Delay-Optimal Data Forwarding in Vehicular Sensor Networks

Okyoung Choi†, Seokhyun Kim†, Jaeseong Jeong†, Hyang-Won Lee‡ and Song Chong†
†Department of Electrical Engineering, KAIST
‡Department of Internet and Multimedia Engineering, Konkuk University
Email: {okyoung,kimseokhyun,jsjeong}@netsys.kaist.ac.kr, leehw@konkuk.ac.kr, songchong@kaist.edu

***Abstract*—Vehicular Sensor Network (VSN) is emerging as a new solution for monitoring urban environments such as Intelligent Transportation Systems and air pollution. One of the crucial factors that determine the service quality of urban monitoring applications is the delivery delay of sensing data packets in the VSN. In this paper, we study the problem of routing data packets with minimum delay in the VSN, by exploiting i) vehicle traffic statistics, ii) anycast routing and iii) knowledge of future trajectories of vehicles such as buses. We first introduce a novel road network graph model that incorporates the three factors into the routing metric. We then characterize the packet delay on each edge as a function of the vehicle density, speed and the length of the edge. Based on the network model and delay function, we formulate the packet routing problem as a Markov Decision Process (MDP) and develop an optimal routing policy by solving the MDP. Evaluations using real vehicle traces in a city show that our routing policy significantly improves the delay performance compared to existing routing protocols.**

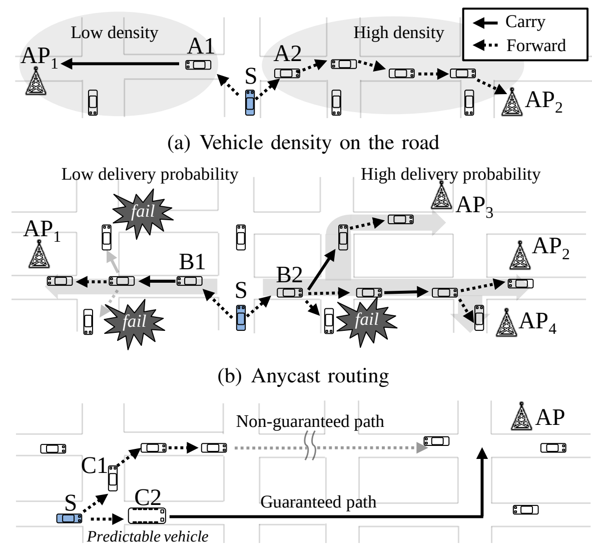
I. Introduction

Recently, Vehicular Sensor Networks (VSNs) have received a great amount of attention as a new solution for monitoring the physical world [9]. In VSNs, vehicles equipped with sensing devices move around an urban area and sense the urban environment periodically. The vehicles use vehicle to vehicle (V2V) or vehicle to infra (V2I) wireless communications to deliver the sensing data to an urban monitoring center. Hence, unlike the traditional sensing system with fixed sensors that experiences limited coverage, the vehicular sensing system can monitor any area where vehicles can reach. Moreover, the vehicular sensor network can be deployed and maintained with relatively low cost since it does not heavily rely on the network infrastructure for sensing data delivery.

Many of the VSN applications such as Intelligent Transportation System (ITS) require frequent updates of sensing information from all over the urban area, and hence it is important to guarantee *timely delivery* of sensing data from *every area* of interest to the urban monitoring center. Such a *coverage guarantee* is rather challenging in VSNs where the links (and thus the routes to destinations) can come and go depending on the mobility of vehicles. For instance, in such a network with

This research was supported by the KCC(Korea Communications Commission), Korea, under the R&D program supervised by the KCA Communications Agency)(KCA-2013- (11-911-05-004))

Hyang-Won Lee was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2012R1A1A1012610)



(a) Vehicle density on the road

(b) Anycast routing

(c) Knowledge of future trajectories

Fig. 1. Important factors on the delay performance in Vehicular Sensor Networks: APs are destinations in anycast routing

intermittent connectivity, a vehicle sometimes has to carry the data while it moves away from the destination. In fact, Delay-Tolerant Networks (DTNs) similarly experience intermittent routes to destinations, and there has been a large body of work that addresses the problem of routing data packets with minimum delay in DTNs [2], [8], [11]. Due to the similarity, the packet routing policies for DTNs could be used for VSNs as well. However, the VSN is distinguished from general DTNs in several aspects. First, vehicles in VSNs only move along the road, whereas mobile nodes in general DTNs are typically assumed to be able to move arbitrarily. Second, VSNs generally adopt anycast with multiple destinations, whereas most of the works in general DTNs assume unicast. Third, there are vehicles with predetermined future trajectories, such as buses, whereas in general DTNs, it is hard to predict the movement of mobile nodes. Therefore, the packet routing policies for DTNs may not be directly applicable to VSNs, or may not be able to fully exploit the characteristics of VSNs. In this paper, we study the packet routing problem in the VSN with anycast.

In particular, we focus on minimizing the packet delivery delay from every area of interest to the urban monitoring center. It is obvious that a packet routing algorithm with minimum delay must take into account the aforementioned characteristics of VSNs. First, since the vehicles can move only along the road, the vehicle density can be different from road to road. Clearly, the road with high density can provide more opportunities of wireless multi-hop transfers, and thus reduce the delivery delay on the road. Consider a source vehicle S in Fig. 1(a), which tries to select a better relay out of vehicles $A1$ and $A2$. Even though $A1$ is closer to a destination (or AP in Fig. 1(a)), forwarding to $A2$ may be more beneficial since the delay of multi-hop transfer over high density road is much smaller than carrying delay. Second, in anycast routing, a data packet just needs to be delivered to any one of the multiple destinations. Hence, the effect of multiple APs can be exploited to reduce the packet delay. As shown in Fig. 1(b), forwarding to $B1$ can fail to deliver packets to the targeted AP (*i.e.*, AP_1) due to the uncertainty in $B1$'s movement. However, since there exist many alternative APs on the direction of $B2$ (*i.e.*, AP_2, AP_3 and AP_4), forwarding to $B2$ may be a better option for reducing the delay. Third, the vehicles with known trajectories such as buses can help further reduce the delay. In Fig. 1(c) where S is far from the destination, such a predictable vehicle $C2$ guarantees to carry packets to the AP, which can significantly improve the routing performance compared to the delivery along a non-guaranteed path. Note that the effect of known future trajectories is greatly appreciated in the scenario where the vehicle density is relatively low.

There exist several routing schemes developed for VSNs by taking into account some of the above factors. In [18], the packet routing problem is formulated using a stochastic road network graph that models roadmap structure and traffic statistics such as vehicle density and speed on roads, and a routing algorithm is developed that outperforms other existing algorithms. Based on the road network model, [6] proposes a packet routing scheme that utilizes vehicle trajectory information (more details will be discussed in Section II).

Our goal in this paper is to develop a routing algorithm in VSNs that minimizes the packet delivery delay by taking into all of the above three factors. In particular, we extend the road network graph model in [18] to account for predetermined vehicle trajectories. This model enables to formulate the routing problem as a simple Markov Decision Process (MDP) that seeks to minimize the expected delay of a packet to one of the destinations. By solving the MDP, we develop an optimal packet routing algorithm that exploits all the factors in Fig. 1. Our contributions can be summarized as follows:

- We develop a new road network graph model that captures the effect of vehicles with predetermined trajectories (such as buses) and enables a simple formulation of the routing problem.
- We develop a delay-optimal routing algorithm that fully accounts for vehicle statistics, anycast routing and predetermined vehicle trajectories.

The rest of this paper is organized as follows: In Section II, we discuss related work. In Section III, we present the road network graph model of the VSN. In Section IV, we formulate the packet routing problem as an MDP, and develop an optimal routing policy that solves the MDP. In Section V, we evaluate the performance of our routing algorithm using real vehicle traces.

II. Related Work

There are a number of papers that study packet routing algorithms in VSNs [9], [10] and Vehicular Networks [6]–[8], [15], [16], [18]. Their common goal is to minimize the packet delivery delay to the destination. The existing routing algorithms can be classified into multi-copy schemes and single-copy schemes.

In multi-copy routing schemes [2], [5], [8], [13], packets are replicated and forwarded to have a better chance of reaching the destination. However, such a replication can result in heavy congestion, which in turn hinders the packets from reaching the destination. In single-copy routing schemes [6], [11], [16], [18], packets are not replicated, but instead, the characteristics of vehicular networks are better utilized for reducing the packet delivery delay. For instance, the Vehicle-Assisted Data Delivery (VADD) algorithm in [18] makes a routing decision based on the road layout, vehicle density and speed, and is shown to outperform the routing algorithms that do not utilize the characteristics of vehicular networks.

Trajectory-Based Data (TBD) forwarding scheme in [6] improves upon the VADD algorithm by taking into account predetermined trajectories. In particular, each vehicle is assumed to predetermine its path and utilizes the path for data forwarding. However, TBD does not fully exploit the knowledge of trajectories in that the routing option to carry data is given lower priority than the option to forward data to other vehicles. Furthermore, the trajectory of a vehicle is not shared among other vehicles, and thus each vehicle must compute its own routing policy. Shared-Trajectory-based Data Forwarding Scheme (STDFS) in [16] aims at better utilizing the trajectory information by assuming that each vehicle knows the (predetermined) trajectories of every other vehicle in the network. However, such a full sharing of information may be prohibitive in practice. All of thses works basically assume unicast, thus they cannot be directly applied to anycast routing or fully utilize the impact of the multiple destinations.

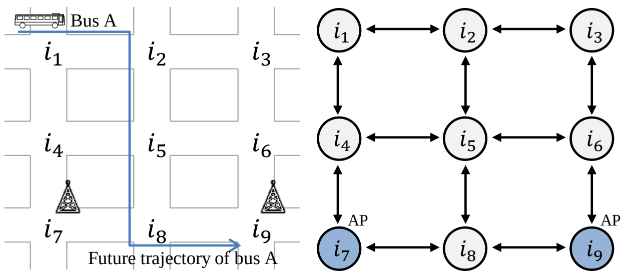
In this paper, we investigate the routing problem in VSNs by formulating a Markov Decision Process (MDP) that fully takes into account the effect of predetermined future trajectories and anycast routing as well as the vehicle density and speed.

III. System Model

Our vehicular sensor network is modeled as a "vehicular sensing system" working on an urban area or a "road network" described in the following.

A. Vehicular Sensing System

We consider a Vehicular Sensor Network (VSN) that consists of vehicles and WiFi Access Points (APs). Vehicles moving along the road sense the urban area, generate sensing data packets periodically, and deliver the packets to one of the APs by carrying or forwarding to others. The APs are deployed only at intersections, and connected to the urban monitoring



(a) Roadmap (b) Road network graph G

Fig. 2. Roadmap and its corresponding road network graph

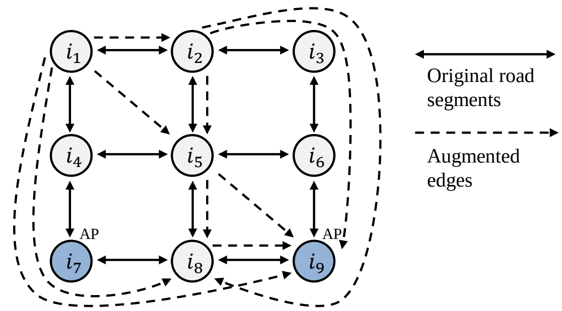


Fig. 3. Augmented road network graph G' incorporating bus line A

center via wired backhaul networks. Hence, the sensing data packets just need to be delivered to one of the APs. There are two types of vehicles including those with predetermined trajectories (such as buses and police patrol vehicles) and those with unpredictable trajectories (such as taxis and cars). For simplicity of exposition, a vehicle with predetermined trajectory will be called "bus" throughout the paper. As in a real city, we assume that a certain fraction of vehicles in the VSN are buses (i.e., vehicles with predetermined paths).

We assume that vehicles can use the digital roadmap and their GPS information, and are equipped with the IEEE 802.11 devices to communicate with other vehicles or the APs. We also assume that once a vehicle forwards a packet to another vehicle, the packet is immediately deleted from the sender vehicle; so that there is always at most one copy of each data packet in the network.

B. Road Network Graph

The urban area or road network to be sensed by vehicles is modeled as a graph $G = (\mathcal{I}, \mathcal{R})$ where $\mathcal{I}$ is the set of intersections and $\mathcal{R}$ is the set of road segments connecting the intersections. The network G is a directed graph, and hence, road segment $e_{ij} \in R$ denotes the road from intersection i to (neighboring) intersection j. Denote by $\mathcal{I}_{\mathcal{AP}} \subset \mathcal{I}$ the set of intersections where APs are placed. In our system, there are N vehicles in total, and we define $\mathcal{V} = \{0, 1, \ldots, M\}$ as the set of the types of vehicles. If the type $v \in \mathcal{V}$ of a vehicle is zero, its trajectories are unpredictable. Otherwise, if $v \neq 0$, then it represents a bus line with a predetermined route. Thus, M is the number of bus lines in the VSN.

Fig. 2 shows the roadmap and its corresponding directed graph G. In Fig. 2(a), two APs are placed at the intersections i_7, i_9 and the path of bus `A` is the sequence of intersections, i_1, i_2, i_5, i_8 and i_9. Note that the path of a packet is a sequence of consecutive road segments and intersections since the packets are carried and forwarded by vehicles moving along the road. Unlike the usual communication network where there is a fixed set of routes all the time, in the VSN, the links on a "data path" are formed by the mobility of vehicles, and thus, they do not always exist. Accordingly, the road network graph G represents a network that can be "potentially" used for delivering data packets, and the existence of data links in the network is highly uncertain. Therefore, due to the movement of vehicles the data packets are delivered as if they are routed over a random graph, and this will be accounted for in our formulation of the packet routing problem in Section IV.

To incorporate the effect of buses into the graph, we note that a bus can carry its packets not only to the neighbor intersections but also to every intersection along its future trajectory with 100% probability in a certain time. Hence, it is as if there is an edge directly connecting an intersection to another intersection which is multiple blocks away. To define these additional edges, we introduce a new notation e_{ij}^v representing the edge from i to j created by type-v vehicle. We denote the set of newly added edges by $\mathcal{L}$ and a new road network graph by $G' = (\mathcal{I}, \mathcal{R}')$ where $\mathcal{R}' = \mathcal{R} \cup \mathcal{L}$. Note that edge $e_{ij}^0 \in \mathcal{R}'$ is the same as $e_{ij} \in \mathcal{R}$ of graph G. Let $\mathcal{R}'_s$ be the set of edges in $\mathcal{R}'$ corresponding to a "single" road segment in $\mathcal{R}$, *i.e.*, $\mathcal{R}'_s = \{e_{ij}^v \in \mathcal{R}' : \exists e_{ij} \in \mathcal{R}\}$. Fig. 3 shows the new road graph G', which is augmented from G in Fig. 2(b) to take into account the effect of bus `A`'s predetermined path. Using the graph G', we formulate the delay-optimal packet routing problem in Section IV.

IV. Delay-Optimal Routing Algorithm

In this section, we develop a routing policy that minimizes the packet delay to any one of the APs. In particular, we formulate the packet routing problem as a Markov Decision Process (MDP) and find an optimal routing policy that solves the MDP.

A. Routing Algorithm Overview

As mentioned in Section III, packets are delivered by vehicles along the intersections and edges in the augmented road network graph G'. We assume that the routing policy is computed in advance using the vehicle traffic statistics, and the vehicles only have a routing table that can be used for forwarding packets. This would reduce the amount of online computations and thus enable fast forwarding of packets. Our routing algorithm specifies the forwarding decision at every intersection and edge as follows:

1) At intersections: Consider a vehicle arriving at an intersection, and assume that it has data packets. Clearly, the vehicle can forward its packets to a neighbor intersection if it meets another vehicle heading to the neighbor intersection or if it moves to the neighbor intersection. Thus, the packet delivery to a neighbor intersection is higly uncertain and totally depends on the existence of the vehicles heading to the intersection.

We take the idea in [18], which is to prioritize the outgoing edges of each intersection. Thus, if the vehicle does not either meet another vehicle along the edge with the first priority or move onto the edge, then it attempts packet forwarding toward the edge with the second priority, and so on. In Section IV-B, we develop a prioritization method (*i.e.*, routing policy) that minimizes the packet delay.

2) On edges: The packet forwarding on an edge, say e_{ij}^v, is divided into two cases. If j is a neighbor intersection of i in the original network graph G (*i.e.*, $e_{ij}^v \in \mathcal{R}'_s$), then a vehicle on e_{ij}^v forwards its packets to a vehicle closer to j. If j is not a neighbor intersection of i in G (*i.e.*, $e_{ij}^v \in \mathcal{R}' \setminus \mathcal{R}'_s$), then e_{ij}^v is an augmented edge by the bus with type v. On those edges, the bus with the corresponding type carries packets to j.

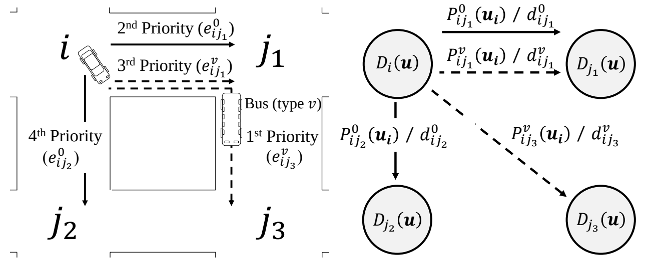
B. MDP Formulation and Optimal Routing Policy

Based on the expected data delivery delay model in [18], we formulate the routing problem as a Markov Decision Process (MDP) and develop a routing policy with minimum expected delivery delay. MDP can effectively capture the essence of the routing problem in VSNs where the delivery of a packet from an intersection to another is probabilistic, and the probability depends on the routing decision at the intersection.

The set of states in our MDP represents the set of intersections $\mathcal{I}$, and the transitions from one state to others occur probabilistically over the edges e_{ij}^v in $\mathcal{R}'$. Then, the control decision at each state in the MDP corresponds to a routing decision at each intersection. Note that the state transition probability from state i to j depends on the vehicle traffic statistics and the routing decision at intersection i. Denote by $\boldsymbol{u}_i$ a routing decision at intersection i. To account for the prioritization discussed above, $\boldsymbol{u}_i$ is defined as a row vector, $\boldsymbol{u}_i = [u_i^1 \quad u_i^2 \quad \dots \quad u_i^{K_i}]$, where $u_i^1, u_i^2, \dots, u_i^{K_i} \in \mathcal{R}'$ are all the outgoing edges from intersection i and K_i is the total number of outgoing edges from i. The order of elements in $\boldsymbol{u}_i$ represents the priority, that is, u_i^k indicates the k-th most preferred next hop from intersection i for minimum packet delay. Let $\mathcal{U}(i)$ be the set of all possible decisions $\boldsymbol{u}_i$ at intersection i, then the size of $\mathcal{U}(i)$ is given by $|\mathcal{U}(i)| = K_i!$.

As mentioned above, the routing decision $\boldsymbol{u}_i$ affects the data forwarding probability from intersection i to other intersections. Let $P_{ij}^v(\boldsymbol{u}_i)$ be the probability that a packet is forwarded from intersection i to j by a type-v vehicle under a routing decision $\boldsymbol{u}_i$. Denote by d_{ij}^v the expected data delay on an edge $e_{ij}^v \in \mathcal{L}$ which is the time it takes to carry and forward a packet along the edge e_{ij}^v. The delay d_{ij}^v can be estimated using the average vehicle speed, density and the total length of road segments from intersection i to j. We discuss the details of the estimation of d_{ij}^v in Section IV-D.

Under a routing policy $\boldsymbol{u} = [\boldsymbol{u}_i, \forall i \in \mathcal{I}]$, let $D_i(\boldsymbol{u})$ be the expected data delivery delay from intersection i to any AP. Thus, for an intersection i where an AP is placed (*i.e.*, $i \in \mathcal{I}_{\mathcal{AP}}$), $D_i(\boldsymbol{u}) = 0$ for any routing policy $\boldsymbol{u}$ and no action is taken at i. On the other hand, at an intersection i where there is no AP, $D_i(\boldsymbol{u})$ depends on d_{ij}^v, $D_j(\boldsymbol{u})$ and $P_{ij}^v(\boldsymbol{u}_i)$ for every outgoing edge e_{ij}^v from i. For better understanding, in Fig. 4, we illustrate an example of the routing decision $\boldsymbol{u}_i$ and related



(a) Outgoing edges at intersection i (b) Markov decision process (MDP) model of (a)

Fig. 4. Formulation of routing problem in graph G'

parameters. Assume that there exist four possible forwarding candidates which are prioritized by $\boldsymbol{u}_i$ as in Fig. 4(a). Based on the routing scenario, the MDP model in Fig. 4(b) has four outgoing edges which correspond to the forwarding candidates, and specifies the forwarding probability $P_{ij}^v(\boldsymbol{u}_i)$ and the edge delay d_{ij}^v. Clearly, $D_i(\boldsymbol{u})$ can be computed as

$$\begin{aligned} D_i(\boldsymbol{u}) =& P_{ij_1}^0 \times \left(d_{ij_1}^0 + D_{j_1}(\boldsymbol{u})\right) + P_{ij_2}^0 \times \left(d_{ij_2}^0 + D_{j_2}(\boldsymbol{u})\right) \\ &+ P_{ij_1}^v \times \left(d_{ij_1}^v + D_{j_1}(\boldsymbol{u})\right) + P_{ij_2}^v \times \left(d_{ij_2}^v + D_{j_2}(\boldsymbol{u})\right) \end{aligned} \quad (1)$$

In general, $D_i(\boldsymbol{u})$ can be expressed as follows:

$$D_i(\boldsymbol{u}) = \sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{I}} P_{ij}^v(\boldsymbol{u}_i) \cdot \left(d_{ij}^v + D_j(\boldsymbol{u})\right), \ i \in \mathcal{I} \setminus \mathcal{I}_{\mathcal{AP}}. \quad (2)$$

Hence, our routing problem can be formulated as

$$\min_{\boldsymbol{u}} D_i(\boldsymbol{u}), \forall i. \quad (3)$$

The optimal solution $\boldsymbol{u}^*$ to the above equation (3) gives a routing policy that minimizes the expected delay from i to any one of the APs. The routing problem can be solved using the value iteration method [4] (see Algorithm 1). For a given

Algorithm 1 Routing Policy Computation

Procedure ComputingOptimalPolicy($\boldsymbol{D}^0$)
  Input: initial value $\boldsymbol{D}^0 = [D_i^0, \forall i \in \mathcal{I}]$
  Output: optimal routing policy $\boldsymbol{u}^* = [\boldsymbol{u}_i^*, \forall i \in \mathcal{I}]$
  Local variable: $k = 0$
  repeat
    $$D_i^{k+1} = \min_{\boldsymbol{u}_i \in \mathcal{U}(i)} \sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{I}} P_{ij}^v(\boldsymbol{u}_i) \cdot (d_{ij}^v + D_j^k) \quad (4)$$
    $k = k + 1$
  until $\max_{i \in \mathcal{I}} |D_i^k - D_i^{k-1}| < \epsilon$
  $$\boldsymbol{u}_i^* = \arg\min_{\boldsymbol{u}_i \in \mathcal{U}(i)} \sum_{v \in \mathcal{V}} \sum_{j \in \mathcal{I}} P_{ij}^v(\boldsymbol{u}_i) \cdot \left(d_{ij}^v + D_j^k\right), \forall i \in \mathcal{I} \quad (5)$$
return $\boldsymbol{u}^*$

initial delay vector $\boldsymbol{D}^0$, the expected delay from intersection i is updated as in (4). The iteration is terminated if the two consecutive delay vectors $\boldsymbol{D}^k$ and $\boldsymbol{D}^{k-1}$ are close enough, *i.e.*,

$$\max_{i \in \mathcal{I}} |D_i^k - D_i^{k-1}| < \epsilon \quad (6)$$

where ϵ is a predetermined threshold value. It is known that for each i the sequence $\{D_i^k\}$ generated by the iteration in (4) converges close to its optimal value $D_i^* = D_i(\boldsymbol{u}^*)$ after a sufficient number of iterations [3]. The optimal routing policy $\boldsymbol{u}^* = [\boldsymbol{u}_i^*, \forall i \in \mathcal{I}]$ is then computed using the estimated optimal delay vector $\boldsymbol{D}^k = [D_i^k, \forall i \in \mathcal{I}]$, as in (5).

Remark: In our anycast setting, multiple APs are deployed at intersections, and each intersection i with an AP will have $D_i(\boldsymbol{u}) = 0$. Thus, the optimal routing policy solving the MDP would try to forward the packets toward one of the intersections with APs. Therefore, our routing policy can take advantage of multiple destinations in anycast routing.

Remark: Note that in [18] and [6], a heuristic algorithm is proposed to solve the problem (3). Specifically, given an initial routing policy $\boldsymbol{u}$, the fixed point equation (2) is solved to find the values of $D_i(u), \forall i$. Finally, a routing policy is computed by comparing the expected delay for each forwarding direction based on the computed D_i as well as d_{ij}. The initial routing policy used in [6], [18] is based on the geographical relationship between intersections and the AP. In Section V, we will show that in some scenarios, such heuristic methods can achieve significantly poor delay performance compared to our optimal solution.

C. Data Forwarding Probability $P_{ij}^v(\boldsymbol{u}_i)$

Next, we discuss how to calculate the data forwarding probability $P_{ij}^v(\boldsymbol{u}_i)$. Let $\mathcal{O}(i)$ be the set of outgoing edges from intersection i. At an intersection i where there is no AP, a vehicle forwards or carries packets to a neighbor intersection along one of the edges in $\mathcal{O}(i)$. Thus, $P_{ij}^v(\boldsymbol{u}_i)$ is a function of the probabilities Q_{ij}^v and C_{ij}^v defined as

– Q_{ij}^v: probability that a vehicle at i moves onto edge e_{ij}^v
– C_{ij}^v: probability of contacting a vehicle moving onto e_{ij}^v.

First, we find an expression for $P_{ij}^v(\boldsymbol{u}_i)$ in terms of Q_{ij}^v and C_{ij}^v, and then describe how to estimate Q_{ij}^v and C_{ij}^v using vehicular traffic statistics. This will provide a complete description of the computation of $P_{ij}^v(\boldsymbol{u}_i)$.

1) Computation of $P_{ij}^v(\boldsymbol{u}_i)$: Consider an event that packets at intersection i are forwarded to j through an edge e_{ij}^v under a routing decision $\boldsymbol{u}_i$. Clearly, this forwarding event can occur if a vehicle with the packets at i meets another vehicle moving onto e_{ij}^v or it moves onto e_{ij}^v. The additional condition for the forwarding event to occur is that a vehicle at i does not encounter vehicles moving onto the edges with higher priority than e_{ij}^v in $\boldsymbol{u}_i$ and it does not move onto those edges. Those conditions are illustrated by three events that are defined as

– $\boldsymbol{A}$: the event that a vehicle at i does not meet a vehicle moving onto the edges with higher priority than edge e_{ij}^v
– $\boldsymbol{B}$: the event that a vehicle at i meets another vehicle moving onto e_{ij}^v and it does not move onto the edges with higher priority than e_{ij}^v
– $\boldsymbol{C}$: the event that a vehicle moves onto e_{ij}^v.

$$P_{ij}^v(\boldsymbol{u}_i) = Pr\big[\boldsymbol{A} \cap (\boldsymbol{B} \cup \boldsymbol{C})\big] = Pr(\boldsymbol{A}) \times Pr(\boldsymbol{B} \cup \boldsymbol{C}) \tag{7}$$

where $Pr(\boldsymbol{E})$ is the probability of event $\boldsymbol{E}$. The equality follows from the fact that the moving direction of a vehicle is independent of that of others. Using Q_{ij}^v and C_{ij}^v, (7) can be rewritten as follows:

$$P_{ij}^v(\boldsymbol{u}_i) = \left[\prod_{e_{ik}^w \in \mathcal{H}(\boldsymbol{u}_i, e_{ij}^v)} (1 - C_{ik}^w)\right] \times \left[C_{ij}^v\Big(1 - \sum_{e_{ik}^w \in \mathcal{H}(\boldsymbol{u}_i, e_{ij}^v)} Q_{ik}^w\Big) + Q_{ij}^v - C_{ij}^v Q_{ij}^v\right] \tag{8}$$

where $\mathcal{H}(\boldsymbol{u}_i, e_{ij}^v)$ is the set of the edges which have higher priority than e_{ij}^v in a routing decision $\boldsymbol{u}_i$. The first product term in (8) corresponds to $Pr(\boldsymbol{A})$, *i.e.*, the probability that a vehicle at i does not meet a vehicle moving onto higher priority edges than the edge e_{ij}^v in routing decision $\boldsymbol{u}_i$. The second product term is equal to $Pr(\boldsymbol{B} \cup \boldsymbol{C})$, *i.e.*, the probability that a vehicle at i carries or forwards its packets onto edge e_{ij}^v.

2) Estimation of Q_{ij}^v and C_{ij}^v: Recall that Q_{ij}^v is the probability that a vehicle at intersection i moves onto edge e_{ij}^v, and C_{ij}^v is the probability of contacting a vehicle moving onto e_{ij}^v. Obviously, Q_{ij}^v and C_{ij}^v are determined by the parameters such as the vehicle density and moving tendency. In particular, the following parameters are used to express Q_{ij}^v and C_{ij}^v:

– q_{ij}^0: the fraction of type-0 vehicles moving to a neighbor intersection j among all vehicles which arrive to i.
– p_{ij}^0: the probability of meeting a type-0 vehicle at i that moves to j.
– q_i^v: the fraction of type-v vehicles among all vehicles which arrive to i,
– p_i^v: the probability of meeting a type-v vehicle at i.

Note that these parameters can be extracted from vehicle traffic statistics [18].

To compute Q_{ij}^v and C_{ij}^v, we consider two cases of the vehicle type v. First, for unpredictable vehicles (*i.e.*, $v = 0$), Q_{ij}^0 and C_{ij}^0 are estimated as q_{ij}^0 and p_{ij}^0, respectively. However, in the case of buses (*i.e.*, $v > 0$), estimating Q_{ij}^v and C_{ij}^v is more complicated because at intersection i, the outgoing edges created by type-v bus become either all available or all unavailable. Thus, if e_{ij}^v is the best edge among all augmented edges from type-v vehicles, Q_{ij}^v and C_{ij}^v for $v > 0$ are equal to q_i^v and p_i^v, respectively. Otherwise, Q_{ij}^v and C_{ij}^v are zero. To describe this as an equation, we introduce a new notation $>_{\boldsymbol{u}_i}$ such that $e_{ij}^v >_{\boldsymbol{u}_i} e_{ik}^w$ if e_{ij}^v has higher priority than e_{ik}^w under a routing decision $\boldsymbol{u}_i$. The following summarizes the computation of Q_{ij}^v and C_{ij}^v for all types of vehicles:

$$Q_{ij}^v(\boldsymbol{u}_i) = \begin{cases} q_{ij}^v & \text{if } v = 0 \\ q_i^v & \text{if } v > 0 \text{ and } e_{ij}^v >_{\boldsymbol{u}_i} e_{ik}^w, \\ & \forall e_{ik}^w \in \mathcal{O}(i) \text{ s.t. } w = v \\ 0 & \text{otherwise} \end{cases}$$
$$C_{ij}^v(\boldsymbol{u}_i) = \begin{cases} p_{ij}^v & \text{if } v = 0 \\ p_i^v & \text{if } v > 0 \text{ and } e_{ij}^v >_{\boldsymbol{u}_i} e_{ik}^w, \\ & \forall e_{ik}^w \in \mathcal{O}(i) \text{ s.t. } w = v \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

D. Expected Delay on Edges

Recall that d_{ij}^v is the expected data delay on an edge $e_{ij}^v \in \mathcal{L}$. The delay d_{ij}^v can be estimated using the average vehicle density and speed on e_{ij}^v and the length of e_{ij}^v, which can be easily obtained from the vehicle traffic statistics. Note that if e_{ij}^v corresponds to a set of multiple road segments in $\mathcal{R}$ (*i.e.*, $e_{ij}^v \in \mathcal{R}' \setminus \mathcal{R}'_s$), then on e_{ij}^v the corresponding bus "carries" the packets all the way to intersection j. On the other hand, if e_{ij}^v corresponds to a single road segment in $\mathcal{R}$ (*i.e.*, $e_{ij}^v \in \mathcal{R}'_s$), then V2V packet forwarding is allowed on e_{ij} as discussed in Section IV-A. Hence, the delay on the edge is estimated differently depending on the number of hops in the edge.

1) d_{ij}^v on $e_{ij}^v \in \mathcal{R}'_s$: Again, on this type of edges, a data packet is forwarded to another vehicle ahead. Clearly, this V2V forwarding can significantly reduce the delay. Thus, the delay d_{ij}^v depends on the vehicle density ρ_{ij} on e_{ij}^v and the WiFi transmission range R since if the density is high or the transmission range is long, then there is a high chance of V2V forwarding. These factors can be integrated in several ways. In this paper, we adopt the delay model in [18] as follows:

$$d_{ij}^v = (1 - e^{-R\cdot\rho_{ij}}) \cdot \frac{l_{ij} \cdot c}{R} + e^{-R\cdot\rho_{ij}} \cdot \frac{l_{ij}}{s_{ij}}, \text{ for } v = 0 \quad (10)$$

where l_{ij}, s_{ij} and c are the length of road segment e_{ij}, the average vehicle speed on e_{ij} and the wireless transmission delay, respectively. The first term in (10) is the expected delay contributed by V2V forwarding, and the second term is the expected carrying delay.

2) d_{ij}^v on e_{ij}^v in $\mathcal{R}' \setminus \mathcal{R}'_s$: In this case, packets are carried by the bus all the way. Let $\mathcal{B}(e_{ij}^v)$ be the set of road segments between intersection i and j along the route of type-v bus. The packet delay on e_{ij}^v depends only on the average speed of the bus with type v, denoted by s_{ij}^v, and the length of road segments in $\mathcal{B}(e_{ij}^v)$. Hence, the expected delay on edge e_{ij}^v can be estimated by the following equation:

$$d_{ij}^v = \sum_{e_{mn} \in \mathcal{B}(e_{ij}^v)} \frac{l_{mn}}{s_{mn}^v}, \text{ for } v > 0 \quad (11)$$

Remark: The delay function clearly shows that our routing algorithm would prefer to forward packets along the edges with high density and high average vehicle speed.

E. Computational Complexity of Algorithm 1

In Algorithm 1, the gap between D_i^k and its optimal value exponentially decrease as the number of iterations increase [4], and thus the iteration terminates in $c\log(1/\epsilon)$ where c is a bounded constant. Therefore, if each iteration can be completed in polynomial time, then Algorithm 1 is a polynomial time algorithm. Accordingly, we discuss the computational complexity of one iteration in Algorithms 1.

We discuss the computational complexity of each iteration. Denote by V and S the number of intersections and the size of action space (*i.e.*, the number of routing decisions) at an intersection. Then, it is clear that the minimization problem (4) can be solved in $O(SV^2)$ time. As explained in Section IV-B, S depends on the number of outgoing edges incident to a node. In the augmented road network graph, each type of bus gives $O(V)$ outgoing edges to each intersection, and thus each intersection has $O(V!)$ possible routing decisions. However, for each bus type v, only the edge with highest priority among the edges augmented by v matters because either all or none of those edges are available at a time. In other words, for an edge e_{ij}^v, $O((V-1)!)$ routing decisions lead to the exactly same delay D_i^{k+1}. Thus, each bus type essentially gives only $O(V)$ routing decisions at an intersection. Consequently, removing such redundant routing decisions, the size S of action space becomes $O(V^M)$ instead of $O(V!)$ where M is the number of bus types. Finally, the computational complexity of each iteration is $O(SV^{M+2})$, and hence, each iteration terminates in polynomial time of V.

It should be noted that the routing policy is computed offline, and thus this complexity can be tolerated in practice.

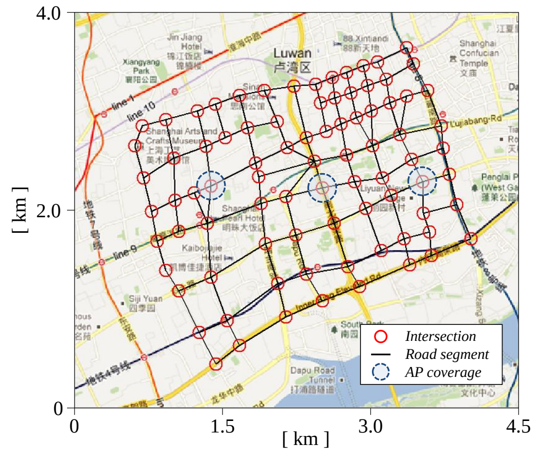


Fig. 5. Road network topology and AP deployment in Shanghai urban area

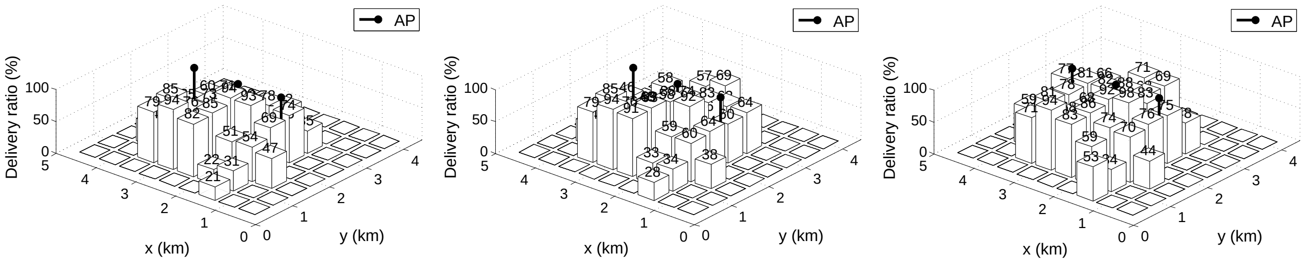
V. Performance Evaluation

In this section, we evaluate the performance of our routing algorithms based on real traces: Shanghai taxi [14] and bus [1] trace. The results show that our routing algorithms improve the delay performance of sensing data against the existing algorithm [6], [18].

A. Simulation Setup

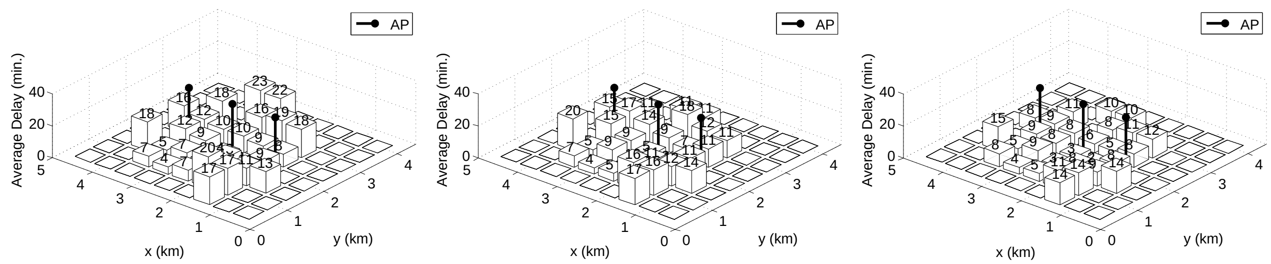
To verify our optimal routing algorithms, we use GPS traces of 4800 taxis [14] and 2300 buses [1] in Shanghai, where the location information of each vehicle is recorded at every 30 seconds in 30km×30km Shanghai for 28 days. To focus on the sensing scenario of downtown area, we select 4.5km×4km Shanghai downtown, which consists of 84 intersections and 112 road segments, as shown in Fig. 5. The selected area is modeled as a road network graph as discussed in Section III. Fig. 5 also shows 3 intersections where APs are placed (see the intersections with dotted circle). We choose 95 taxis and 35 buses (6 types of buses) which have relatively low GPS errors.

We implement the vehicular sensor network on a well-known wireless network simulator, GloMoSim [17], using 802.11a



(a) VADD (b) TBD (c) OVDF

Fig. 6. Data delivery ratio of three tested algorithms : (a) VADD, (b) TBD, (c) OVDF. In all cases, the number of vehicles is 110 (including taxis and 6 different types of buses).



(a) VADD (b) TBD (c) OVDF

Fig. 7. Average delivery delay of three tested algorithms : (a) VADD, (b) TBD, (c) OVDF. In all cases, the number of vehicles is 110 (including taxis and 6 different types of buses).

MAC layer protocol[1]. In the VSN, every vehicle moves along the road and senses the urban area. The vehicles are assumed to generate sensing data packets when they satisfy at least one of the following conditions: 1) a vehicle moves 100m without any data generation. 2) a vehicle moves without any data generation during 30 seconds. 3) a vehicle reaches at one of 84 intersections. Thus, vehicles generate packets based on their moving distance, time and geographic position. All the vehicles and APs periodically send beacon packets to detect each other every second. If they detect each other, they try to send their packets based on their routing algorithms. Table I presents parameters and scenarios.

TABLE I
SIMULATION SETTINGS

Parameter Name	Definition
Number of vehicles (buses)	130(35), 110(30), 90(25), 70(20)
Simulation time	1 hour
Wireless device	802.11a
Data packet size	512 Bytes
Communication range	150m

B. Tested Algorithms

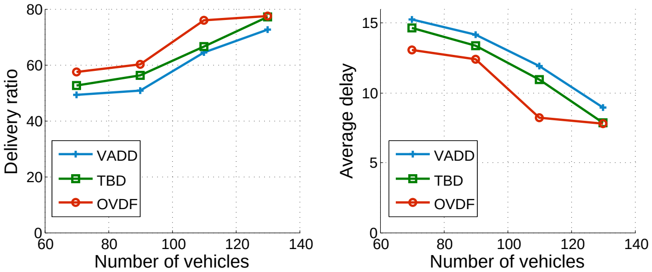
We evaluate the performance of our routing algorithm, Optimal VSN Data Forwarding (OVDF) by comparing with two existing routing algorithms designed for vehicular networks in the literature: VADD [18] and TBD [6]. Both VADD and TBD make the packet routing decisions by estimating the expected data delivery delay to the destination based on the traffic statistics for a given road network. In TBD, each vehicle also utilizes its future trajectory for computing its routing metric. They are basically designed for unicast routing, but can be easily extended to the case of anycast, following the method introduced in [6]. Namely, at an intersection, the expected delay to an AP through each neighbor intersection is estimated assuming unicast. For anycast routing, at each intersection, the expected delay on a routing path is assumed to be the minimum of the estimated delays to all APs. While this method tries to minimize the minimum of expected delays, it does not fully exploit the effect of multiple destinations since the actual optimal solution is obtained by minimizing the expectation of minimum delays to all APs. As will be shown in the following, this extension can lead to significant performance degradation.

C. Simulation Results

1) Data Delivery Ratio in Sensing Coverage: We first evaluate the sensing coverage of the three algorithms mentioned above. One of the most important performance metrics in VSN routings is the delivery ratio within a certain deadline. In our simulation, the deadline of a packet is fixed to 10 minutes since its creation. To show the spatial coverage performance, we divide 4.5km $\times$ 4km of Shanghai downtown into a grid of 72 0.5km $\times$ 0.5km squares and measure the delivery ratios of the algorithms for each square. Those are 110 vehicles (including 30 buses) and 3 APs.

Fig. 6 plots the delivery ratio within 10 minutes where the x-y plane represents the grid and z-axis represents the delivery of sensing data packets generated in the corresponding squares.

[1]Although the last update year of GloMoSim was 2001, the IEEE 802.11a MAC protocol stack of GloMoSim is still consistent to the current standards and many of physical layer models are included [12].



(a) Average delivery ratio (b) Average delay

Fig. 8. Delivery ratio with 10 minutes and average delay versus the number of vehicles.

Out of all the total packets generated, more than 95% of them are from 22 squares (this is due to the road structure of Shanghai downtown imbalance among squares). Note that the areas that generated too few packets do not give meaningful results. Thus, we only consider those squares in the results and call them the "valid squares."

Compared to VADD and TBD, OVDF shows higher packet delivery ratios in most of the regions. Especially, as shown in Fig. 6(a) and Fig. 6(c), 11 squares (out of 22 valid squares) achieve at least 20% (and up to 168%) higher delivery ratio under OVDF than under VADD. The delivery ratio gain of OVDF over VADD is 30% on average for all the valid squares. Similary, from Fig. 6(b) and Fig. 6(c), 11 squares achieve at least 15% (and up to 122%) higher delivery ratio under OVDF than under TBD, and the delivery ratio gain of OVDF over TBD is 23% on average for all the valid squares.

Clearly, OVDF shows the best sensing coverage in terms of the data delivery ratio among all of the tested routing algorithms. Especially, in the edge areas where created data are hard to be delivered with a small delay, OVDF greatly improves the data delivery ratio compared to VADD and TBD.

2) Average Data Delay: We also compare the delay performance of the three routing algorithms in Fig. 7 where the number on a bar is the average delivery delay of a packet generated in the corresponding square. As shown in the figure, OVDF achieves lower delay than VADD and TBD in most of the squares. On average, OVDF reduces the delay by about 25% and 20% compared to VADD and TBD respectively. In brief, our routing algorithm achieves faster delivery of sensing data than the existing algorithms.

3) Performance against the Number of Vehicles: Next, we examine the routing algorithms by changing the number of vehicles. Fig. 8 shows the average delivery ratio within 10 minutes and the average delivery delay of each algorithm. When the vehicle density is relatively low, OVDF performs much better than the other algorithms. However, as the vehicle density increases, the performance gap reduces. This is because in densely connected vehicular networks, packets are delivered to the destination mostly by V2V forwarding, and thus the routing decision at intersections has marginal impact on the performance. Nonetheless, Fig. 8 shows that our algorithm outperforms VADD and TBD regardless of the vehicle density.

VI. Concluding Remarks

Many of emerging VSN applications require timely delivery of sensing data and wide sensing coverage. However, this is a challenging problem in the VSN where the data links are intermittently connected. To address the issues, we develop a delay-optimal VSN routing algorithm, capturing three key features in urban VSNs: (i) vehicle traffic statistics, (ii) anycast routing and (iii) known future trajectories of vehicles such as bus. Using real traces of 95 taxis and 30 buses in Shanghai we conduct extensive simulations on GloMoSim simulator, and show that our optimal algorithm outperforms other existing algorithms. Our results demonstrate that carefully designed packet routing algorithms can greatly improve the delay performance in the VSN, and thus are the key to the success of VSN applications that require stringent delay performance guarantee. In this paper, we focused single-copy routing algorithms. Although multi-copy routings can incur serious congestion, there are scenarios where multi-copy routings outperform single-copy routing. Therefore, it would be interesting to extend our framework to account for multi-copy routing as a future work.

References

[1] Shanghai grid. online http://www.cse.ust.hk/dcrg.
[2] A. Balasubramanian, B. N. Levine, and A. Venkataramani. DTN routing as a resource allocation problem. In *Proceedings of SIGCOMM*, 2007.
[3] R. Bellman. A markovian decision process. *Indiana Univ. Math. J.*, 6:679–684, 1957.
[4] D. P. Bertsekas. Dynamic programming and optimal control. volume 2. *Athena Scientific 3rd edition*, 2007.
[5] J. Burgess, B. Gallagher, D. Jensen, and B. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networking. In *Proceedings of IEEE INFOCOM*, 2006.
[6] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du. Trajectory-based data forwarding for light-traffic vehicular ad hoc networks. *Parallel and Distributed Systems, IEEE Transactions on*, 22(5):743 –757, may 2011.
[7] B. Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *Proceedings of ACM MobiCom*, 2000.
[8] K. Lee, Y. Yi, J. Jeong, H. Won, I. Rhee, and S. Chong. Max-contribution: On optimal resource allocation in delay tolerant networks. In *Proceedings of IEEE INFOCOM*, pages 1 –9, march 2010.
[9] U. Lee, E. Magistretti, M. Gerla, P. Bellavista, and A. Corradi. Dissemination and harvesting of urban data using vehicular sensing platforms. *Vehicular Technology, IEEE Transactions on*, 58(2):882 –901, feb. 2009.
[10] X. Li, W. Shu, M. Li, H. Huang, and M.-Y. Wu. Dtn routing in vehicular sensor networks. In *Proceedings of IEEE GLOBECOM*, 2008.
[11] C. Liu and J. Wu. Practical routing in a cyclic mobispace. *IEEE/ACM Transactions on Networking*, 19(2):369–382, Apr. 2011.
[12] E. Magistretti, O. Gurewitz, and E. W. Knightly. 802.11ec: Collision avoidance without control messages. In *Proceedings of ACM MOBICOM*, 2012.
[13] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, WDTN '05, 2005.
[14] S. J. U. Traffic Information Grid Team, Grid Computing Center. Shanghai taxi trace data. http://wirelesslab.sjtu.edu.cn/.
[15] Y. Wu, Y. Zhu, and B. Li. Trajectory improves data delivery in vehicular networks. In *Proceeding of IEEE INFOCOM*, 2011.
[16] F. Xu, S. Guo, J. Jeong, Y. Gu, Q. Cao, M. Liu, and T. He. Utilizing shared vehicle trajectories for data forwarding in vehicular networks. In *Proceedings of IEEE INFOCOM*, 2011.
[17] X. Zeng, R. Bagrodia, and M. Gerla. Glomosim: a library for parallel simulation of large-scale wireless networks. In *Parallel and Distributed Simulation. Proceedings. Twelfth Workshop on*, 1998.
[18] J. Zhao and G. Cao. Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks. *Vehicular Technology, IEEE Transactions on*, 57(3):1910 –1922, may 2008.