(s_i) \geq \overline{\#}(s); \\ \max\left\{0.3, \beta \cdot \frac{\#(s_i)}{\overline{\#}(s)}\right\} & \text{if } \#(s_i) < \overline{\#}(s). \end{cases} \tag{3}$$

This equation maps the value of γ ranging from 0.3 to 0.9 based on different vehicular densities in different grids. This enables us to distinguish different grids (i.e., when $0 \leq \gamma \leq 1$). At the same time, we do not want the Q-value gained from the neighbor grids to be too large (i.e., when $\gamma = 1$), which means Q-value is influenced dramatically by the maximized Q-value of the neighbor grid. We also do not want the Q-value to be too small (i.e., when $\gamma = 0$), which means the Q-value of neighbor grid has no effect on the Q-value calculation of current grid. Therefore we take a trade off by letting $\gamma \in [0.3, 0.9]$. β is set to 0.6 empirically. γ can also be extended to measure many other aspects, such as link bandwidth, cache size, node speed and direction. Initially the agent has no knowledge regarding the entire environment. All elements of Q-value table are set to be zero initially and computed offline by utilizing Q-learning method. Q-value table is pre-stored in each vehicle, then messages are transmitted based on querying the Q-value table and follows the direction with the highest Q-values.

QGrid is a hierarchical routing protocol based on reinforcement learning. In the macroscopic aspect, the vehicle does not need to maintain the routing table, as it selects next-hop grid which has the maximized Q-value learned offline. In the microscopic aspect, the vehicle locally determines the specific vehicle inside the optimal selected grid as next-hop vehicle online, using *Vehicle Selection Strategy* which will be discussed in Section III-D. Thus QGrid capitalizes on the advantages of both offline and online methods.

We define v_i as the current vehicle holding the message, $neighbor(v_i)$ as the set of neighbor vehicles of v_i, s_j as the next-hop grid, $mem(s_j)$ as the set of vehicles in grid s_j, $d(v_i, v_j)$ as the distance between v_i and v_j. After we have selected the next-hop grid s_j through Q-learning, there are three cases we need to consider when choosing the relay vehicle inside this grid:

- *Case 1:* If there exist more than one v_i's neighbor vehicles inside the optimal next-hop grid s_j, current vehicle v_i will forward message to the selected next-hop vehicle in the neighbor vehicles based on *Vehicle Selection Strategy*. This strategy determines how to choose one vehicle inside the selected grid to be the next-hop vehicle, which will be discussed in Section III-D.
- *Case 2:* If there is no v_i's neighbor vehicle inside the next-hop grid s_j, current vehicle v_i will select the next-hop vehicle among the neighbor vehicles which has the shortest distance to the destination.
- *Case 3:* If current vehicle v_i can not find a neighbor vehicle which has closer distance to the destination than v_i itself, v_i will hold the message and wait for next forwarding opportunity.

In addition, to prevent the deadlock loop in our algorithm and handle the node leave, each message has a TTL field. In every transmission opportunity, when the message can not be transferred to destination $\boldsymbol{D}$, TTL is decreased by 1. Then if the $TTL > 0$, the message is transferred according to the designed strategy. If TTL expires, the message will be discarded. The details of QGrid algorithm is described in Algorithm 1. Assume every vehicle has a copy of Q-value table learned offline by using Equation (1).

D. Vehicle Selection Strategy Inside Optimal Grid

After querying Q-value table learned offline, we know which neighbor grid is the optimal next-hop grid, but how to select the relay vehicle inside this grid is still unknown. In this subsection, we propose a *Vehicle Selection Strategy* to select the specific vehicle. Two strategies will be introduced, i.e., *greedy selection strategy* (named QGrid_G) and *Markov selection strategy* (named QGrid_M). Regarding *greedy selection strategy*, vehicle v_i selects the nearest neighbor vehicle v_k to the destination *D*.

Markov selection strategy uses two-order Markov chain to predict the vehicle's next location grid. Some researches has been done utilizing Markov property in the design of Delay Tolerant Networks and Vehicular Ad Hoc Networks [41]–[43]. The meeting time span between nodes can be predicted by using the Markov model in DTN [41]. The message delivery process can be modeled as a Markov chain and finding an optimal value for the message-replication limit of each message can reduce

Algorithm 1: QGrid: Q-learning and Grid Based Routing.

```
Vehicle $v_i$ ($v_i \in Grids_i$) has a message destined to a
specific location $\boldsymbol{D}$, which is inside Grid $s_d$.
if $\boldsymbol{D}$ locates in vehicle $v_i$'s transmission range then
    Transfer the message to $\boldsymbol{D}$;
else
    $TTL = TTL - 1$;
    if $TTL > 0$ then
        Select next-hop Grid $s_j$ from querying its Q-value
        table whose destination grid is $s_d$ and has the
        maximized Q-value through this moving direction;
        if $mem(s_j) \bigcap neighbor(v_i) \neq \Phi$ then
            Select next-hop vehicle inside Grid $s_j$ by
            applying Vehicle Selection Strategy;
        else
            for all vehicle $v_k \in neighbor(v_i)$ do
                Calculate the distance between neighbor $v_k$
                and destination location $\boldsymbol{D}$;
            end for
            Select the neighbor $v_m$ which has the minimum
            distance to destination location;
            if $d(v_m, \mathbf{D}) < d(v_i, \mathbf{D})$ then
                $v_i$ sends message to its neighbor $v_m$;
            else
                $v_i$ continues to hold the message and wait for
                next forwarding opportunity;
            end if
        end if
    else
        Discard the message;
    end if
end if
```

Algorithm 2: Markov Selection Strategy, QGrid_M.

```
Let $\{v_1, v_2, \ldots, v_q\} = mem(s_j) \bigcap neighbor(v_i)$;
if $q = 1$ then
    $v_i$ forwards $M$ to this only vehicle;
else
    Find the next-hop grid $s_l$ of current Grid $s_j$ by
    querying Q-value table;
    for all $v_j \in (mem(s_j) \bigcap neighbor(v_i))$, $j \in [1, q]$
    do
        Use two-order Markov chain to calculate
        conditional probability $Pr_{v_j}(s_l|s_j s_q)$, and $s_q$ is the
        previous grid of $v_j$.
    end for
    Choose the vehicle with the maximized conditional
    probability $Pr_{v_j}(s_l|s_j s_q)$ as the next-hop vehicle;
end if
```

the delivery delay in [42]. The regular pattern of movements are fully mined by using Markov model to improve the delivery ratio of the DTN routing in [43]. In our work, We choose the neighbor vehicle as the relay vehicle which has the highest conditional probability of moving to the optimal next-hop grid based on two-order Markov chain referring to historical information. We first extract the grid sequence of each vehicle based on the historical GPS trajectory data, which can be described as $\{s_{k_1}, s_{k_2}, \ldots, s_{k_n}\}$. This grid sequence is a set of sample values of random process $\{G_i | G_i = s_{k_i}\}$, $i \in [1, n]$. The probability of using m order Markov chain to predict vehicle's next position grid is expressed as: for $n > m$,

$$Pr(G_n = s_{k_n} | G_{n-1} = s_{k_{n-1}}, G_{n-2} = s_{k_{n-2}}, \ldots, G_1 = s_{k_1})$$
$$= Pr(G_n = s_{k_n} | G_{n-m} = s_{k_{n-m}}, \ldots, G_{n-1} = s_{k_{n-1}}.$$

We assume that the current state situation is only related to the past m states, in other words, the future state depends on the past m states. Our problem can be expressed for a given set of past states of random process as $\{G_j | G_j = s_{k_j}\}$, $j \in [1, m]$, how to predict the state of G_{m+h}, denoted by

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

where m stands for the past m states and h stands for the h steps forwards from the current state. In our proposed protocol, we set $m = 2$ and $h = 1$. In other words, we use the past two-grid sequence to predict the next grid. The reason why we apply two-order Markov chain instead of higher order is that higher order Markov chain leads to high computational complexity. Additionally, the increase of complexity may not improve the precision of the prediction. We calculate one step transition matrix of the two-order Markov chain based on the historical vehicular trajectory data.

Again, we assume the current vehicle v_i in grid s_i carries message M to destination *D*. Grid s_d is the grid covers destination *D*. The optimal next-hop grid of s_i is s_j, and the next-hop grid of s_j is s_l determined by Q-learning. In other words, the optimal grid sequence passing by message M is $s_i \rightarrow s_j \rightarrow s_l$. There are two cases when choosing optimal next-hop vehicle:

- *Case 1:* If there exists only one v_i's neighbor vehicle inside grid s_j, it is no doubt that this vehicle is chosen as the relay.
- *Case 2:* If there exists q ($q > 1$) v_i's neighbor vehicles inside s_j, namely, $v_j \in (mem(s_j) \bigcap neighbor(v_i))$, $j = 1, 2, \ldots q$. We choose relay vehicle based on transition matrix, i.e., we select the vehicle which has the highest conditional probability among $Pr_{v_j}(s_l|s_j s_q)$ ($j = 1, 2, \ldots q$) as the optimal next-hop vehicle, where s_q is the previous location grid of neighbor vehicle, s_j is the current location grid of neighbor vehicle, and s_l is the next location grid of neighbor vehicle.

Algorithm 2 describes the *Markov selection strategy*.

In this QGrid proposal, it is possible for any vehicle to forward a message into eight different neighboring grids, which is also evident from Fig. 3. The transmission range of the vehicles is depicted by blue circles. The vehicle find the optimal next-hop grid by checking the Q-value table, regardless of its own grid. The total number of grids in the area determines the size of Q-value table, hence, larger the number of grids, larger the Q-value table will be.

We use Fig. 3 to illustrate the idea of QGrid_M routing protocol. In this figure, the blue vehicle v_0 in grid s_{26} carrying

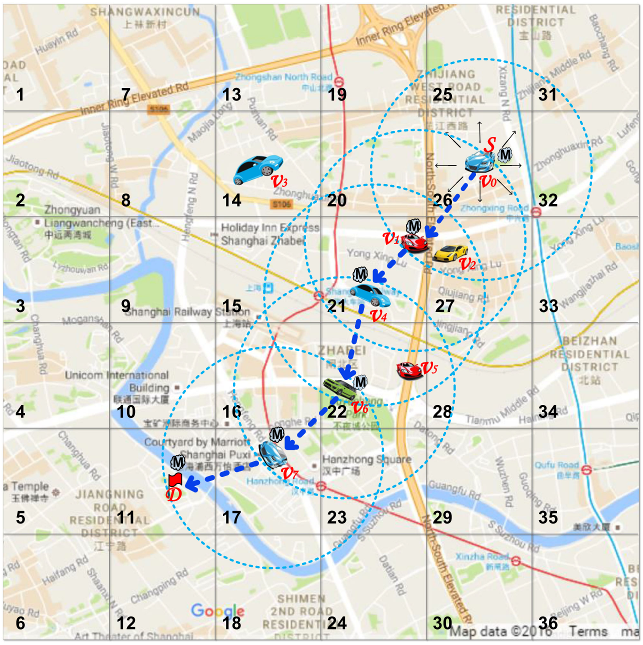


Fig. 3. Example of message forwarding using QGrid_M. The blue vehicle v_0 in grid s_{26} carrying message M destined for fixed destination D in grid s_{11}.

message M destined to a fixed location *D* which is inside grid s_{11}. The blue vehicle v_0 finds that the optimal next-hop grid to destination grid s_{11} is grid s_{21} by querying the Q-value table loaded in advance. There is no doubt that the red vehicle v_1 in grid s_{21} will serve as the next-hop vehicle because there is only one vehicle in grid s_{21} within the transmission range of the blue vehicle. Next, the optimal next-hop grid is grid s_{22} when message M is in grid s_{21}. Unfortunately, there is no vehicle in grid s_{22} among red vehicle's neighbor list. Then the red vehicle v_1 selects the nearest neighbor vehicle (i.e., the blue vehicle v_4) to the destination as the next-hop vehicle. The optimal next-hop grid is still grid s_{22}. There are two candidate vehicles in grid s_{22}. By applying *Markov selection strategy*, we find that the green vehicle v_6 has higher conditional probability to the optimal next-hop grid s_{17}. Thus the message is forwarded to this green vehicle v_6 then the next-hop vehicle selected is the blue vehicle in grid s_{17} since it is the only choice. Finally, this blue vehicle v_7 forwards the message to the destination location *D* in s_{11} which is within its transmission range.

In Algorithm 2, we first select the neighbor vehicles of v_i in grid s_j, the time complexity is bounded by $O(n)$, where n is the scale of the vehicles. Then finding the next-hop grid s_l of current grid s_j by querying Q-value table can be done within $O(m)$, where m is the scale of the grids. Last, to calculate conditional probability $Pr_{v_j}(s_l|s_j s_q)$ by using two-order Markov chain for selected neighbor vehicles, the time complexity is also bounded by $O(n)$. This is the situation that one node has message needed to be transferred. When all the nodes are considered, the time complexity of Algorithm 2 is $O(n(n+m))$. Likewise, the time complexity of Algorithm 1 or Algorithm 3 is $O(n(n+m))$ too.

Note that our method allows vehicle to hold the message if it cannot find an appropriate neighbor to transfer the message. We select next-hop grid s_j by querying its Q-value table which has the maximized Q-value to the destination grid s_d, so the message has the highest opportunity to be forwarded to the destination from a global network perspective. Since the route selection first finds the optimal grid towards the destination instead of the individual vehicle, the individual node leave or failure does not affect much of the performance of our method. In addition, we select the next hop vehicle which has the maximized probability moving towards the next optimal grid using two-order Markov chain, so the message has little chance to be transferred back from the vehicle perspective. Therefore, it is almost unlikely that the loop happens. Even if this little probability event happens, the message will not be trapped in the loop for a long time, since TTL will ensure the message being discarded when $TTL = 0$. Moreover, each message only has one copy in our method, the number of message transferred in the network will not increase significantly even if the TTL is large.

IV. Advanced QGrid Routing With Bus-Aided

Buses usually have fixed routes and schedules, which do not change frequently in quite a long time, thus we further improve the performance of QGrid by giving higher priority to buses in vehicle selection. We design two types of enhanced QGrid routing methods with the aid of buses. QGrid is a hierarchical routing protocol, as the same as the macroscopic aspect of basic QGrid described in Section III-C. *Advanced QGrid* first selects the optimal next-hop grid though maximized Q-value from querying its Q-value table. Assume the current vehicle v_i ($v_i \in s_i$) has a copy of message destined to a specific location *D*, which is inside Grid s_d. The optimal next-hop grid is s_j.

In the first method, namely ***AdvQGrid1***, if there exists some v_i's neighbors which are also inside grid s_j, and if there is a bus when it travels along its fixed route, its transmission range covers *D*, then *Advanced QGrid* forwards the message to this bus. Otherwise, it chooses the vehicle as the relay inside grid s_j based on *Vehicle Selection Strategy* which is discussed in Section III-D. Remember that we propose *greedy selection strategy* and *Markov selection strategy*, thus we named this first method as ***AdvQGrid1_G*** or ***AdvQGrid1_M***, respectively.

Fig. 4 shows an example of ***AdvQGrid1***. Vehicle v_1 carries message M and the message's destination is *D* in grid s_{16}. Vehicle v_1 selects grid s_{11} as the next-hop grid by querying Q-value table.Vehicle v_5 is a bus and it travels along a fixed route passing through grid s_{11}, s_{10}, and s_{16} (shown in green line). In grid s_{11}, there are four neighbor vehicles of v_1. Based on *AdvQGrid1*, v_5 is selected as the next-hop vehicle for message forwarding because its route containing grid s_{16}. It will hold the message until it reaches grid s_{16} and finally deliver the message to the destination *D*.

Although the message can be taken to the destination definitely once it was transferred to the bus using the first method, but the transmission delay may be too long because the bus holds the message and forwards it to the destination only when the bus travels near the destination. In the second method, we still give bus the higher priority in the vehicle selection because the message will definitely be forwarded to the destination. At

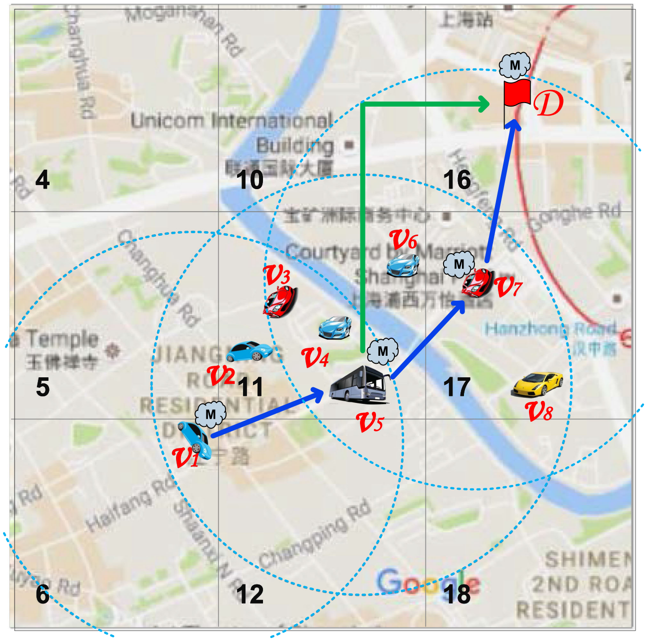


Fig. 4. An example of *Advanced QGrid*: ***AdvQGrid1*** follows green route, while ***AdvQGrid2_G*** follows blue route.

the same time, we allow the message transferring between buses and normal vehicles. If there are some normal vehicles which have relatively high probability to reach the destination by multiple hops, we transfer the message from bus to normal vehicles to take the advantage of high speed transferring in the air. Thus the second method has shorter delay than the first method but still maintain high message delivery ratio.

In the second method, the vehicle searches all its neighbors and selects the top k vehicles according to *greedy selection strategy* and *Markov selection strategy*. If there exits one bus in the top k vehicles and its route containing the message's destination, it is selected as the relay node. If there exists more than one bus meeting this condition, the top one bus is selected. If there is no bus in the top k vehicles, the top one vehicle of them is selected to be the relay node. Unlike in the first method, we allow message transferring between buses and taxis in this method. We name the second method as ***AdvQGrid2_G*** or ***AdvQGrid2_M***, respectively. The second method with *Bus Assistant Markov Selection Strategy* (i.e., ***AdvQGrid2_M***) is described in Algorithm 3.

Fig. 4 shows an example of *Advanced QGrid* with *greedy selection strategy*, i.e., ***AdvQGrid2_G***. v_1 has four neighbors v_2, v_3, v_4, and v_5. According to greedy strategy, v_1 searches all its neighbors and sorts them with the distance to the destination location *D*. Then top k vehicles are selected, in this paper, we set $k = 3$, which are v_4, v_3, and v_5. There is one bus, e.g., v_5, in the top three vehicles, so v_5 is selected to be the relay node. If there is no bus in the top three vehicles, v_4 is selected to be the relay because it is the top one in the rank. By querying Q-value table, v_5 knows the optimal next-hop grid is grid s_{17}. In grid s_{17}, v_5 has two neighbors, v_7 and v_8. v_7 is selected to be the relay node because it has the shorter distance to *D*. Then the message M will be delivered to the destination *D* by vehicle v_7 directly because its communication range covers the destination.

Algorithm 3: Markov Selection Strategy, ***AdvQGrid2_M***.

```
Let $\{v_1, v_2, \ldots, v_q\} = mem(s_j) \bigcap neighbor(v_i)$;
if $q = 1$ then
    $v_i$ forwards $M$ to this only vehicle;
else
    Find the next-hop grid $s_l$ of current Grid $s_j$ by
      querying Q-value table;
    for all $v_j \in (mem(s_j) \bigcap neighbor(v_i)), j \in [1, q]$
      do
        Use two-order Markov chain to calculate
          conditional probability $Pr_{v_j}(s_l|s_j s_i)$;
    end for
    Sort these vehicles by descending order according to
      their conditional probability $Pr_{v_j}(s_l|s_j s_i)$;
    Select top $k$ vehicles, defined as set $Top\mathcal{K}$; Assume
      $\mathcal{B}$ is the set of buses;
    if $Top\mathcal{K} \bigcap \mathcal{B} \neq \Phi$ then
        for all $v_j \in Top\mathcal{K} \bigcap \mathcal{B}$ do
            if $v_j$'s route containing the message destination
              $D$ then
                Select the top bus $v_j$;
            end if
        end for
    else
        Select the first vehicle in $Top\mathcal{K}$ as the relay
          vehicle;
    end if
end if
```

V. Simulations

We have conducted extensive simulations using real life vehicular data to evaluate the performance of our proposed QGrid and Advanced QGrid vehicular routing protocols and compare them with three other existing position-based or bus-aided protocols.

A. Compared Routing Protocols

We have used the following routing protocols for comparison with our QGrid and Advanced QGrid proposals.

- *GPSR* [11]: A geographic based routing algorithm which chooses routes with greedy policy. The current vehicle selects the neighbor vehicle which has the shortest distance to the destination.
- *HarpiaGrid* [28]: Each vehicle sends message following the shortest path, and it is assumed that each vehicle has the digital map.
- *CBS_like*: CBS [37], [38] proposes a community-based bus system as routing backbone for VANETs and the community division relies on the contact relationship of bus lines. However, the taxies do not have the fixed routes which can not be formed into the community, so we only use CBS

TABLE I
PARAMETERS USED IN THE SIMULATIONS

Parameter	Value or Range
α	0.8
β	0.6
γ	$[0.3, 0.9]$
experimental area	$3000m \times 3000m$
message generating rate	10 / second
communication radius	500m
grid length	$500m$
reward R	$0, 100$
time slot ΔT	$5s, 10s, 20s$
TTL	$10, 20, 30, 40, 50$
top-k in AdvQGrid	$k = 3$

for the performance comparison with bus-aided simulation. In the design of CBS, it uses 10 copies of message in the routing protocol. For the purpose of comparison, we only allow single copy of message during the routing process for all the compared protocols including our proposed QGrid and Advanced QGrid. Thus we call this modified CBS protocol as CBS_like.

B. Evaluation Metrics

We use the following metrics for evaluation:

- *Delivery ratio:* The portion of the messages that are received by the destinations out of the total messages generated.
- *Hop count:* The average number of hops during each successful delivery.
- *Delay:* The average interval of time required for successfully delivered messages.
- *Number of forwarding:* The average number of message forwarding in the network during the whole simulation period.
- *Throughput:* The average number of messages successfully transferred from source nodes to destination in unit time in the network versus node density.

C. Data Processing

Fig. 3 shows the selected are of 3000×3000 m which is around the Shanghai railway Station, taken from Shanghai vehicular dataset described earlier (Section III-A). Analysis of data set shows an anomaly, that, two vehicles are considered disconnected at any time t, even if they exist in each others transmission range but uploaded their GPS information at different times. To overcome this anomaly, we propose the use of time slots. In any given time slot ΔT, if two vehicles are in each others transmission range, they are considered to be neighbors for that time slot. The time slot is represented by ΔT, and the time field as $t = time/\Delta T$.

The proposed QGrid protocol is hierarchical in nature, with a two step process. The first step determines the appropriate next-hop grid. In the second step, a next-hop vehicle is selected from the grid selected in earlier step. In our evaluation process, we selected the source vehicle and destination location randomly. Table I summarizes the parameters for the evaluation process.

Since we use the real life vehicular data in the evaluation, thus the average number of vehicles in experimental area is constant. So in order to evaluate the throughput versus different node densities, in the low density vehicle scenario, we randomly remove one third of vehicles from the raw vehicle dataset; in the high density vehicle scenario, we interpolate one third of vehicles in the raw vehicle dataset.

D. Discussions on Parameters

How to set the value of the time slot ΔT is important. Large ΔT value means longer time slots during which vehicles may move across multiple grids. This will result in possible inaccurate location prediction. On the other hand, smaller ΔT may give very few GPS trajectory points, resulting in lower delivery ratio. As fine grained ΔT values can reflect the accurate algorithm performance, we evaluate the protocols on a range of time slot sizes, i.e., $\Delta T = 5$ s, 10 s, and 20 s in our simulations. In practice, the optimal value of ΔT can be obtained via similar simulations or experiments.

Another important parameter is the length of grid, which needs to be carefully selected to avoid the following scenarios:

1) *Grid Length too Small:* This sharply increases the number of grids in the area (and also the status in Q-learning), which will lead to slow convergence. In addition, the protocol we proposed selects the next-hop grid firstly, so the adding of grids will increase the hop count to message destination.
2) *Grid Length too Large:* This may cause that no vehicle is found in the optimal next-hop grid due to beyond transmission range (as compared to grid size). Moreover, the message might be transmitted within one grid and cannot be carried to another grid because the neighbor vehicles may be in the same grid with the vehicle carrying the message.

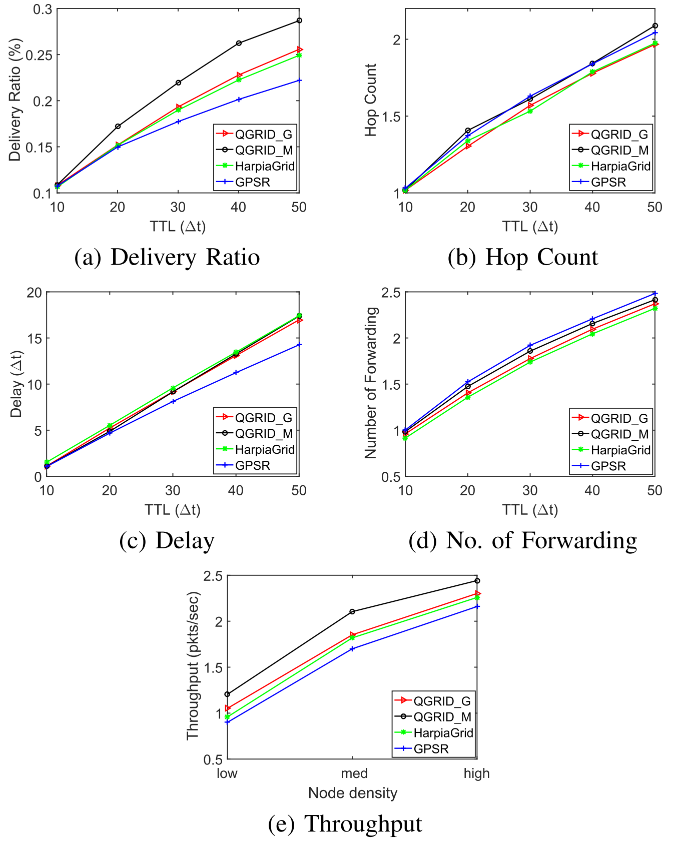
Hence, in our evaluation we have set the grid length to 500 m, which is equal to the transmission radius.

E. Simulation Results With Taxies

To evaluate the performance of QGrid, we have made use of two different selection strategies (i.e., QGrid_M and QGrid_G) for it. We then use the Shanghai dataset of real vehicles to test against existing protocols described in Section V-A.

QGrid_M predicts the next possible grid, using a two-order Markov chain, hence we can query the Q_Value table to obtain the optimal next-hop grid. Following this, QGrid_M determines the vehicle with highest conditional probability that will move to the earlier predicted next-hop grid.

Fig. 5 shows the simulation results with $\Delta T = 5$ s. In Fig. 5(a) we observe a gradual rise in delivery ratio with increase in TTL, which is due to longer life time of messages as it gets more time to reach the destination. The overall delivery ratio of both QGrid_M and QGrid_G is better than competing protocols, where QGrid_M is significantly higher due to selection of next-hop vehicle using historical trajectory data (two-order Markov chain). GPSR lacks the global information of networks, and only picks nearest neighbor, which will be suboptimal. HarpiaGrid

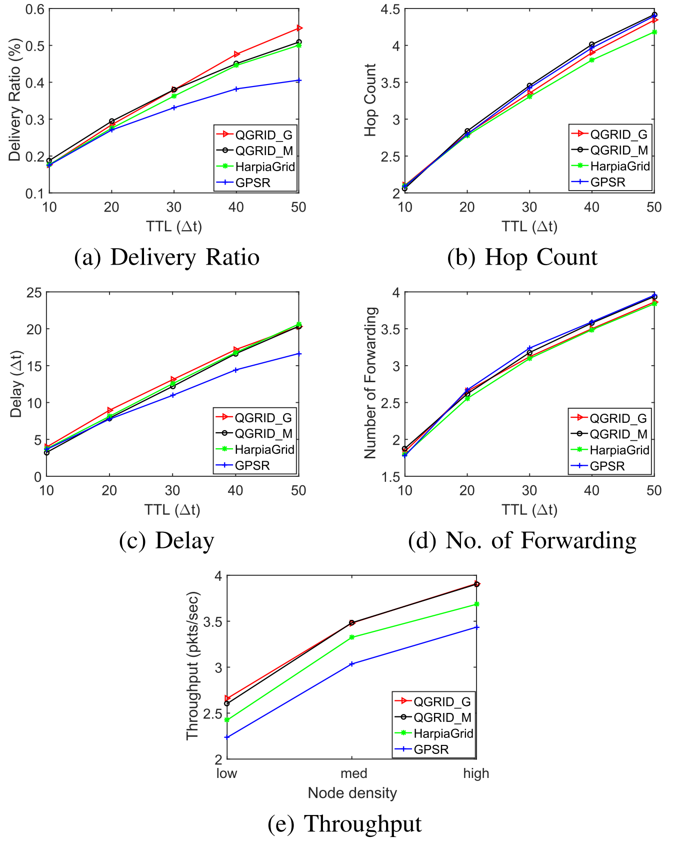


(a) Delivery Ratio (b) Hop Count

(c) Delay (d) No. of Forwarding

(e) Throughput

Fig. 5. Simulation comparison among different routing protocols with $\Delta T = 5$ s.



(a) Delivery Ratio (b) Hop Count

(c) Delay (d) No. of Forwarding

(e) Throughput

Fig. 6. Simulation comparison among different routing protocols with $\Delta T = 10$ s.

transfers the messages to fixed road. QGrid_M and QGrid_G hence better deliver as they select next-hop grid from a long term perspective. The hop count required by all the four protocols are relatively similar, with QGrid_G and HarpiaGrid performing marginally better than GPSR and QGrid_M, as seen in Fig. 5(b). Same behavior is observed for the delay metric in Fig. 5(c). At higher TTL, GPSR tends to have lower delay due to its nearest neighbor selection strategy, but the delivery performance is also lesser. Fig. 5(d) indicates the number of times a packet is forwarded, which is different then the number of hops. We observe that QGrid_G and HarpiaGrid tend to forward less number of times as compared to GPSR and QGrid_M. We simulate the throughput versus node density with $TTL = 30$. Fig. 5(e) indicates the results of throughput versus node density. We observe that the throughput of all protocols improve with the increase of node density, and QGrid_M and QGrid_G are better than competing protocols. In addition, the improvement of throughput from low node density to medium node density is higher than that from medium to high node density, which means that increasing node density can improve the throughput but the improvement is not unlimited.

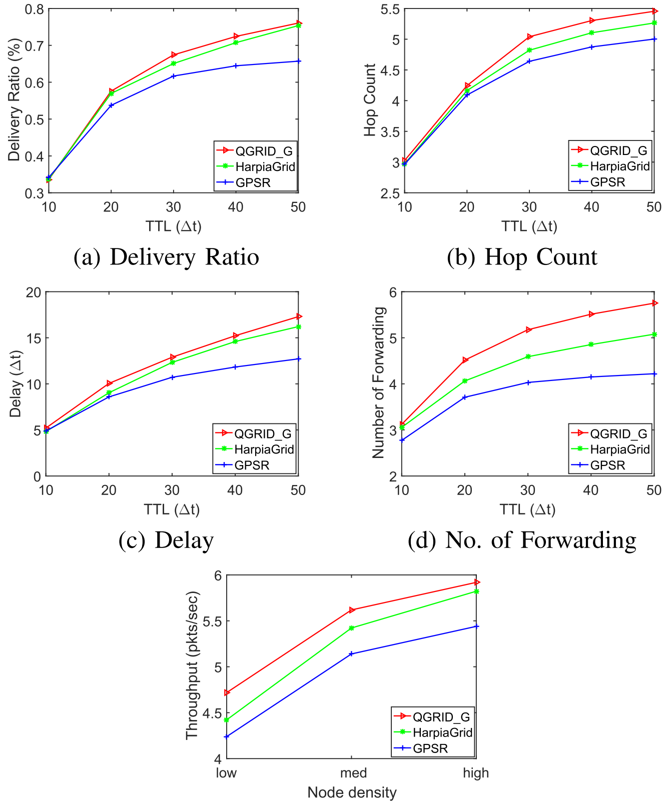
The overall delivery ratio observed with $\Delta T = 5$ s is not very high. This is mainly attributed to the fact that GPS trajectory points inside each grid are very few. The number of vehicles in neighborhood can be increased by increasing the time slot duration, which is represented in Figs. 6 and 7.

Fig. 6(a) with $\Delta T = 10$ s clearly shows an increase in overall delivery ratio as compared to that of $\Delta T = 5$ s. This proves that the higher number of available neighbor vehicles to be selected from can improve the delivery ratio. At the same time, due to larger ΔT value, vehicles can move multiple grids in the given time slot, resulting in slightly inaccurate Markov chain predictions (QGrid_M). For this reason QGrid_G has better performance. Fig. 6(b)–(e) show similar behavior for delay, hop count, number of forwards and throughput as that of smaller ΔT, for all protocols.

In Fig. 7 we have excluded QGrid_M due to even larger value of $\Delta T = 20$ s. The overall performance of delivery ratio increases further, with QGrid_G showing higher delivery ratio than GPSR and HarpiaGrid. At the same time, we also observe higher hop count, delay, and number of forwards for QGrid_G, which is mostly due to selection of *possible successful paths* rather than the shortest path. Hence the increase in delivery ratio at the cost of higher delay & hops can be accepted to a certain degree. Fig. 7(e) shows similar behavior for throughput as that of smaller ΔT, for all protocols.

F. Simulation Results With Bus-Aided

In this set of simulations, we provide the comparison among the algorithms by considering buses aided or not. The QGrid algorithms with bus-aided includes four types of methods, namely AdvQGrid1_G/M and AdvQGrid2_G/M, as described in

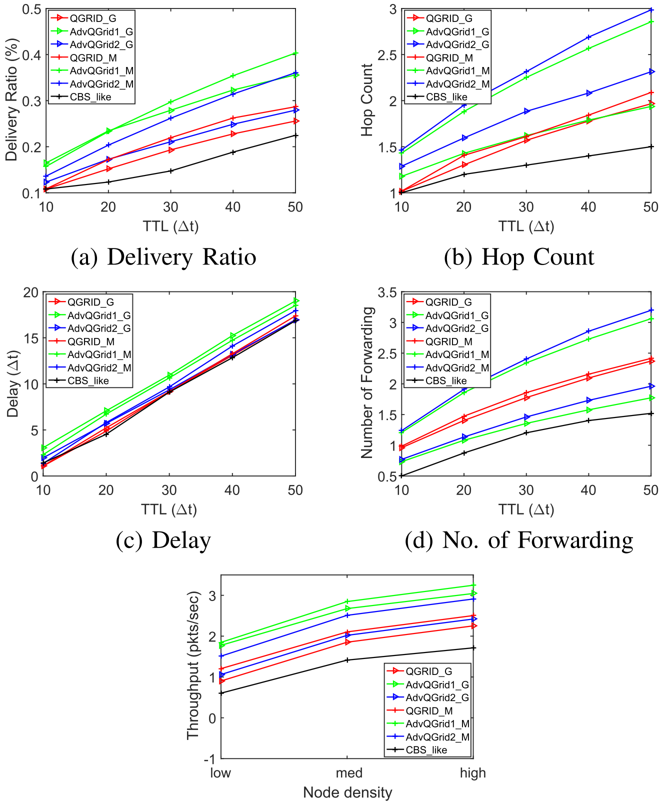


(a) Delivery Ratio (b) Hop Count

(c) Delay (d) No. of Forwarding

(e) Throughput

Fig. 7. Simulation comparison among different routing protocols with $\Delta T = 20$ s.



(a) Delivery Ratio (b) Hop Count

(c) Delay (d) No. of Forwarding

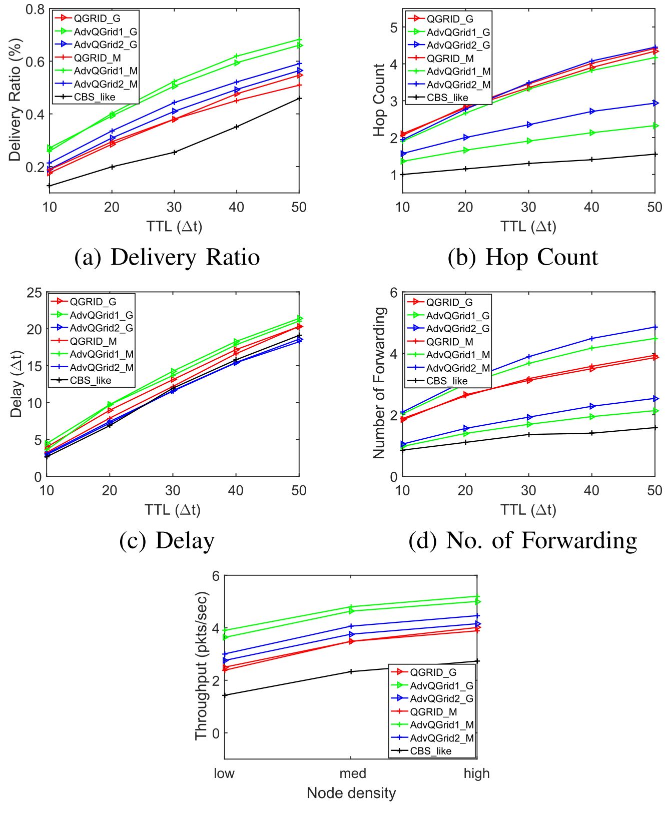
(e) Throughput

Fig. 8. Simulation comparison among AdvQGrid (with bus-aided), CBS_like (with bus-aided), and QGrid (without bus aided) when $\Delta T = 5$ s.

Section IV. We set k to 3 in AdvQGrid2_G/M, as shown in Table I. Unlike the evaluation of QGrid_G/M in Section V-E, we add buses trajectories in the simulation for the sake of fairness. In addition, we also compare our proposed method with CBS_like described in Section V-A. The CBS_like is a method which leverages communication among buses as the backbone, so in our simulation, the message can be generated by any vehicles, but it only can be transmitted among buses once it is transmitted to buses. We also perform the comparison by considering three situation: $\Delta T = 5$ s, 10 s and 20 s.

Fig. 8 depicts the simulation results with $\Delta T = 5$ s. Fig. 8(a) shows the delivery ratios of different routing algorithms. We can see that the delivery ratios of methods with Markov selection strategy are higher than those methods with greedy selection strategy which demonstrates that Markov prediction is better at selecting relay vehicle. QGrid with bus-aided have better delivery ratios than those without bus-aided and AdvQGrid1_G/M have the highest delivery ratios among all methods. Among all the protocols, the performance of CBS_like is not good. We think the most important reason is that CBS_like has few opportunity to transfer the message. And different from the experiments in [37], [38], we only allow single copy of message during the routing process (for fairness among different methods). Fig. 8(b) shows that the hop count of methods with Markov selection strategy is higher than those methods with greedy selection strategy. The CBS_like has the minimum Hop count among all the protocols as expected. We count the average numbers of hops during each successful delivery. Since CBS has fewer opportunity to transfer message, for each successful delivery, it has smaller hop count. In Fig. 8(c), the delay of methods AdvQGrid1_G/M is higher than other methods but the differences are not that much. Fig. 8(d) shows the number of forwarding with Markov selection strategy is higher than those with greedy selection strategy, followed by CBS_like. In Fig. 8(e), the throughput of methods AdvQGrid1_G/M is higher than other methods and the throughput of all the protocols improve with the increase of node density. In addition, the improvement of throughput from low node density to medium node density is higher than that from medium to high node density.

In AdvQGrid1_G/M, when a message is delivered to a bus, we assume that the message is delivered successfully and calculate the message delivery delay by consulting historical information, e.g., the average time from one bus stop to next bus stop and the average time stopped at the bus station. As shown in Fig. 8, AdvQGrid1_G/M obtain the highest delivery ratios with little more delay cost. In addition, the delivery ratios of AdvQGrid2_G/M are higher than QGrid_G/M, but are not dramatically different in delay, because AdvQGrid2_G/M utilize the feature of buses, e.g., fixed route, which increases their delivery ratios, and meanwhile, it considers the high density of taxis, which help to reduce the delay. In summary, the methods considering buses increase the delivery ratio with little higher delay and number of forwardings.



(a) Delivery Ratio

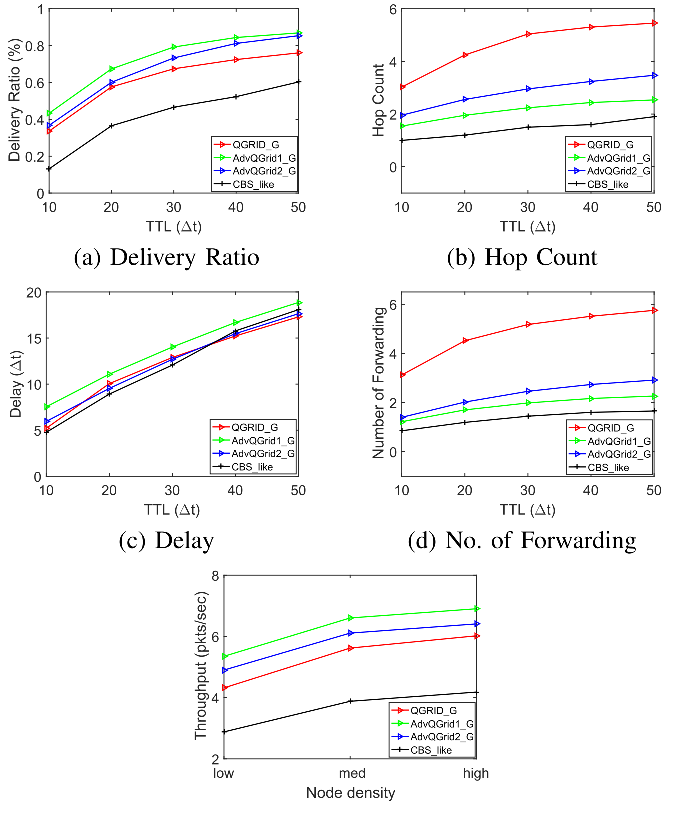
(b) Hop Count

(c) Delay

(d) No. of Forwarding

(e) Throughput

Fig. 9. Simulation comparison among AdvQGrid (with bus-aided), CBS_like (with bus-aided), and QGrid (without bus aided) when $\Delta T = 10$ s.



(a) Delivery Ratio

(b) Hop Count

(c) Delay

(d) No. of Forwarding

(e) Throughput

Fig. 10. Simulation comparison among AdvQGrid (with bus-aided), CBS_like (with bus-aided), and QGrid (without bus aided) when $\Delta T = 20$ s.

Similar to earlier sub-sections, various values of ΔT are used to evaluate the performance of different protocols. Fig. 9 shows the result when $\Delta T = 10$ s. The delivery ratios in Fig. 9(a) increase compared to those in Fig. 8(a) because the number of vehicles within the communication range will increase when $\triangle T$ becomes longer. AdvQGrid1_G/M still have the highest delivery ratios among all methods because when there is a bus available whose routes covers the peripheral of the destination, the message will be forwarded to this bus and the bus still hold the message till it reaches that area. Fig. 9(b) shows that the hop count of methods with Markov selection strategy is higher than those methods with greedy selection strategy, the same as in Fig. 8(b). In Fig. 8(b), the hop count of QGrid_G is lower than AdvQGrid1&2_G. But in Fig. 9(b), we can see that the hop count of QGrid_G is the highest, because when $\triangle T$ is longer, buses have higher possibilities to be selected. Meanwhile, selecting buses means less hop count. Fig. 9(c) shows that the delays of all methods are not that different. Number of forwardings with Markov selection strategy is higher than those with greedy selection strategy as seen from Fig. 9(d). CBS_like has the similar performance with Fig. 8. Fig. 9(e) shows similar behavior for throughput as that of smaller ΔT, for all the protocols.

When the time slot increases to 20 s, we do not include QGrid with Markov selection in the simulation due to the possibility of inaccurate prediction. Fig. 10(a) demonstrates that the delivery ratios get higher compared with previous two cases. Fig. 10(b) shows the hop count has the similar trends with those in Fig. 9(b). Fig. 10(c) shows the delay of AdvQGrid1_G is higher than the other two methods which are not dramatically different. Fig. 10(d) shows the comparison for number of forwarding, where the value of QGrid_G is the highest, followed by AdvQGrid2_G , AdvQGrid1_G and CBS_like. The result of throughput shows the similar behavior as that of smaller ΔT in Fig. 10(e). The throughput of AdvQGrid1_G is highest, followed by AdvQGrid2_G, QGrid_G and CBS_like. In addition, the improvement of throughput from low node density to medium node density is higher than that from medium to high node density. From Fig. 10, we can find that the methods with bus-aided can increase the delivery ratio, decrease the hop count, and does not increase number of forwardings & delay significantly.

The control information overhead in our proposed protocol is the hello packets for discovering the neighbor nodes. The communication range in our simulation is 500 m and the velocity of vehicle in urban environment is usually $30-50$ km/h, which means it needs $36-60$ s for a vehicle from entering to leaving another vehicle's communication range. Therefore, the period of sending the hello packets should not be larger than 36 s. For discovering the neighbor nodes more efficiently, we set the period to send the hello packets as 20 s. This control information exists for all protocols we have evaluated in our simulation. Compared with CBS_like, which also has extra computing overhead when it builds the community-based backbone in advance, our protocols has much higher delivery ratio and throughput than CBS_like (shown in Figs. 8–10). Compared with HarpiaGrid and GPSR, our proposed protocols need extra overhead for computing the Q-table, but the computing is carry out offline and only needed once in a long time. So the extra overhead caused by the computing is slight and constant. The simulation results validate that our proposed protocol improves the performance effectively with introducing slight extra overhead.

VI. Conclusion

Routing and message forwarding in vehicular ad hoc networks is major challenge. In this paper, we propose QGrid, which is an hierarchical routing protocol. QGrid improves delivery ratio from vehicle to fixed destination, by reinforcement learning. It has a data forwarding mechanism, which keeps macroscopic and microscopic aspects in view. The geographic area of vehicle is divided into grids, where size of grid is dependent on the transmission range of vehicle. The macroscopic aspect deals with the selection of optimal next-hop grid. This is done by using a Q-value table which is learned offline. The microscopic aspect is the selection of best vehicle for forwarding from the previously selected grid. Two algorithms can be used for vehicle selection, i.e., greedy selection of nearest neighbor towards destination, or selection based on two-order Markov chain prediction of a vehicle which will move to the next-hop grid. Furthermore, vehicles with predefined routes and better communication capabilities are given preferences as next-hop vehicles. Hence, QGrid becomes an efficient offline and online solution. We carry out extensive simulations using real-world vehicular traces. The simulation comparison among QGrid with/without bus-aided and existing position based or bus-aided routing protocols confirms that our proposed reinforcement learning based hierarchical routing can increase the delivery ratio, throughput, decrease the hop count without introducing too many message forwardings and delay.

References

[1] S. Misra, I. Zhang, and S. C. Misra, *Guide to Wireless Ad Hoc Networks*. London, U.K.: Springer, 2009.

[2] H. Hartenstein and K. Laberteaux, *VANET: Vehicular Applications and Inter-Networking Technologies*, vol. 1. New York, NY, USA: Wiley, 2010.

[3] P. Toth and D. Vigo, *Vehicle Routing: Problems, Methods, and Applications*, vol. 18. Philadelphia, PA, USA: SIAM, 2014.

[4] H. Han *et al.*, “SenSpeed: Sensing driving conditions to estimate vehicle speed in urban environments,” in *Proc. IEEE Conf. Comput. Commun.*, 2014, pp. 727–735.

[5] Z. Wu, J. Li, J. Yu, Y. Zhu, G. Xue, and M. Li, “L3: Sensing driving conditions for vehicle lane-level localization on highways,” in *Proc. IEEE Conf. Comput. Commun.*, 2016, pp. 1–9.

[6] A. Rasheed, S. Gillani, S. Ajmal, and A. Qayyum, *Vehicular Ad Hoc Network (VANET): A Survey, Challenges, and Applications*. Singapore: Springer, 2017.

[7] F. Li and Y. Wang, “Routing in vehicular ad hoc networks: A survey,” *IEEE Veh. Technol. Mag.*, vol. 2, no. 2, pp. 12–22, Jun. 2007.

[8] B. T. Sharef, R. A. Alsaqour, and M. Ismail, “Vehicular communication ad hoc routing protocols: A survey,” *J. Netw. Comput. Appl.*, vol. 40, pp. 363–396, 2014.

[9] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Efficient routing in intermittently connected mobile networks: The multiple-copy case,” *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 77–90, Feb. 2008.

[10] L. Zhang, X. Wang, J. Lu, M. Ren, Z. Duan, and Z. Cai, “A novel contact prediction based routing scheme for DTNs,” *Trans. Emerg. Telecommun. Technol.*, vol. 28, no. 1, 2014.

[11] B. Karp and H.-T. Kung, “GPSR: Greedy perimeter stateless routing for wireless networks,” in *Proc. 6th Annu. Int. Conf. Mobile Comput. Netw.*, 2000, pp. 243–254.

[12] R. Jain, A. Puri, and R. Sengupta, “Geographical routing using partial information for wireless ad hoc networks,” *IEEE Pers. Commun.*, vol. 8, no. 1, pp. 48–57, Feb. 2001.

[13] Y.-W. Lin, Y.-S. Chen, and S.-L. Lee, “Routing protocols in vehicular ad hoc networks a survey and future perspectives,” *J. Inf. Sci. Eng.*, vol. 26, no. 3, pp. 913–932, 2010.

[14] L. Zhao, F. Li, and Y. Wang, “Hybrid position-based and DTN forwarding in vehicular ad hoc networks,” in *Proc. IEEE Veh. Technol. Conf.*, 2012, pp. 1–5.

[15] Y. Huang, M. Chen, Z. Cai, X. Guan, and T. Ohtsuki, “Intersection-based forwarding protocol for vehicular ad hoc networks,” *Telecommun. Syst.*, vol. 62, no. 1, pp. 67–76, 2016.

[16] Q. Xiang, X. Chen, L. Kong, and L. Rao, “Data preference matters: A new perspective of safety data dissemination in vehicular ad hoc networks,” in *Proc. Comput. Commun.*, 2015, pp. 1149–1157.

[17] H. Zhu, S. Chang, M. Li, K. Naik, and S. Shen, “Exploiting temporal dependency for opportunistic forwarding in urban vehicular networks,” in *Proc. IEEE Conf. Comput. Commun.*, 2011, pp. 2192–2200.

[18] A. Stagkopoulou, P. Basaras, and D. Katsaros, “A social-based approach for message dissemination in vehicular ad hoc networks,” in *Proc. Int. Conf. Ad Hoc Netw.*, 2014, pp. 27–38.

[19] H. Zhu, M. Dong, S. Chang, Y. Zhu, M. Li, and X. S. Shen, “Zoom: Scaling the mobility for fast opportunistic forwarding in vehicular networks,” in *Proc. IEEE Conf. Comput. Commun.*, 2013, pp. 2832–2840.

[20] P. Fazio, F. De Rango, and C. Sottile, “A predictive cross-layered interference management in a multichannel MAC with reactive routing in VANET,” *IEEE Trans. Mobile Comput.*, vol. 15, no. 8, pp. 1850–1862, Aug. 2016.

[21] S.-G. Yoon, S. Jang, Y.-H. Kim, and S. Bahk, “Opportunistic routing for smart grid with power line communication access networks,” *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 303–311, Jan. 2014.

[22] X. Deng, L. He, X. Li, Q. Liu, L. Cai, and Z. Chen, “A reliable QoS-aware routing scheme for neighbor area network in smart grid,” *Peer-to-Peer Netw. Appl.*, vol. 9, no. 4, pp. 616–627, 2016.

[23] W. Sun, H. Yamaguchi, K. Yukimasa, and S. Kusumoto, “GVGrid: A QoS routing protocol for vehicular ad hoc networks,” in *Proc. IEEE Int. Workshop Qual. Service*, 2006, pp. 130–139.

[24] F. De Rango, F. Veltri, P. Fazio, and S. Marano, “Two-level trajectory-based routing protocol for vehicular ad hoc networks in freeway and Manhattan environments,” *J. Netw.*, vol. 4, no. 9, pp. 866–880, 2009.

[25] M. Ayaida, M. Barhoumi, H. Fouchal, Y. Ghamri-Doudane, and L. Afilal, “HHLS: A hybrid routing technique for VANETs,” in *Proc. IEEE Global Commun. Conf.*, 2012, pp. 44–48.

[26] W. Kiess, H. Fussler, J. Widmer, and M. Mauve, “Hierarchical location service for mobile ad-hoc networks,” *SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 8, no. 4, pp. 47–58, Oct. 2004.

[27] Y. Luo, W. Zhang, and Y. Hu, “A new cluster based routing protocol for VANET,” in *Proc. 2nd Int. Conf. Netw. Secur. Wireless Commun. Trusted Comput.*, 2010, vol. 1, pp. 176–180.

[28] 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,” *J. Inf. Sci. Eng.*, vol. 26, no. 3, pp. 817–832, 2010.

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

[30] M. Dorigo and L. Gambardella, “Ant-Q: A reinforcement learning approach to the traveling salesman problem,” in *Proc. 12th Int. Conf. Mach. Learn.*, 2016, pp. 252–260.

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

[32] M. Littman and J. Boyan, “Reinforcement learning scheme for network routing,” in *Proc. Int. Workshop Appl. Neural Netw. Telecommun*, 2013.

[33] Y. Goldberg, R. Song, and M. R. Kosorok, “Adaptive Q-learning,” *From Prob. Statist. Back, High-Dimensional Models Processes*, vol. 9, pp. 150–162, 2013.

[34] W. Celimuge and K. Kumekawa, “Distributed reinforcement learning approach for vehicular ad hoc networks,” *IEICE Trans. Commun.*, vol. 93, no. 6, pp. 1431–1442, 2010.

[35] B.-C. Seet, G. Liu, B.-S. Lee, C.-H. Foh, and K.-K. Lee, "A-STAR: A mobile ad hoc routing strategy for metropolis vehicular communications," in *Proc. Int. Conf. Res. Netw.*, 2004, pp. 989–999.

[36] J. Luo, X. Gu, T. Zhao, and W. Yan, "A mobile infrastructure based VANET routing protocol in the urban environment," in *Proc. Int. Conf. Commun. Mobile Comput.*, 2010, vol. 3, pp. 432–437.

[37] F. Zhang, H. Liu, Y.-W. Leung, X. Chu, and B. Jin, "Community-based bus system as routing backbone for vehicular ad hoc networks," in *Proc. IEEE 35th Int. Conf. Distrib. Comput. Syst.*, 2015, pp. 73–82.

[38] F. Zhang, H. Liu, Y. W. Leung, X. Chu, and B. Jin, "CBS: Community-based bus system as routing backbone for vehicular ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 8, pp. 2132–2146, Aug. 2017.

[39] L. Zhang, B. Yu, and J. Pan, "GeoMob: A mobility-aware geocast scheme in metropolitans via taxicabs and buses," in *Proc. IEEE Conf. Comput. Commun.*, 2014, pp. 1279–1787.

[40] *SUVnet-Trace Data*, Wireless and Sensor networks Lab (WnSN), Shanghai Jiao Tong University, 2007. [Online]. Available: http://wirelesslab.sjtu.edu.cn

[41] E. Wang, Y. Yang, B. Jia, and T. Guo, "The DTN routing algorithm based on Markov meeting time span prediction model," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 9, 2013.

[42] Y.-K. Ip, W.-C. Lau, and O.-C. Yue, "Forwarding and replication strategies for DTN with resource constraints," in *Proc. IEEE Veh. Technol. Conf.*, 2007, pp. 1260–1264.

[43] S. Liu, F. Li, Q. Zhang, and M. Shen, "A Markov chain prediction model for routing in delay tolerant networks," in *Proc. Int. Conf. Big Data Comput. Commun.*, 2015, pp. 479–490.

Fan Li received the Ph.D. degree in computer science from the University of North Carolina at Charlotte, Charlotte, NC, USA, in 2008, the M.Eng. degree in electrical engineering from the University of Delaware, Newark, DE, USA, in 2004, and the M.Eng. and B.Eng. degrees in communications and information system from the Huazhong University of Science and Technology, Wuhan, China, in 2001 and 1998, respectively. She is currently a Professor with the School of Computer Science, Beijing Institute of Technology, Beijing, China. Her current research focuses on wireless networks, ad hoc and sensor networks, and mobile computing. Her papers have won Best Paper Awards from IEEE MASS (2013), IEEE IPCCC (2013), ACM MobiHoc (2014), and Tsinghua Science and Technology (2015). She is a Member of the ACM and the IEEE.

Xiaoyu Song received the B.E. degree from the School of Computer Science, Zhengzhou University of Light Industry, Zhengzhou, China, in 2013. She is currently working toward the Ph.D. degree with the School of Computer Science, Beijing Institute of Technology, Beijing, China. Her research interests include delay tolerant network, vehicular ad hoc network, and mobile computing.

Huijie Chen received the B.E. degree from the School of Computer Science, Henan University of Economics and Law, Zhengzhou, China, in 2010 and the M.S. degree from the School of Computer Science, Taiyuan University of Science and Technology, Taiyuan, China, in 2013. She is currently working toward the Ph.D. degree with the School of Computer Science, Beijing Institute of Technology, Beijing, China. His research interests include delay tolerant network, vehicular ad hoc network, and mobile computing.

Xin Li received the B.Sc. and M.Sc. degrees in computer science from Jilin University, Changchun, China, and the Ph.D. degree in computer science from Hong Kong Baptist University, Hong Kong. She is currently an Associate Professor with the School of Computer Science, Beijing Institute of Technology, Beijing, China. Her research focuses on the development of algorithms for representation learning, reasoning under uncertainty and machine learning with application to information networks, vehicular networks, and recommender systems.

Yu Wang (F'18) received the B.Eng. and M.Eng. degrees in computer science from Tsinghua University, Beijing, China, and the Ph.D. degree in computer science from the Illinois Institute of Technology, Chicago, IL, USA. He is a Professor of computer science with the University of North Carolina at Charlotte, Charlotte, NC, USA. His research interest includes wireless networks, mobile social networks, smart sensing, mobile computing, and algorithm design. His research has been continuously supported by federal agencies including US National Science Foundation, US Department of Transportation, and National Natural Science Foundation of China (NSFC). He has authored and coauthored more than 150 papers in peer reviewed journals and conferences, with four Best Paper awards. He has served as the Editorial Board Member of several international journals, including IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS. He was the recipient of Ralph E. Powe Junior Faculty Enhancement awards from Oak Ridge Associated Universities (2006), Outstanding Faculty Research Award from College of Computing and Informatics at UNC Charlotte (2008), and Overseas Young Scholars Cooperation Research Fund from NSFC (2014). He is a Senior Member of the ACM.