Data-driven Traffic Flow Analysis for Vehicular Communications

Yang Wang[†,*],Liusheng Huang[†], Tianbo Gu[†,△], Hao Wei[†,◇], Kai Xing[†], Junshan Zhang[‡]
[†]School of Computer Science and Technology, University of Science and Technology of China
[‡]School of Electrical, Computer and Energy Engineering, Arizona State University
[△]School of Computer Science, University of Massachusetts at Amherst
[◇]Dept. of Systems Engineering and Engineering Management, The Chinese University of Hong Kong
Email:{angyan*,lshuang,kxing}@ustc.edu.cn, tbgu@cs.umass.edu, hwei@se.cuhk.edu.hk, junshan.zhang@asu.edu

***Abstract*—Due to high mobility and frequent disconnections in a vehicular network, reliable and efficient vehicular communication is very challenging. Previous studies focus on predicting the trajectories of single vehicles. Due to many random factors, however, there is little regularity in the movements of a single vehicle in an urban area, and this motivates us to take a holistic network perspective. With this insight, we model the time varying regularities of road traffic flows in road segments and intersections by mining statistic trajectories of all vehicles in the network. Based on these regularities and local real-time traffic information, we propose a new method to calculate the expected transfer delay from a current position to a given destination. We also propose a method to collect updated destination information. By combining the above two methods, we design a routing algorithm for vehicle-to-vehicle data transmission in vehicular networks, and then prove that it is a linear-time algorithm. Finally, we evaluate our algorithm by using information of real taxi vehicles. The results show that the performance of our algorithm is significantly better than other solutions in terms of packet delay.**

I. Introduction

With the dramatically increasing traffic load around the world, much attention is being paid to vehicular communications, aiming to avoid accidents or alleviate congestions in advance and to provide comfortable driving environments for drivers. Notably, vehicular networks are used for monitoring vehicles and road traffic, and providing vehicle-vehicle or vehicle-roadside communications. In particular, vehicular networks are being developed for active driving safety [3], [4], [16], intelligent transportation [17], [23], and entertainment [3], [6], which rely heavily on the quality of data delivery by vehicular networks.

Data delivery protocols in vehicular networks have attracted significant attention during the past years, and many methods have been proposed, including algorithms based on bus networks [19], trajectories of vehicles [5], [9], [10], [12], [14], [22], network connectivity [11], [20], and geography or road intersections [1], [7], [13], [18], [21], [25]. Despite of the recent studies, reliable data delivery is still not well understood in real-world vehicular networks. Recent algorithms mostly focus on the predictability of vehicles' trajectories. However, in the real world, the movement of an individual vehicle (e.g., a taxi) can be influenced by many uncertain factors, such as the driver, passenger, traffic status and weather, so it is difficult for people to predict the trajectory of single vehicle accurately. Further, packet delivery to moving destinations is particularly challenging for vehicular networks. Worth noting is that existing results on this problem are mostly based on opportunistic routing.

To address these issues, this paper proposes a new algorithm based on data-driven traffic flow analysis and destination collection mechanisms. Specifically, by mining the massive trajectory datasets of 3997 taxis in the main urban area of Suzhou over the year 2012, our extensive experiments reveal that the traffic flows of the road network exhibit time varying patterns, and that the traffic flow of a road segment or an intersection is closely related to its location and the time of the day. With this insight, we model the traffic flow of a road segment and that of an intersection by a Markov model. In addition, to achieve reliable data delivery to moving destinations, we design a new method by exchanging metadata of destinations upon meeting. In this way, the expected transfer delay can be quantified more accurately from the current packet carrier to the updated destination position, thereby enabling the selection of the optimal path.

The proposed algorithm is distributed and needs one hop information only. For performance evaluation, we study the classic algorithms including VADD [12] and TBD [9], and compare our algorithm with them on real taxi datasets. The results demonstrate the superior performance of our algorithm in terms of packet delay.

The main contributions of this paper are summarized below:

- We carry out data-driven traffic flow analysis of the road network to characterize the time varying patterns, and devise Markov models for the patterns accordingly.
- We design a method to calculate the expected delay of packet delivery from a specific position to another in vehicular networks.
- Under the restriction that a node can only obtain one hop neighbor information, we design a new method to obtain the updated information of moving destinations.
- We propose a routing algorithm and prove it is a linear-time algorithm.

The rest of this paper is organized as follows. After discussing related work, we formalize the problem in section III. In section IV, we analyze the patterns of the traffic flows and propose a model. Section V describes the data delivery algorithm. In Section VI, we discuss the convergence property of the algorithm, and prove its time complexity strictly. Numerical studies are presented in Section VII and conclusion of this paper is given in Section VIII.

II. Related Work

There has been a significant body of work on data transmission by taking advantage of trajectory prediction of single vehicles, including VADD [12], TBD [9], DMC [14] and TSF [10] to name a few .

VADD is among the earliest algorithms that exploit the predicted mobilities of vehicles for packet delivery. To address the issues of high mobility and frequent disconnections of vehicular networks, VADD adopts the idea of carry-and-forward. Simply put, to calculate the expected delay of delivering a packet to a fixed destination, a critical step in VADD is to compute the packet's expected delay in a road segment and its forwarding probability in an intersection. Based on the results and the mobility patterns, VADD then determines the corresponding action. Finally, VADD presents the idea of sending the query data back to a moving vehicle when the trajectory of the moving vehicle is determined by its destination. Compared to VADD, the TBD algorithm presented a more accurate link delay model of a road segment that computes the expected forwarding distance. Built on that, the TBD algorithm provides a more accurate mechanism for computing the expected end-to-end delivery delay under the condition that the packet carrier moves along the predicted individual trajectory. DMC uses a mobile pattern that only considers the inter-meeting times following an exponential distribution to predict the current location of a vehicle, and then proposes a greedy routing algorithm to moving destinations based on localized information and predicted information. TSF uses the moving destination vehicle's trajectory effectively to calculate an optimal meeting point of the packet and the destination vehicle, and to deliver packets reliably from the access points to a moving destination vehicle with a minimum delay. In summary, existing algorithms are based on the assumption that the future trajectory of a vehicle is predictable.

III. Problem Formulation

A. Basic Settings

Considering the road network in the urban area of Suzhou, this paper aims to find the path with minimum delivery delay from one mobile car to another. The basic setting is outlined as follows:

- **Fact 1:** All vehicles can move independently in the urban area of Suzhou which covers 695 square kilometers. There are 3997 taxis with GPS-based digital map navigation and monitoring systems in Suzhou, China.
- **Fact 2:** Each vehicle can upload its current position to the server by its equipped GPRS or WCDMA device, and this is the case for all taxis in Suzhou, China. The vehicles are divided into four groups, and only one group can upload their position data in one specific round. The round interval is 30 seconds, which means each vehicle can upload its data to server once in every 2 minutes. Every packet includes the information about vehicle ID, sample time, longitude, latitude, speed, direction and occupied status.
- **Fact 3:** The source and the destination of a packet are both vehicles. Each vehicle knows the current position and the historical trajectory of itself.
- **Assumption 1:** Each vehicle has a short range communication system and participates in a vehicular network where the communication range of the wireless device is R. The communication device can be a reliable realtime communication device [15] or a Dedicated Short Range Communications (DSRC) device [2].
- **Assumption 2:** Each intersection in the road network can communicate with vehicles nearby, and the communication range of an intersection is also R. Each vehicle knows the one hop information about all road segments related to the intersection which the vehicle can communicate with.

B. Vehicular Network Model

The vehicular network, consisting of all taxis, is denoted as

$$VN = \{n_0, n_1, n_2, \cdots, n_{|VN|-1}\}, \tag{1}$$

where n_i corresponds to the i-th node in the network.

Denote the location of node n_i at time t_x as $p_{n_i}(t_x)$. It follows that the collected location sequence of node n_i, corresponding to its trajectory from time t_0 to t_x, is given as:

$$TR_{n_i} = \{p_{n_i}(t_0), p_{n_i}(t_1), \cdots, p_{n_i}(t_x)\} \tag{2}$$

Any two nodes in the vehicular network n_i and n_j, can communicate with each other only when the Euclidean distance between them $ED_{n_i n_j}$ is less than R, which we can also phrase as "vehicle n_i and n_j meet each other". Therefore all possible links at time t_x can be denoted as:

$$TL(t_x) = \{(n_i, n_j) | ED_{n_i n_j}(t_x) \leq R; \forall n_i, n_j \in VN\} \tag{3}$$

All the vehicles move independently in the road network RN. The road network is comprised of intersections and road segments between adjacent intersections. Denote an intersection as I_i and the road segment between intersection I_i and I_j as r_{ij}.

After generating RN, we say that a node n_i in VN arrives at an intersection I_j if they can communicate with each other, i.e., the Euclidean distance between them $ED_{n_i I_j}$ is less than R. It follows that vehicles arriving at intersections at time t_x, is given by:

$$AI(t_x) = \{(n_i, I_j) | ED_{n_i I_j} \leq R; \forall\ n_i \in VN \& \forall\ I_j \in RN\} \tag{4}$$

Without loss of generality, consider the delivery of one packet P in VN, where the source of P is n_S and the destination is n_D, and $n_S, n_D \in VN$. The packet starts from the source at time t_0 and reaches the destination at time t_x. Thus, the delivery route of this packet can be written as:

$$DR_P = \{p_{n_S}(t_0), I_{i_1}, r_{i_1 i_2}, I_{i_2}, r_{i_2 i_3}, \cdots, I_{i_{y-1}}, r_{i_{y-1} i_y}, I_{i_y}, p_{n_D}(t_x)\} \tag{5}$$

C. Problem Formulation

A primary goal of the data transmission in vehicular networks is to deliver packets to their mobile destinations as soon as possible. Assume the all packets are given a fixed lifetime limit, achieving a lower average transmission delay is intimately related to achieving a higher average delivery ratio. In this paper, we will focus on the transmission delay of one single packet firstly, and then quantify the costs.

The transmission delay of one single packet P is the total time for delivering P from its source to destination via VN. It is obvious that the delay of transferring a packet via a road segment r_{ij} can be greatly influenced by its current traffic status. Suppose the transfer delay in one specific road segment r_{ij} at time t is $td_{ij}(t)$ and the delivery route of packet P is DR_P. Then, we can write the total delay of P while the transfer starts at time t as:

$$TD_P(t) = \sum_{x=1}^{y-1} td_{[i_x][i_{x+1}]}(t + \sum_{z=0}^{x-1} td_{[i_z][i_{z+1}]}) \tag{6}$$

Having presented the basic problem settings, we aim to design a proper routing strategy to minimize the packet transfer delay:

$$\min \quad TD_p(t) \tag{7}$$

IV. Analysis of Vehicle trajectory

It is clear that the knowledge of current and historical positions of vehicles can facilitate transferring packets more efficiently and accurately in vehicular networks. As mentioned earlier, existing algorithms focus on predicting the movements of a single vehicle. Due to many random factors, however, there is little regularity in the movements of a single vehicle in an urban area, and this motivates us to take a holistic network perspective. With this insight, we turn our attention to model the time varying regularities of road traffic flows in road segments and intersections by mining statistic trajectories of all vehicles in the network.

A. Analysis of Trajectory Prediction for Single Vehicle

In existing studies a common used assumption is that there exist spatio-temporal patterns in the movements of a single vehicle. However, as one can observe, the future trajectory of one individual vehicle, especially a taxi, can be influenced by many uncertain factors. For instance, the destination of a taxi is determined by its passenger, so the destination of a taxi is random. Even if the destination of two passenger is the same, the trajectory of a taxi is probably different because of different traffic status and drivers. Some early studies [8] show that it is difficult to propose a model for the real trajectory of a single vehicle, and then predict it. As an alternative, it is more plausible to incorporate road traffic flow information.

B. Analysis of Road traffic Flows

In related work on urban computing [24], it has been concluded that the traffic flows in a city have obvious relationships with people's living and travelling patterns, which can be influenced by time and position. For example, during work days, most people go to work at about 7:00am to 9:00am and go back home at 4:30pm to 7:00pm, so the traffic flows from residence to workplaces during 7:00am to 9:00am and from workplaces to residence during 4:30pm to 7:00pm in one work day are similar to another work day. This kind of patterns also exist between some other functional zones such as residence and commercial districts, depending on whether it is a work day or a weekend. These patterns are decided by the function layouts of city, so the traffic flows of one specific road segment or intersection have obvious time varying regularities.

For one road segment r_{ij}, denote the number of vehicles and speed at time t in the kth day of year 2012 by $s_{r_{ij}}(k,t)$ and $v_{r_{ij}}(k,t)$ respectively. Then, we can calculate the expectation of the vehicle number at time t depending on whether it is a work day or a rest day as $E[s_{r_{ij}}(t)]_W$, $E[s_{r_{ij}}(t)]_R$, and the same applies to the speed $E[v_{r_{ij}}(t)]_W$, $E[v_{r_{ij}}(t)]_R$. The time interval of updating the expectations is 2 minutes, because all vehicles update their position information once every 2 minutes. The period of these two parameters is 24 hours and the source data comes from all 366 days of year 2012.

$$\begin{cases} E[s_{r_{ij}}(t)]_W = \frac{\sum_{k \in work\ days\ of\ 2012} s_{r_{ij}}(k,t)}{|work\ days\ of\ 2012|} \\ E[s_{r_{ij}}(t)]_R = \frac{\sum_{k \in rest\ days\ of\ 2012} s_{r_{ij}}(k,t)}{|rest\ days\ of\ 2012|} \end{cases} \tag{8}$$

$$\begin{cases} E[v_{r_{ij}}(t)]_W = \frac{\sum_{k \in work\ days\ of\ 2012} v_{r_{ij}}(k,t)}{|work\ days\ of\ 2012|} \\ E[v_{r_{ij}}(t)]_R = \frac{\sum_{k \in rest\ days\ of\ 2012} v_{r_{ij}}(k,t)}{|rest\ days\ of\ 2012|} \end{cases} \tag{9}$$

For one intersection I_i, suppose it is an m-way intersection. It is of great interest to find the probabilities of the moving vehicles turning from one direction to another $m-1$ directions at a specific time t, which can help to calculate the probabilities of forwarding packets to each direction in this intersection at that time. Assume the adjacent intersections of intersection I_i are $N(i) = \{I_{j_1}, I_{j_2}, \cdots, I_{j_m}\}$, and the roads between them are $r_{ij_1}, r_{ij_2}, \cdots, r_{ij_m}$. Based on the trajectories of all taxis in 2012, we can quantify the probability of one vehicle coming from r_{ij_x}, crossing intersection I_i, and then leaving by r_{ij_y} at time t, which can be denoted as $TP_{i(j_x j_y)}(t)$. At intersection I_i, each vehicle has $m-1$ turning options and each road can get vehicles from other $m-1$ directions, thus we can formulate the turning-probabilities of intersection I_i at time t by a Markov matrix:

$$M_{TP_i}(t) = \begin{pmatrix} 0 & TP_{i(j_1 j_2)}(t) & \cdots & TP_{i(j_1 j_m)}(t) \\ TP_{i(j_2 j_1)}(t) & 0 & \cdots & TP_{i(j_2 j_m)}(t) \\ \vdots & \vdots & \ddots & \vdots \\ TP_{i(j_m j_1)}(t) & TP_{i(j_m j_2)}(t) & \cdots & 0 \end{pmatrix} \tag{10}$$

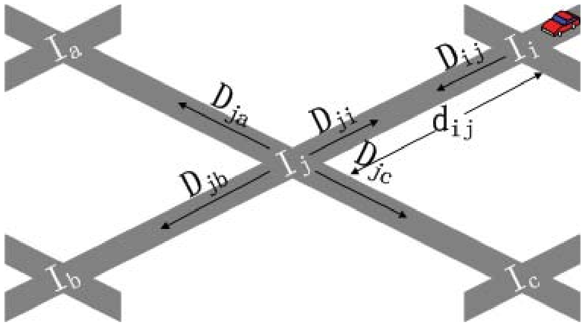

Fig. 1: An illustration of the delay model

Each column of the matrix shows the probabilities of a vehicle turning to its corresponding road segment from the other $m-1$ directions. The xth row of the matrix shows the probabilities of a vehicle from road r_{jx_i} turning to the other $m-1$ directions, which satisfy that:

$$\sum_{y=1}^{m} TP_{i(j_x j_y)} = 1 \tag{11}$$

So far, we have demonstrated the uncertainty of predicting.

V. V2V DATA TRANSMISSION BASED ON TRAFFIC FLOW ANALYSIS

In this section, we propose a routing algorithm by analyzing the traffic flows of road networks. There are three main steps: 1) For the case where the destination is partially known, find an efficient and accurate way to decide the transfer direction and select a path; 2) Develop online schemes for updating the destination information; 3) Design the algorithm.

A. Delay Model Based on Road Traffic Flows

For calculating the excepted packet delay from one specific intersection to the destination, we resort to a very successful and typical method used in VADD [12] and TBD [9]. On the basis of their results, we will revisit the expected delay evaluation method by using the road traffic flow information. The delay model is the basis of routing, and it will have crucial impact on the performance of the algorithm.

As shown in Figure 1, one packet at intersection I_i will be sent to its destination. Let $D_{ij}(t)$ denote the expected delay of transferring this packet from intersection I_i, through intersection I_j, to the destination at time t. We can calculate $D_{ij}(t)$ in a recursive manner:

$$D_{ij}(t) = td_{ij}(t) + \sum_{k\in N(j)} DP_{jk}(t+td_{ij}(t)) \times D_{jk}(t+td_{ij}(t)) \tag{12}$$

where $DP_{jk}(t)$ is the probability of a packet leaving intersection I_j for intersection I_k at time t, whether it is carried or forwarded.

Next, we will give an example to illustrate how to calculate the expected delay of $D_{ij}(t)$. As shown in figure 2, we calculate the expected delay from intersection 1 to intersection 3 by crossing intersection 2. So we can calculate delay according to equation (12) as followed.

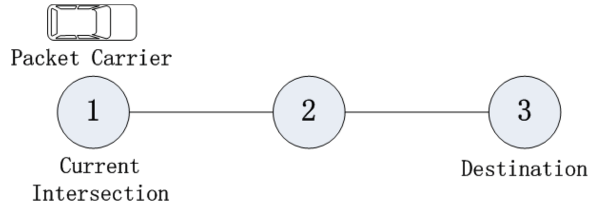

Fig. 2: An example of calculating expected delay

$$\begin{cases} D_{12}(t) = td_{12}(t) + DP_{23}[t+td_{12}(t)] * D_{23}[t+td_{12}(t)] \\ \qquad + DP_{21}[t+td_{12}(t)] * D_{21}[t+td_{12}(t)] \\ D_{23}[t+td_{12}(t)] = td_{23}[t+td_{12}(t)] \\ D_{21}[t+td_{12}(t)] = td_{21}[t+td_{12}(t)] \\ \qquad + DP_{12}[t+td_{12}(t)+td_{21}(t+td_{12}(t))] \\ \qquad * D_{12}[t+td_{12}(t)+td_{21}(t+td_{12}(t))] \\ \qquad \vdots \end{cases}$$

If D_{12}, D_{23} and D_{21} are fixed, it would be simple to solve the above equations and get D_{12} which corresponds to the expected delay of travelling from intersection 1, through intersection 2 and finally arriving at the destination which is intersection 3. Unfortunately, in a time varying setting $D_{12}(t) \neq D_{12}[t+td_{12}(t)+td_{21}(t+td_{12}(t))]$, indicating it is impossible to get $D_{12}(t)$ by solving the linear equations. At the same time, we observe that the average statuses of traffic flows do not change rapidly during the delivery period of one packet. Instead, we can approximate equation (12) as:

$$D_{ij}(t) = td_{ij}(t) + \sum_{k\in N(j)} DP_{jk}(t) \times D_{jk}(t) \tag{13}$$

According to (13), we can quantify each outgoing road segment of every intersection, and get an $n \times n$ linear equation system. Equation (13) gives an explicit way to calculate the expected delay of a packet, and it has already been proposed by existing studies. Nevertheless, there are still some key issues that are ignored before when calculating equation (13). In particular, it remains unclear how to estimate the parameters td_{ij} and DP_{jk} by traffic information.

1) Calculating td_{ij} With Road Traffic Flows: For one road segment r_{ij}, the expected transfer delay is relevant to the current traffic situation, so td_{ij} is also a time varying parameter, and we can denote the expected delay of road segment r_{ij} at time t as $td_{ij}(t)$. Given the following information:

- l_{ij}: the length of road segment r_{ij}, which is a constant.
- $s_{ij}(t)$: the vehicle number in r_{ij} at time t.
- $v_{ij}(t)$: the vehicle speed in r_{ij} at time t.
- α and β: weighting coefficients. They are two constants which can make $td_{ij}(t)$ more rational.

The expected transfer delay can be calculated as below:

$$td_{ij}(t) = \begin{cases} \alpha \times l_{ij} & if \frac{l_{ij}}{s_{ij}(t)} \leq R \\ \frac{l_{ij}}{v_{ij}(t)} - \beta \times \frac{s_{ij}(t)}{l_{ij}} & if \frac{l_{ij}}{s_{ij}(t)} > R \end{cases} \tag{14}$$

Equation (14) calculates $td_{ij}(t)$ in two different cases. In one road segment, if the vehicles density is big enough then all packet transmissions are forwarding; otherwise vehicles have to carry data packets sometimes.

2) Calculating DP_{ij} With Road Traffic Flows: For one intersection, the delivering probabilities are closely related to current traffic flows as well. For example, if there is no vehicle on one direction and no vehicle will turn to that direction, the delivering probability of that direction must be 0. When calculating DP_{ij}, we continue to use the following assumptions: the adjacent intersections of intersection I_i are $N(i) = \{I_{j_1}, I_{j_2}, \cdots, I_{j_m}\}$, and the roads between them are $r_{ij_1}, r_{ij_2}, \cdots, r_{ij_m}$. Firstly, For a packet at intersection I_i, we can count the contacting probability $CP_{ij_x}(t)$ of forwarding this packet to intersection I_{j_x} via r_{ij_x} at time t. The $CP_{ij_x}(t)$ can also be regarded as the probability of contacting at least one vehicle which is going towards road r_{ij_x}, and we can calculate it using the Poisson Process probability as:

$$\begin{aligned} CP_{ij_x}(t) &= P(N(\Delta T) \geq 1) \\ &= 1 - P(N(\Delta T) = 0) \\ &= 1 - e^{-\lambda_{ij_x}(t)\Delta T} \times \frac{(\lambda_{ij_x}(t)\Delta T)^0}{0!} \\ &= 1 - e^{-\lambda_{ij_x(t)}\Delta T} \end{aligned} \tag{15}$$

In equation (15), ΔT is the duration that a vehicle can communicate with other vehicles around an intersection. For simplification, we count ΔT as $2R/v$, and v is a constant speed value. $\lambda_{ij_x}(t)$ is the average rate of contact vehicles leaving intersection I_i and going towards r_{ij_x} at time t, and they are composed by vehicles coming from other $m-1$ directions. For one road r_{ij_y}, all vehicles on the road will leave it at most in $l_{ij_y}/v_{ij_y}(t)$, so the leaving rate of a vehicle at r_{ij_y} is $\frac{s_{ij_y}(t)}{l_{ij_y}/v_{ij_y}(t)}$. For all cars leaving r_{ij_y}, $TP_{i(j_y j_x)}(t)$ of them will turn towards road r_{ij_x}. In calculating $\lambda_{ij_x}(t)$, all parameters we need to confirm are $s_{ij_y}(t)$,$v_{ij_y}(t)$ and $TP_{i(j_y j_x)}(t)$. For $s_{ij_y}(t)$ and $v_{ij_y}(t)$, if the position is next to the current intersection, they can use the realtime information, otherwise they can only use historical information. For $TP_{i(j_y j_x)}(t)$, it can be obtained from the time varying Markov matrix of corresponding time, day and day type directly. Now we can calculate $\lambda_{ij_x}(t)$ as:

$$\lambda_{ij_x}(t) = \sum_{y=1, y\neq x}^{m} \frac{s_{ij_y}(t) \times TP_{i(j_y j_x)}(t) \times v_{ij_y}(t)}{l_{ij_y}} \tag{16}$$

Suppose the m intersections adjacent to I_i are sorted by the length of their shortest path to destination, the probability $DP^*_{ij_x}(t)$ of a packet being forwarded to r_{ij_x} at time t is:

$$DP^*_{ij_x}(t) = \begin{cases} CP_{ij_1}(t) & if\ x = 1 \\ [\prod_{y=1}^{x-1}(1 - CP_{ij_y}(t))]CP_{ij_x}(t) & if\ x = 2 \ldots m \end{cases} \tag{17}$$

If the packet carrier goes along r_{ij_y}, it is obvious that the packet will never be forwarded to those directions whose expected delay is bigger than road r_{ij_y}. So we can define the probability of a packet being forwarded to r_{ij_x} while its carrier goes along r_{ij_y} at time t:

$$DP_{ij_x|ij_y}(t) = \begin{cases} DP^*_{ij_x}(t) & if\ x < y \\ 1 - \sum_{z=1}^{x-1} DP^*_{ij_z}(t) & if\ x = y \\ 0 & if\ x > y \end{cases} \tag{18}$$

Assume the possibility of a packet carrier turning to r_{ij_y} at time t is $LP_{ij_y}(t)$ while it leaves intersection I_i, we have:

$$LP_{ij_y}(t) = \frac{\sum_{z=1}^{m} TP_{i(j_z i_y)}(t) * s_{ij_z}(t)}{\sum_{z=1}^{m} s_{ij_z}(t)} \tag{19}$$

$$DP_{ij_x}(t) = \sum_{j_y \in N(i)} LP_{ij_y}(t) * DP_{ij_x|ij_y}(t). \tag{20}$$

3) Calculating D_{ij} With Road Traffic Flows: At this point, the key steps of calculating the expected delay have been presented above. When calculating $D_{ij}(t)$, only the vehicle number and average speed of those road segments that are next to the current intersection of the packet carrier are realtime, and the rest information comes from mining the historical traffic information. For example, given a specific road network as illustrated in Figure 2, $D_{ij}(t)$ is calculated as below:

$$\begin{cases} D_{12}(t) = td_{12}(t) + DP_{23}(t) * D_{23}(t) + DP_{21}(t) * D_{21}(t) \\ D_{23}(t) = td_{23}(t) \\ D_{21}(t) = td_{21}(t) + DP_{12}(t) * D_{12}(t) \end{cases}$$

The parameters left to compute are $td_{12}(t)$, $td_{21}(t)$, $td_{23}(t)$ and $DP_{12}(t)$, $DP_{21}(t)$, $DP_{23}(t)$. Given the one hop realtime information of intersection 1, which are $s_{12}(t)$ and $v_{12}(t)$ of road segment r_{12} at current time t, we can calculate $td_{12}(t)$ and $td_{21}(t)$ with $s_{12}(t)$ and $v_{12}(t)$, calculate $DP_{12}(t)$ and $DP_{21}(t)$ with both realtime information and historical information, calculate $td_{23}(t)$ and $DP_{23}(t)$ with historical traffic flow information.

B. Online Updating Information of Moving Destinations

In the pervious subsection we focus on how to calculate the expected transfer delay by analyzing traffic flow information. However, in the data-driven vehicular communication problem, the destination is moving, so it is impossible to know the exact location of the destination in advance. In this section, we will study how to obtain the approximate location of the destination to help delivering.

Here, given the assumption that vehicles can only obtain one hop realtime information, it is almost impossible for a packet carrier to know the exact position of its destination. Instead, it is more feasible to approximate the location of the destination. The destination vehicle is moving in road networks, and it can meet other vehicles frequently. Similarly, the current packet carrier is also moving in road networks and can also meet other vehicles frequently. Therefore there are some vehicles having met both the destination and the packet carrier. If these vehicles can all memorize when and where the encounters between themselves and the destination happen, and pass the information to the packet carrier once they meet, then the packet carrier can obtain the updated information about the destination which can help our transmission. Here we summarize the principles of the approximate information collection mechanism:

- **Rule 1:** Suppose any vehicle n_i will maintain a metadata for all the other $|VN| - 1$ vehicles. In the dataset maintained by the vehicle n_i, the metadata of vehicle $n_k(i \neq k)$ stores the latest time, position, speed and direction information of n_k which n_i can obtain. If we denote the metadata of n_k maintained by n_i as $MD_{n_k}^{n_i}$, the metadata set of n_i can be expressed as $\{MD_{n_k}^{n_i}, k \in [1, |VN|] \& i \neq k\}$. Each metadata in the set has a lifetime limit L_t, and a metadata expires once its lifetime gets over the limit.

- **Rule 2:** Each intersection in the road network RN will also maintain a metadata for each vehicle in VN. With the same format as the metadata kept in vehicles. We can denote the latest metadata of n_k maintained by intersection I_i as $MD_{n_k}^{I_i}$, and the metadata set of intersection I_i is $\{MD_{n_k}^{I_i}, k \in [1, |VN|]\}$. This kind of metadata also has a lifetime limit L_t.
- **Rule 3:** Once a vehicle n_i meets an intersection I_j, firstly the intersection will update the metadata of vehicle n_i, and then they will compare the metadata of other vehicles and divide all metadata stored in n_i and I_j into three cases. 1) both n_i and I_j maintain the metadata of n_k; 2) metadata of n_k is only stored in n_i; 3) metadata of n_k is only stored in I_j. In the first case, n_i should compare the metadata of n_k maintained by itself with I_j and find out which one is newer, then the data is updated accordingly. In the second case, we will copy the metadata of n_k from n_i to I_j and I_j will get a newer metadata of n_k. In the last case, we will copy the metadata of n_k from I_j to n_i and n_i will get a newer metadata of n_k.
- **Rule 4:** Once a vehicle n_i meets another vehicle n_j, firstly they will update the metadata of the other one maintained by itself, and then they will compare their metadata set with each other. After that, the same operations like rule 3 will be executed so as to keep all metadata sets in vehicles as new as possible.

Given the above rules, if we focus on the information of one specific vehicle n_k, and then we can analyze the propagation of this kind of information at an intersection or in a road segment as follows: While a vehicle n_i with the newest information of n_k arrives at intersection I_j, this information must be transferred to the intersection and it will be broadcasted to other directions of intersection I_j by vehicles towards those directions. While a vehicle n_i with the newest information of n_k builds a communication link with vehicle n_j, it will forward this information to n_j immediately. Thus, the realtime position of the destination will be spread as fast as possible.

C. V2V Communication Algorithm

The overall algorithm consists of two parts. The first part is a centralized algorithm that analyzes the historical traffic flow to get average speeds and vehicle numbers of all the road segments and the Markov matrices of all the intersections at the time slices throughout one year. This part only needs to be executed once and the results will be sent to all the vehicles before any data packet transmission begins. The second part is the V2V communication algorithm, which is a distributed algorithm to be executed in different vehicles with one hop realtime information.

The V2V communication algorithm consists of two stages, the first stage involves approximately collecting information of the moving destination, the second stage is to calculate the expected delay based on traffic flows and the approximate location information collected in the first stage. The communication algorithm can be triggered by two types of events, one is a vehicle meets another vehicle, and the other is a vehicle arrives at an intersection.

Once a vehicle meets another vehicle, the algorithm will be executed immediately. During its first stage, this two vehicles will exchange their metadata set with each other, so as to keep them as new as possible. During its second stage, suppose one of them is a packet carrier, if they have the same moving direction and the other vehicle is ahead of the current carrier, the current carrier must forward the packet to the other vehicle.

Once a vehicle arrives at an intersection, firstly the vehicle and the intersection will also exchange their metadata set according to the updated information collection mechanism during the first stage. During the second stage, if the vehicle is a packet carrier, it should calculate the expected delays of different directions from the current position to the currently known position of destination which is collected in the first stage and select the path with the minimum expected delay to transfer the packet. Here is a special case: If a packet carrier don't have the metadata of the destination, it will keep the packet until it obtains the metadata of the destination.

The pseudo code of the distributed algorithm is shown in Algorithm 1. All parameters in Algorithm 1 are defined above. Once a vehicle meets another vehicle, the algorithm will be triggered and Procedure "Vehicle Meets Vehicle" will be executed. Once a vehicle reach an intersection, the algorithm will be triggered and Procedure "Vehicle Reach Intersection" will be executed.

VI. Performance Analysis

In this section, we will discuss the convergence property of the algorithm firstly, and then analyze its time complexity.

Proposition 1. *The algorithm converges in the sense that the packet will be delivered to the destination.*

Proof:

Firstly, a packet can be forwarded or carried, and the speed of forwarding v_f is significantly faster than the speed of carrying v_c, so we have $v_f \gg v_c$. Secondly, according to the number of vehicles and the road network, the vehicle density is relatively big and a packet have high probability to be forwarded. As a result, a packet travels faster than a vehicle in general, we can denote it as $v_p > v_v$, where v_p means the average speed of delivering packets and v_v means the average speed of moving vehicles. Suppose the distance between a packet and its corresponding destination is L, the packet carrier has the position of its destination before Δt, so we have $\Delta t = \frac{L}{v_s}$, where v_s is the speed of the destination's position information been spread, and $v_s \geq v_p$. Because $v_p > v_v$, so the distance L between them keeps decreasing and the Δt

Algorithm 1 The V2V Communication Algorithm

procedure VEHICLEMEETSVEHICLE(t, n_i, n_j)
  if $(n_i, n_j) \in TL(t)$ **then**
    Renew metadata by n_i *and* n_j
    $n_i.metadata\ set \leftrightarrow n_j.metadata\ set$
    if $n_i(n_j)$ *is a packet carrier* **then**
      if $n_j(n_i) = Destination$ **then**
        Send Packet to $n_j(n_i)$
        Transmission Succeed
      else if $(n_i.direction = n_j.direction)$ and $(n_j(n_i)$ *is ahead of* $n_i(n_j))$ **then**
        Send Packet to $n_j(n_i)$
      end if
    end if
  end if
end procedure
procedure VEHICLEREACHINTERSECTION(t, n_i, I_j)
  if $(n_i, I_j) \in AI(t)$ **then**
    $n_i.metadata\ set \leftrightarrow I_j.metadata\ set$
    if n_i *is a packet carrier* **then**
      $Mini\ Delay = \infty$
      if $n_i.metadata.destination \neq \emptyset$ **then**
        for all $I_k \in N(j)$ **do**
          Calculate $D_{jk}(t)$ *to Destination*
          if $D_{jk}(t) < Mini\ Delay$ **then**
            $Mini\ Delay \leftarrow D_{jk}(t)$
            $Next\ Route \leftarrow r_{jk}$
          end if
        end for
      end if
    end if
  end if
  Transfer Packet to Next Route
end procedure

also keeps decreasing, so the accuracy of the transmission will increase. Because the distance L keeps decreasing every step and running time is enough, so the packet will definitely reach its destination. Thus, the algorithm is convergent. ■

Proposition 2. *The time complexity of the algorithm is $O(L)$ where L is the initial Euclidean distance between the source and the destination of a packet.*

Proof:

Suppose the percentage of forwarding in all transmission is δ, the average speed of carrying is v, and the average speed of forwarding is $kv(k \gg 1)$, thus the average speed of packet delivery is $v + (k-1)\delta v$.

A road network can be treated as a Manhattan grid, that is to say, given two positions, the distance between them through the road network is no more than $\sqrt{2}$ times the Euclidean distance between them. Here we suppose a packet carrier knows the realtime position of its destination, and then we can simplify the delivery problem to the pursuing problem in a flat. In our algorithm, a packet is always transferred to the

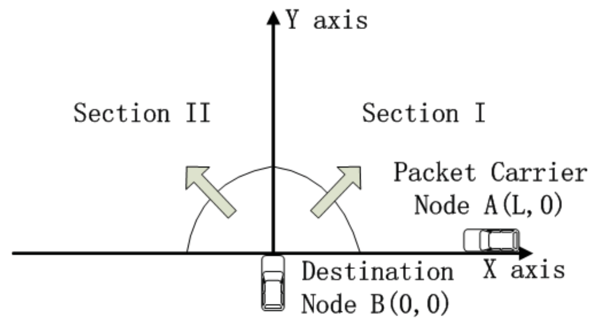

Fig. 3: The scenario of pursuing

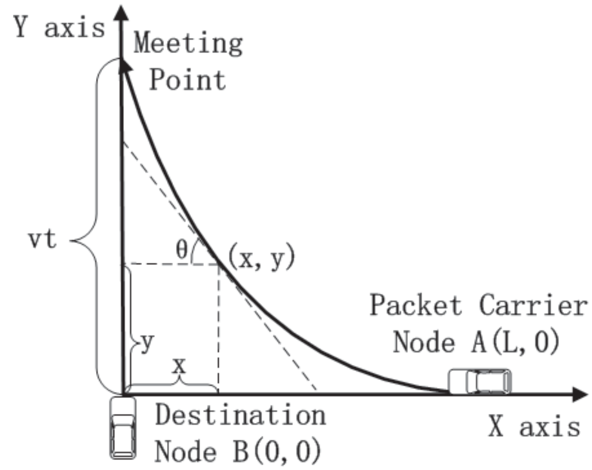

Fig. 4: The schematic of pursuing

path with minimum delay, so after the simplification, we can assume a pursuer always keeps running directly to the realtime position of its destination. Now the problem can be written as: All nodes are moving in a flat, node A knows the real time position of node B, the speed of A is $v + (k-1)\delta v$ and the speed of B is v. The initial Euclidean distance between them is L. Suppose the initial position of B is the origin of the coordinate system, the initial position of A is $(L, 0)$ as illustrated in Figure 3. If B moves in the negative direction of X axis, the time cost is maximal. On the contrary, if B moves in the positive direction of X axis, the time cost is minimal. Suppose if B moves in the positive direction of Y axis, the time cost is T, then if B moves to section I, the time cost must be less than T and if B moves to section II, the time cost must be more than T. Section I and II are symmetric, so we can regard T as the average time cost for A to catch B. Now, our problem is to calculate the time cost T. Figure 4 shows the principle of catching. Suppose the position of A at time t is (x, y), so the tangent line of the catching trajectory which crosses (x, y) certainly meets Y axis at $(0, vt)$. Regarding the curve, we have the following ordinary differential equations:

$$\begin{cases} \frac{dy}{dx} = \tan\theta = \frac{vt-y}{x} \\ \left[\frac{dy}{dt}\right]^2 + \left[\frac{dx}{dt}\right]^2 = [v + (k-1)\delta v]^2 \end{cases} \quad (21)$$

By consolidating and simplifying the equation system (21), we can write the curve equations as:

$$\begin{cases} y = \frac{1}{2\sqrt[q]{L}}(1+\frac{1}{q})x^{(1+\frac{1}{q})} - \frac{\sqrt[q]{L}}{2}(1-\frac{1}{q})x^{(1-\frac{1}{q})} + \frac{q}{q^2-1}L \\ q = 1+(k-1)\delta \end{cases} \tag{22}$$

In equation (22), if $x = 0$, then $y = \frac{1+(k-1)\delta}{(k-1)^2\delta^2+2(k-1)\delta}L$. So A and B can meet each other at $(0, \frac{1+(k-1)\delta}{(k-1)^2\delta^2+2(k-1)\delta}L)$, and the meeting time is $\frac{1+(k-1)\delta}{[(k-1)^2\delta^2+2(k-1)\delta]v}L$.

Note that, A can only obtain the position of B before Δt, and Δt keeps decreasing with the distance between them. The vehicle density is pretty big so we can calculate the initial value of Δt as L/kv. Suppose A always tries to catch the position of B before L/kv, this problem is similar to above. After obtaining the position of B before L/kv, A tries to catch the realtime position of B along Y axis and the time cost is $\frac{L}{[(k-1)^2\delta^2+(k-1)\delta]v}$. It is clear that the practical time for A to catch B must be less than above, so the realtime cost must be less than $\frac{[1+(k-1)\delta]L}{[(k-1)^2\delta^2+2(k-1)\delta]v} + \frac{L}{[(k-1)^2\delta^2+(k-1)\delta]v}$. Back to real vehicular networks, the moving distances of A and B will increase for no more than $\sqrt{2}$ times, so the increasement of the time cost must be no more than a constant C times. Thus, the average time cost for delivering a packet is:

$$t_{A\rightarrow B} \leq \left\{ \begin{array}{c} \frac{[1+(k-1)\delta]}{[(k-1)^2\delta^2+2(k-1)\delta]v} \\ +\frac{1}{[(k-1)^2\delta^2+(k-1)\delta]v} \end{array} \right\} * C * L \tag{23}$$

In equation (23), k, δ, v and C are all fixed constants, so the time complexity of the algorithm is $O(L)$, which means it is a linear time complexity algorithm. ■

VII. NUMERICAL STUDIES

In this section, we evaluate the performance of our algorithm of Vehicle-to-Vehicle Routing(V2VR) by comparing with VADD [12] and TBD [9] which is also based on calculating the expected end-to-end delay. Before evaluation, we improve VADD and TBD by giving them the ability to obtain updated information of moving destination. The numerical investigation involves the following:

We use the average delivery delay and the percentage of vehicles who have the information of destination as the performance metric. We investigate (i) the impact of workdays vs. weekends, (ii) the impact of time period and (iii) the effect of the updated destination information.

The total number of packets generated for one experiment is 500 and the simulation is continued until all packets are transferred to destination. Without loss of generality, the system parameters are set based on the classical DSRC scenario. Unless otherwise specified, the default values are set according to Table I or as defined in Section III.

Label Parameter	Label Description
Road Network	1402 intersections/4332 road segments
Communication Range	R = 200 meters
Number of Vehicles	All 3997 taxis in Suzhou
Lifetime of Packet	∞
Vehicle Speed	Real data from monitoring
Vehicle Monitor Interval	2 minutes

TABLE I: Simulation Setup

A. Impact of Day Type

The day type determines the patterns of traffic flows in the road networks. In this subsection, we investigate the influence of the day type. Figure 5 shows the average delivery delay of the V2VR, VADD and TBD in different day types. As shown in Figure 5, the average delay of V2VR in workdays and weekends is significantly lower than the average delay of both VADD and TBD in corresponding day types. Another very interesting observation is the difference between the average delay of V2VR in two different day types is much smaller than that of the other two algorithms. This is because V2VR uses time varying models to approximate the mobile patterns of road traffics, so its performance is more stable.

B. Impact of Time Period

In this section, we are interested with the performance of the algorithms during different time periods in one day. Figure 6 and 7 shows the average delay during 9:00am to 17:00pm in workdays and weekends. As shown in Figure 6 and 7, the average delay of V2VR is lower than the other two algorithms most of the time, and the lines of TBD and VADD are much sharper, which proves again that V2VR is more stable benefiting from the data-driven traffic flow analysis. Moreover, in Figure 7, the lines of TBD and VADD are more zigzag because the predicted trajectories of vehicles which can help delivery are mined from historical information of all days, and it models workdays better for most of days are workdays.

C. Effect of The Online Destination Information Collection Mechanism

In this section, we evaluate the effect of our online destination information collection mechanism. As shown in Figure 8, the percentage of vehicles who have the updated information of destination is considerably high and different traffic flows during different times in a day can rarely influence it.

VIII. CONCLUSION

In this paper, we carry out data-driven analysis to characterize the patterns of the traffic flows. A method is designed to calculate the expected delay of transferring packets by taking advantage of the analysis. Further, an approach is proposed to obtain the updated information of the moving destination, and a distributed vehicular communication algorithm is presented. Performance evaluation confirms that the performance of our algorithm is better than the state-of-art solutions. With the increasing popularity of vehicular networks, the proposed algorithm offers a new perspective to use vehicle trajectory information and deliver packets efficiently.

ACKNOWLEDGMENT

We gratefully acknowledge Prof. Xiangyang Li from Illinois Institute of Technology for valuable comments and suggestions on draft versions of this paper. And also thank anonymous reviewers for reading this paper and giving valuable comments. This paper is supported by the NSFC (No.61379131),(No.U1301256), and Jiangsu Natural Science Foundation (No.BK2011359).

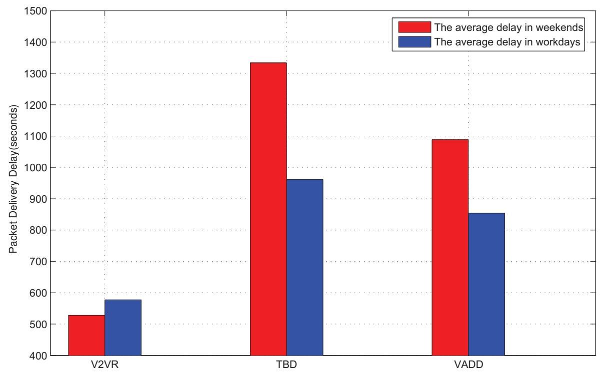

Fig. 5: Impact of workdays vs. weekends

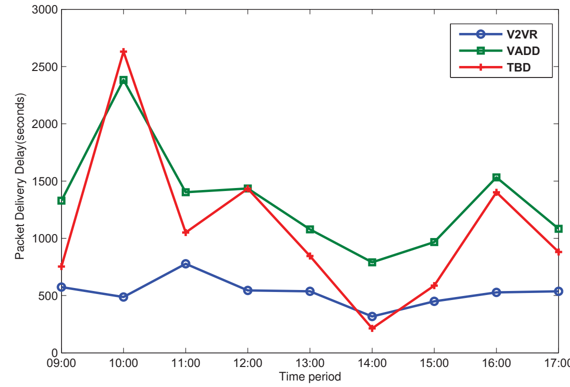

Fig. 6: Impact of time period in workdays

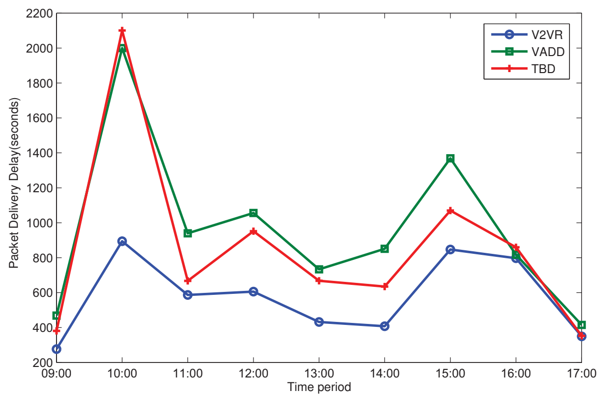

Fig. 7: Impact of time period in weekends

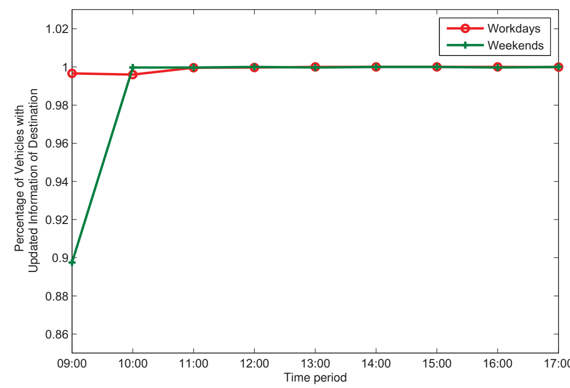

Fig. 8: Effect of online updating destination information

REFERENCES

[1] Geographic routing in city scenarios. *ACM SIGMOBILE*, 9(1):69–72, 2005.

[2] A. Carter. The status of vehicle-to-vehicle communication as a means of improving crash prevention performance. Tech. Rep. 05-0264, National Highway Traffic Safety Administration United States of America, 2005.

[3] CE. Palazzi, M. Roccetti and S. Ferretti. An intervehicular communication architecture for safety and entertainment. *IEEE Transactions on Intelligent Transportation Systems*, 11(1):90–99, September 2009.

[4] D. Jiang, V. Taliwal, A. Meier, W. Holfelder and R. Herrtwich. Design of 5.9ghz dsrc-based vehicular safety communication. *Wireless Communications, IEEE*, 13(5):36–43, 2006.

[5] J. Jeong Y. Gu Q. Cao M. Liu F. Xu, S. Guo and T. He. Utilizing shared vehicle trajectories for data forwarding in vehicular networks. In *Proc. IEEE INFOCOM*, pages 441–445, 2011.

[6] H. Hartenstein, B. Bochow, A. Ebner, M. Lott, M. Radimirsch and D. Vollmer. Position-aware ad hoc wireless networks for inter-vehicle communications: the fleetnet project. In *Proc. ACM Mobihoc*, pages 259–262, New York, NY, USA, 2001.

[7] I. Leontiadis and C. Mascolo. Geopps: geographical opportunistic routing for vehicular networks. In *Proc. WoWMoM*, pages 1–6, 2007.

[8] F. Filali J. Harri and C. Bonnet. Mobility models for vehicular ad hoc networks: A survey and taxonomy. *Communications Survey and Tutorials, IEEE*, 11(4):19–41, 2009.

[9] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du. Tbd:trajectory-based data forwarding for light-traffic vehicular networks. In *Proc. IEEE ICDCS*, pages 231–238, 2009.

[10] J. Jeong, S. Guo, Y. Gu, T. He, and D. H. C. Du. Tsf: Trajectory-based statistical forwarding for infrastructure-to-vehicle data delivery in vehicular networks. In *Proc. IEEE ICDCS*, pages 557–566, 2010.

[11] J. Nzouonta, N. Rajgure, W. Guiling, and C. Borcea. Vanet routing on city roads using real-time vehicular traffic information. *IEEE Transactions on Vehicular Technology*, 58(7):3609–3626, 2009.

[12] J. Zhao and G. Cao. Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks. In *Proc. IEEE INFOCOM*, pages 1–12, 2006.

[13] 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 and Engineering*, 26(3):817–832, 2010.

[14] K. Lee, Y. Yi, J. Jeong, H. Won, I. Rhee and S. Chong. Max-contribution: On optimal resource allocation in delay tolerant networks. In *Proc. ACM SIGMOBILE*, 2007.

[15] K. Xing, T. Gu, Z.G. Zhao, L. Shi, Y.H. Liu, P.F. Hu, Y.P. Wang, Y. Liang, S. Zhang, Y. Wang and L.S. Huang. Approaching reliable realtime communications: A novel system design and implementation for roadway safety oriented vehicular communications. In *Proc. IEEE INFOCOM*, pages 1188–1210, 2013.

[16] L. Chisalita and N. Shahmehri. A peer-to-peer approach to vehicular communication for the support of traffic safety applications. In *Proc. IEEE ITSC*, pages 336–341, Singapore, Sept 2002.

[17] P. Papadimitratos, A. La Fortelle, K. Evenssen and R. Brignolo. Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. *Communications Magazine, IEEE*, 47(11):84–95, November 2009.

[18] R. Lu, X. Lin, and X. Shen. Spring: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks. In *Proc. IEEE INFOCOM*, pages 632–640, 2010.

[19] U. Acer, P. Giaccone, D. Hay, G. Neglia, and S. Tarapiah. Timely data delivery in a realistic bus network. In *Proc. IEEE INFOCOM*, pages 446–450, 2011.

[20] V. Naumov and T. R. Gross. Connectivity-aware routing (car) in vehicular ad-hoc networks. In *Proc. IEEE INFOCOM*, pages 1919–1927, 2007.

[21] X. Lin, R. Lu, X. Liang, and X. S. Shen. Stap: A social-tier-assisted packet forwarding protocol for achieving receiver-location privacy preservation in vanets. In *Proc. IEEE INFOCOM*, pages 2147–2155, 2011.

[22] Y. Wu, Y. Zhu, and B. Li. Trajectory improves data delivery in vehicular networks. In *Proc. IEEE INFOCOM*, pages 2183–2191, 2011.

[23] Y. Zhao. *Vehicle Location and Navigation Systems*. Artech House Publishers, 1997.

[24] Y. Zheng, Y.C. Liu, J. Yuan and X. Xie. Urban computing with taxicabs. In *Proc. ACM Ubicomp*, 2011.

[25] Y.S. Chen, Y.W. Lin, and C.Y. Pan. Dir:diagonal-intersection-based routing protocol for vehicular ad hoc networks. *Telecommunication Systems*, 46(4):299–316, 2011.