VDNet: An Infrastructure-less UAV-Assisted Sparse VANET System with Vehicle Location Prediction

***Abstract*—Vehicular Ad Hoc Network (VANET) has been a hot topic in the past few years. Compared with vehicular networks where vehicles are densely distributed, sparse VANET have more realistic significance. The first challenge of a sparse VANET system is that the network suffers from frequent disconnections. The second challenge is to adapt the transmission route to the dynamic mobility pattern of the vehicles. Also, some infrastructural requirements are hard to meet when deploying a VANET widely. Facing these challenges we devise an infrastructure-less unmanned aerial vehicle (UAV) assisted VANET system called VDNet, which utilizes UAVs, particularly quadrotor drones, to boost vehicle-to-vehicle (V2V) data message transmission under instructions conducted by our distributed vehicle location prediction algorithm. VDNet takes the geographic information into consideration. Vehicles in VDNet observe the location information of other vehicles to construct a transmission route and predict the location of a destination vehicle. Some vehicles in VDNet equips an on-board UAV, which can deliver data message directly to destination, relay messages in a multi-hop route and collect location information while flying above the traffic. The performance evaluation shows that VDNet achieves high efficiency and low end-to-end delay with controlled communication overhead.**

I. INTRODUCTION

Propelled by high demand of road safety and navigation accuracy, vehicle communications are becoming increasingly popular nowadays. After years of development of wireless communication and mobile ad hoc network, the concept of VANET (Vehicular Ad Hoc Network) has come forward and built foundation for unlimited forms of vehicle-to-vehicle applications. New standards for vehicular communication such as DSRC (Dedicated Short-Range Communications [20]) and more recent IEEE 802.11p (Wireless Access in Vehicular Environment, WAVE [3]) are emerging, which enhances the effectiveness and feasibility of vehicular communications. With the innovation and rapid development in personal digital gadgets, especially smart phones and wearable devices, people have naturally increased their demand in the interconnectivity of things around them. Vehicles are now an indispensable part in our life. By embedding new technology, manufacturers are broadening our view of vehicles from a source of transportation to an integrated center of information and recreation. People have been exploring new possibilities in vehicular applications such as the vehicle-to-vehicle communication protocol for cooperative collision warning proposed in [21] and the smart parking scheme for large parking lots based on VANET proposed in [10]. However, most of these VANET applications need extra infrastructure, which makes them hard to deploy.

Unmanned Aerial Vehicle (UAV) is another popular topic in recent years. UAVs are semi-autonomous or fully autonomous unmanned aircrafts that have some embedded sensors, cameras, communication equipment and so on. Originally UAVs were mainly deployed in military applications. Recently increasing interest has been found in diverse civilian and commercial applications such as aerial photography, traffic monitoring, express delivery and so on [15]. Also, many UAVs have storage space and some on-board intelligence, and they are small in size, which make them a possible way to improve connectivity and efficiency of VANET. Quadrotor drones (or quadcopters) are the most popular among other forms of civil UAVs for its ability to hover, flexibility, scalability and cheap expenses. Vehicle manufacturer Renault has revealed a car prototype Kwid which possesses a traffic spotting quadrotor drone [14]. On-vehicle UAVs will stimulate innovations on new solutions and applications.

Though VANET has established its attractiveness by its all kinds of possible applications, it has some typical problems yet to be solved, e.g. connectivity. In VANET, the mobility of vehicles results in a highly dynamic topology, the connectivity of the whole vehicular network is hardly steady in time scale and space scale since the connection between two vehicles can disappear quickly. Especially in sparse network scenario, e.g. limited amount of vehicles locate in a large space, the vehicular network tend to be disconnected. In consideration of this disconnectivity in sparse VANET, the concept of Delay Tolerant Network (DTN) was proposed. In DTN, packet delivery is augmented by allowing nodes to store the data packets when there is no connection with other nodes, to carry the packets for some distance until meeting with other nodes, and to forward based on some metric on nodes' neighbors (called carry-and-forward strategy [7][9][11]). Some vehicular routing protocols were designed to meet the requirement of sparse vehicular network. VADD [22] is a vehicular routing strategy aimed at improving routing by the idea of carry-and-forward based on the use of predictable vehicle mobility, but restricts the destination to be a fixed site. GeOpps [8] takes advantage of the suggested route of the navigation system embedded in the vehicle to select vehicles that are likely to move closer to the final destination of a packet as a next-hop node, but need assistance from an exterior navigation system. And other proposed systems mostly need infrastructural support such as Roadside Units (RSUs), which makes those systems hard to deploy widely.

In this paper, we are committed to developing a novel infrastructure-less sparse VANET system (VDNet) with assisting on-board UAVs (quadroter drones, "drones" for short) and a vehicle location prediction algorithm to provide highly efficient vehicle-to-vehicle data message delivery with relatively low end-to-end delay, meanwhile reduce the communication overhead within the network.

Firstly, we provide a distributed location-based routing protocol for this Vehicle-Drone hybrid vehicular ad hoc Network (VDNet). This protocol takes geographic information into consideration and guarantees highly efficient data transmission under both inter-vehicle immediate transmission situation and carry-and-forward situation, along with an efficiency boost provided by on-vehicle drones, which only some vehicles are equipped with.

Secondly, we design a distributed database scheme for location information. Traffic trajectories can be inferred from the information data maintained by each vehicle within the network. Also, it helps to improve data message delivery efficiency.

Thirdly, in order to improve the accuracy of vehicles' next-hop selection and the drone dispatching, we propose a distributed vehicle location prediction algorithm based on peer-observed history location information.

Finally, we developed a simulation platform designed for vehicle-drone hybrid vehicular ad hoc network, and evaluated the performance of our protocol and algorithm.

The remainder of this paper is organized as follows. In Section II we discuss related work. In Section III, we introduce the system model and the data transmission scheme in VDNet. In Section IV, we propose a vehicle location prediction algorithm to improve efficiency in VDNet. In Section V, we evaluate VDNet's performance. In Section VI, we conclude this paper and discuss our future work.

II. RELATED WORK

GVGrid [17] partitions region into small square grids and aims to find a route from source to areas surrounding the destination point. GVGrid assumes each vehicle knows its position and direction information from Global Positioning System (GPS) with a digital map. It also proposes a neighbor selection algorithm to decide which neighbor to forward messages. Therefore, any vehicle in a grid could serve as an intermediate and provides robustness. Instead of in dense ad-hoc network with high multi-hop connectivity, our system considers a sparse VANET scenario, where the connectivity could be much lower with frequent disconnections.

In [11], a data message transmission scheme is proposed to increase higher transmission ratio and low end-to-end delay with high mobility and sparse vehicle distribution. The author combines both dynamic inter-vehicular wireless communication and a store-carry-forward method. To reduce the communication overhead, a data transmission route is consisted of a sequence of street fragments, where, by evaluation, the number of fragments of most roads are small. Instead of requiring the instantaneous accurate position of target vehicles, we have considered a distributed message transmission algorithm based on route prediction. Also, we use drones to increase network connectivity and decrease end-to-end delay.

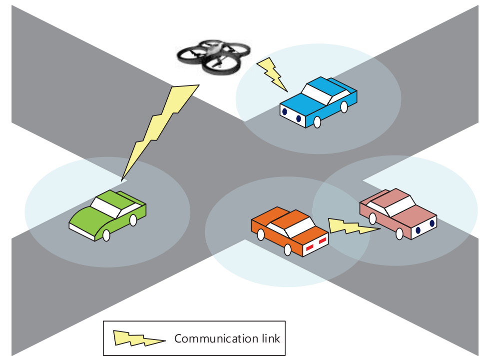


Fig. 1. VDNet architecture

[19] proposes a store-carry-forward approach in mobile ad-hoc networks, using node mobility to increase connectivity. In this model, nodes are controlled to move or stay stationary with bound on moving distance for each communication. They propose a scheduling algorithm of nodes and redistribute nodes dynamically according to traffic rate and system demands. The movement of vehicles in our system is not controlled by a central controller. We exploit this randomness and mobility in network to deliver messages and increase connectivity.

In [16], a method of placing unmanned aerial vehicles (UAVs) platform is proposed. They use performance metrics to ensure that resources are fairly allocated to different flow classes with different QoS requirements. They also propose an algorithm to compute the optimal admission rate split and thresholds for each flow class among the routes. Also, an iterative computation scheme is proposed to predict the best location of the UAV when the platform's location is changed. In our work, we use drones to transmit or relay messages to increase the efficiency of data transmission in this network, while the placement of drones is based on location prediction.

III. SYSTEM MODEL AND DATA TRANSMISSION SCHEME

A. System Model

An overview of the VDNet architecture is shown in Figure 1. The system consists of sparsely distributed vehicles and a relative smaller amount of drones.

- Vehicles in our system are all capable for short range communication. Only some vehicles possess one on-board drone. Each vehicle maintains a *location information database* (Table I). Also vehicles can temporarily store the data message that need to be forwarded until it finds a next-hop. Vehicles have basic GPS functions that can detect their own location. Each vehicle continuously updates its own location information in the database.
- Drones are small UAVs which are equipped by some vehicles in our system. One important feature of drones in VDNet is the ability to hover, which makes quadcopters a particularly appropriate choice. Drones are also capable for communication and autonomous flying. However, a drone can only be intermediate nodes of a multi-hop route or be dispatched to directly deliver the message.

  Since the actual communication range between a drone and a vehicle is determined by the smaller one of the drone's and the vehicle's, when forwarding a data message between a drone and a vehicle, we treat their range as the same. A drone only belongs to one certain vehicle. A drone can fly and hover within a predetermined control range centering at its parent vehicle. Also, we allow a vehicle to dispatch its drone to a certain street intersection, then drives to that intersection and collects the drone, as shown in Figure 2.

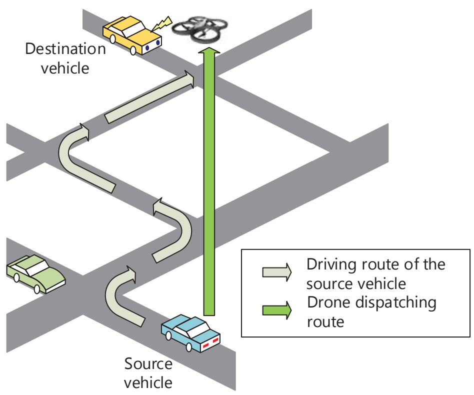


Fig. 2. A source vehicle dispatches its on-board drone to forward a data message to the destination vehicle

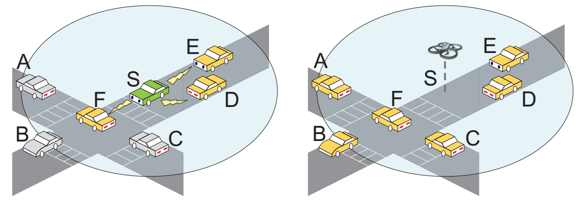
- Location information database is a database mounted on every vehicle and drone, storing location information entries. As shown in Table I, each entry has the following attributes.
  - Vehicle ID (VID): A preassigned ID V_i for the vehicle.
  - Location (LOC): The observed location tuple $\mathcal{L}(V_i)$.
  - Direction (DIR): The heading direction h_i of vehicle V_i. If V_i is driving away from the origin of the road, $h_i = 1$. Else if V_i is driving towards the origin of the road, $h_i = -1$.
  - Velocity (VEL): The observed velocity of vehicle V_i
  - Refresh time (RFR): This field records how much time have passed since last time this entry is refreshed. Notice that both observations and the compare-and-exchange process (Section III-C) may refresh database entries.
  - Receive time (RCV): This field records how much time have passed after receiving this entry.

TABLE I
THE STRUCTURE OF LOCATION INFORMATION DATABASE

VID	LOC	DIR	VEL	RFR	RCV

We need to make some necessary assumptions in our model.

- **Assumption 1**: Drones' actual transmission range is large enough for air-to-ground communications regardless of the altitude.
- **Assumption 2**: A vehicle can only communicate with other vehicles which are on the same street. Even within transmission range, the communication between vehicles which are on different street may be blocked by obstacles like building. However since drones can fly high enough, generally they will not be affected (Figure 3(a)).
- **Assumption 3**: Drones always have sufficient battery after taking off from its parent vehicle.



(a) (b)

Fig. 3. (a) S is a source vehicle, by Assumption 2 it can communicate with vehicles D, E and F, not with A, B and C. The dashed circle represents the transmission range of S. (b) S is a source drone, it can communicate with all vehicles within its transmission range. The dashed line shows the projection of the drone on the street.

B. Transmission Route Specification

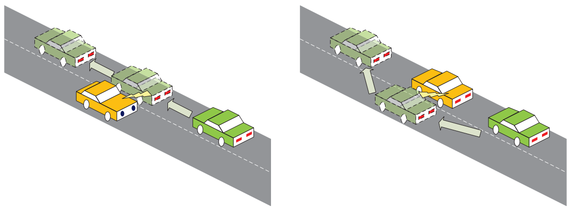
In dense vehicular networks, it may be possible for each vehicle to forward the data message to its next-hop immediately after receiving the message. So the choice of a next-hop vehicle may be independent of streets. However, it may be impossible for vehicles to receive and forward the data message at the same time if the vehicles are sparsely distributed. In this case, data message is stored and carried forward by the vehicles or drones until it is able to select a next-hop vehicle and drop the message to it. Thus, when determining a transmission route, we need to take geographic information of the vehicles into consideration.

Similar with [11], in our discussion, street is a line segment or curve with two end points and no branch on the map. We assign each street an ID R_i and set one of its two end points as the origin. The location $\mathcal{L}(V_i)$ of the vehicle V_i is a tuple of street ID on which the vehicle is located and distance d_i from the origin of the street to the vehicle V_i along the street, i.e.

$$\mathcal{L}(V_i) = (R_i, d_i) \tag{1}$$

The location of a drone is similarly defined. Note that a location $\mathcal{L}(V_i) = (R_i, d_i)$ can be transformed into an x-y coordinate (x_i, y_i), vice versa. A crossing can be represented with different street IDs since it belongs to multiple streets, i.e. $\mathcal{L}(X) = (R_i, d_i), (R_j, d_j), \ldots$.

When a vehicle V_s initiates a data transmission, it will look up the destination vehicle's ID V_d in the *location map* which is generated by the on-board location information database with the latest location information of other vehicles and outdated information removed. If V_d is not on the map, V_s will turn into passive mode and wait for V_d to appear in received location broadcasts. If V_d is already on the map, V_s will calculate a transmission route from its location to the location of V_d according to some path finding algorithms as described in [2], [1]. A street segment is represented by a tuple of its two end points, i.e. $((R, d^s), (R, d^d))$ where (R, d^s) is a location of the source vehicle or a crossing on the route and (R, d^d) is a location of the destination vehicle or a crossing on the route. A data



(a) (b)

Fig. 4. Two cases when a vehicle in passive mode communicates with another vehicle

transmission route is a sequence of street segments and is represented as $(((R_1, d_1^s), (R_1, d_1^d)), \ldots, ((R_n, d_n^s), (R_n, d_n^d)))$ where (R_1, d_1^s) is the location of a source vehicle V_s, (R_n, d_n^d) is the location of a destination vehicle V_d. Notice that $(R_i, d_i^d) = (R_{i+1}, d_{i+1}^s)$ for $i = 1 \ldots n-1$.

C. Routing Information Acquisition and Communication Overhead Reduction

In VDNet, a transmission route is composed of multiple vehicles and drones. Vehicles have greater impact on the whole network routing process because drones are not always flying away from their parent vehicle. Since there is no central server in our network, the routing algorithm and route table are fully distributed. Actually the transmission route can be uniquely deduced from the on-board location information database (Table I). Intuitively, in a distributed routing protocol like AODV [13], the propagation of route information produces large broadcast overhead. This especially reduces the efficiency of a sparse VANET because it will occupy the limited channel resource and may cause data message transmission fail in the short window of connection establishment. We propose two methods to solve this problem.

First, our system categorize the vehicles into two modes: *active* and *passive*. Passive vehicle compare-and-exchanges its location information database each time it changes its relative position along the street with another vehicle, as shown in Figure 4. We will show the detailed process of compare-and-exchange in the following paragraph. When a passive vehicle wants to forward a data message, it will switch its mode into active. Active vehicle broadcasts its location message to neighbor vehicles every unit time (one second in our simulation setting). If an active vehicle does not initiate new data transmissions for some time, it will switch back its mode to passive. We identify each vehicle by its ID V_i. The indicator function $\mathcal{I}(V_i)$ returns the vehicle's mode, i.e. We set all flying drones to be active since the flight duration should not be long.

$$\mathcal{I}(V_i) = \begin{cases} 1 & V_i \text{ is active} \\ 0 & V_i \text{ is passive} \end{cases} \tag{2}$$

Second, rather than simply copying its own database to others, every vehicle and drone in VDNet will gain routing knowledge and contribute to peer's routing knowledge by a process called *compare-and-exchange* as shown in Algorithm 1. The algorithm has $\mathcal{O}(n)$ running time, it reduces much overhead and almost does not require additional preparing time for transmission.

Algorithm 1: Compare-and-exchange

Exchange a subtable containing only VID and RFR columns with the vehicle or the drone;
foreach *entry* $\mathbf{E}_i$ *of the obtained subtable* **do**
  if $\mathbf{E}_i \to$ *VID is already in local database* **then**
    Find the entry $\mathbf{E}_j^L$ in local database with $\mathbf{E}_j^L \to \text{VID} = \mathbf{E}_i \to \text{VID}$ and $\mathbf{E}_j^L \to \text{RFR}$ is the smallest;
    if $\mathbf{E}_j^L \to RCV \leqslant \mathbf{E}_i \to RFR$ **then**
      continue;
  else
    Retrieve the complete entry $\mathbf{E}_i^C$ and store it in local database;

Here we give a simple analysis of the convergence of this routing information acquisition model. Actually we only need to prove the convergence under all-passive mode setting since intuitively this setting propagates location information in the slowest way. We first divide each street into n segments, so the total number of segments in an N by N network is $2nN(N+1)$. Then we assume the movement of vehicles from segment to segment is an Markovian process [6], where each segment stands for a state. Since the network is limited and the number of states is finite, this process will converge and tend to be stable after some finite time t_0, where there exists a segment s that the probability of a vehicle being in this segment is p, $p \geqslant p_0 = \frac{1}{2nN(N+1)}$. We denote $P_{i,t}$ as the probability of vehicle i being in segment s at time t. For a time $t_k > t_0$, the probability that every vehicle knows the position of other vehicles is $P_{all\,known}(t_k)$. It is lower bounded by the probability that all vehicles have met each other in segment s. $P_{all\,known}(t_k) > \prod_{i=1}^{M} P_{i,t_k} \geqslant p_0^M$. So, $P_{not\,all\,known}(t_k) < 1 - p_0^M$. For some time t_K, $P_{not\,all\,known} = \prod_{i=1}^{K} P_{not\,all\,known}(t_i) \leqslant (1-p_0^M)^K$. When $K \to \infty$, $P_{not\,all\,known} \to 0$.

D. Data Message Transmission

In this section, we propose a data message transmission strategy based on the transmission route discussed in the previous section. Since the vehicular network is sparsely distributed in a wide area, there are basically two ways for a message to be delivered:

1) Instantaneously forwarded through inter-vehicle communication;
2) Stored in vehicles or drones, carried along the route then forwarded with certain delay.

Each data message is unicasted by intermediate vehicles or drones, so that no copy of a data message is required, the communication overhead is low. The source vehicle can forward its data message to its own drone and dispatch it, or it can forward the data message to an intermediate vehicle or drone. Intermediate drones and vehicles forward its received data message to a selected next-hop vehicle.

In Section III-C we showed that by the mutual-inform mechanism of the location message, all locations of the vehicles in our system will be acquired in finite time. Though there may exist slight difference of location databases between different vehicles due to the sparsity of the network, intuitively since vehicles keep encountering with each other, the difference may be small and the effect it took on next-hop selection is insignificant. When vehicle V_i needs to forward a data message, it will invoke Algorithm 2 to select a next-hop vehicle V_{i+1}. If the algorithm returns V_i itself, V_i will keep running the algorithm while driving until a next-hop is found. Notice that the calculated transmission route is independent with vehicle's driving route, and we assume vehicles should not detour to deliver a data message.

Algorithm 2: Next-hop selection

```
Input: A list V containing all vehicles and drones which are in
       transmission range of the vehicle V_i, and the vehicle's own
       drone D_i if available; The destination vehicle V_t;
Output: The selected next-hop vehicle or drone V_{i+1};
1  if V_t ∈ V then
2      return V_t;
3  else if V_i possesses a drone D_i then
4      if the drone D_i is already dispatched and not yet collected then
5          goto 12;
6      else
7          if L(V_t) is on the driving route of V_i then
8              return D_i;
9          if L(V_t) is not on the driving route of V_i then
10             goto 12;
11 else if V_i does not possess a drone then
12     append V_i to V;
13     find V_k in V whose distance from V_t is the shortest by the record
       in location information database;
14     return V_k;
```

IV. Vehicle Location Prediction

In order to find an effective transmission route, every vehicle in our network will obtain location information of other vehicles by both observation and comparing database with other vehicles. However, the location information every vehicle can infer from the database it maintains is not real-time due to the location message acquisition mechanism discussed in previous sections. So we propose a vehicle location prediction algorithm based on history and recent location data.

A. Raw Prediction

Recall the structure of the location information database every vehicle maintains, which is shown in Table I. Described in Section III-B, vehicles are categorized into active mode and passive mode. Every vehicle compares and exchanges its location information database with another vehicle when they encounter, also both will revise corresponding entries of the database.

When vehicle V_s initiates a data message transmission to vehicle V_d, first it will look up the location information map which shows the latest observed locations of the vehicles according to the database. If V_d is not found, V_s keeps this transmission session waiting until V_d appears on the location information map as discussed previously. If V_d is already on the map with refresh time ts , V_s will calculate the current along-the-street location of V_d by

$$\hat{\mathcal{L}}(V_d) = (\hat{R_d}, \hat{d_d}) = (R_d, d_d + h_d \cdot v_d \cdot t) \tag{3}$$

Notice that the distance from the origin is measured along the street, not always a straight-line distance. However, if $\hat{d_d} < 0$ or $\hat{d_d} > l_d$ where l_d is the length of street R_d, the raw prediction is obviously unreliable. Actually it is unlikely that we can use raw prediction in a sparse VANET. To solve this problem, we propose an advanced location prediction algorithm.

B. Advanced Location Prediction

We can obtain more informative probabilistic information from prior history of vehicle's locations. Some prediction methods based on efficient routes and partial trajectories were proposed in [4] and [5]. We propose a novel probabilistic vehicle location prediction approach based on accumulated location information and recently observed vehicle location.

An intuitive way to predict the location of a vehicle is to give higher probability to the places where the vehicle has been before. This is the main principle behind much previous work in pervasive computing on modeling and predicting transportation routines [5]. We first assume that a vehicle will only visit locations that it has been observed to visit in the past. We refer to this assumption as *closed* assumption. This assumption is the foundation of much previous work on predicting a person's location. However, vehicles actually can visit locations where they have not been observed to visit yet, and intuitively this is a common situation especially in early phases of observing a vehicle. In order to improve accuracy, we need to take the likelihood of vehicle drives to a previously unobserved location into consideration. An *open* model is needed to use the closed probability mass function calculated at a relatively early phase in the network to approximate the steady state probability that would be observed at the time vehicles actually need a location data of another vehicle, or even several minutes after.

We expect to compute the probability of each street section being the vehicle's location under the condition of the location history of the vehicle and its recent appearances, i.e. $P(L = \mathcal{L}|\mathbf{S} = \mathbf{s})$, where L is a random variable representing the location and $\mathbf{S}$ is a random variable representing the vector of recent observations of the locations, heading directions and timestamps from the vehicle's trip.

Definition 1. *We denote $\mathcal{L}_i$ as a shorthand for the location tuple of the i-th indexed candidate location. Given a finite set K of location tuples $\mathcal{L}$, if $\|\mathcal{L}_i - \mathcal{L}_j\| < \epsilon, \forall \mathcal{L}_i, \mathcal{L}_j \in K$, then we call the region which is composed of these locations in K a preferred region. We divide preferred regions into three categories:*

1) *Area*
2) *Route*
3) *Isolated location*

The categorization of location record should follow the priority order given above. After this categorization, we obtain a set of preferred regions $\mathbf{K}$, each element $K_n^{(c)}$ in $\mathbf{K}$ represents a preferred region. We assume there are M areas, N routes and P isolated locations after categorization. Then we can calculate the probability of each street segment being the vehicle's location. The superscript c shows the category. It may be M, N or P. The subscript n in $K_n^{(c)}$ represents that it is the n-th preferred region in category c. $|\cdot|$ returns the amount of elements that construct this set.

1) Area: We regard each area as a potential location of the vehicle, and the probability is given by:

$$P_M(L = \mathcal{L}_i) = \frac{|K_i^{(M)}|}{\sum_{m=1}^{|M|} |K_m^{(M)}|} \tag{4}$$

2) Route: Our intuition is that the two end points of a route is likely to be potential locations. For each route $K_n^{(N)}$, we can conclude the probabilities of two end points being the location are equal, because such routes may by a predetermined route that connects two commonly visited place for the vehicle, for example, home and work place. $P_N(L = \mathcal{L}_j)$ represents the probability of $\mathcal{L}_j$, which is one end point of $K_j^{(N)}$, being the vehicle's location. Thus, we conclude the following equation:

$$P_N(L = \mathcal{L}_i) = \frac{0.5|K_i^{(N)}|}{\sum_{n=1}^{|N|} |K_n^{(N)}|} \tag{5}$$

3) Isolated location: The isolated locations cannot be seen as the potential location, because most of them are arbitrary. However, some of them can be used to indicate the driving habit of the vehicle. Thus, we now define:

$$d_{K_i^{(P)}} = min\|K_i^{(P)} - K_j^{(M\cup N)}\|, K_i \in P, K_j \in M\cup N \tag{6}$$

Where $\|K_i^{(P)} - K_j^{(M\cup N)}\|$ denotes the minimum distance between $K_i^{(P)}$ and all locations constructing $K_j^{(N\cup M)}$. For all isolated locations, we use the following indicator function to divide them into two sets:

$$K_i^{(P)} \in \begin{cases} \mathcal{P}_{\mathcal{D}}, & \text{if } d_{K_i^{(P)}} \leqslant d_{TH} \\ \mathcal{P}_{\mathcal{E}}, & \text{if } d_{K_i^{(P)}} > d_{TH} \end{cases} \tag{7}$$

Where each element in $\mathcal{P}_{\mathcal{D}}$ is discarded from future analysis, because these locations can be viewed as the occasional deviation from the location of $M \cup N$, while $\mathcal{P}_{\mathcal{E}}$ can be used to estimate the probability that the user tend to go for the place that he has never been before. d_{TH} denotes the threshold distance. Thus, we may now define a coefficient α to indicate this probability:

$$\alpha = \frac{\sum_{p=1}^{|\mathcal{P}_{\mathcal{E}}|} |K_p^{(\mathcal{P}_{\mathcal{E}})}|}{\sum_{m=1}^{|M|} |K_m^{(M)}| + \sum_{n=1}^{|N|} |K_n^{(N)}| + \sum_{p=1}^{|\mathcal{P}_{\mathcal{E}}|} |K_p^{(\mathcal{P}_{\mathcal{E}})}|} \tag{8}$$

We now calculate the probability under close assumption:

$$\underset{\mathcal{L}_i \in K_i^{(c)}}{P_{close}(L = \mathcal{L}_i)} = \frac{|K_i^{(c)}|}{\sum_{m=1}^{|M|} |K_m^{(M)}| + 2\sum_{n=1}^{|N|} |K_n^{(N)}|} \tag{9}$$

Notice that $\mathcal{L}_i \in K$ in (9). If $\mathcal{L}$ is not recorded in the database, $P_{close}(L = \mathcal{L}) = 0$.

For a location $\mathcal{L}_t$, we can uniquely convert it into a rectangular coordinate expression (x, y), e.g. $\mathcal{L}_t = (x_t, y_t)$. In order to deduce the open-world probability, we now define a *central location* of some given K detected area location:

Definition 2. *Given K isolated locations $\mathcal{L}_i$, we obtain $\mathcal{L}_z = (x_z, y_z)$ by*

$$x_z = \frac{1}{K}\sum_{i=1}^{K} x_i \qquad y_z = \frac{1}{K}\sum_{i=1}^{K} y_i$$

Then we call $\mathcal{L}_z$ the central location of these K area locations.

Thus, we can get M central location $\mathcal{L}_z^m$ for $m \in M$ by *definition 2*. Moreover, we can also get $2N$ central locations $\mathcal{L}_z^n$ for $n \in N$ because each route has two terminals.

The probability of location in open model is given by the following relationship:

$$P_{open}(L = \mathcal{L}) = (1 - \alpha)P_{close} + \alpha P_{M\cup N}^{Open} \tag{10}$$

where $\mathcal{L}$ is an arbitrary location and

$$P_{M\cup N}^{Open}(L = \mathcal{L}) = \sum_{m=1}^{|M|} \frac{|K_m^{(M)}|}{\|\mathcal{L}_i - \mathcal{L}_m^z\|^\beta} + \sum_{n=1}^{|N|} \frac{2|K_n^{(N)}|}{\|\mathcal{L}_i - \mathcal{L}_n^z\|^\beta} \tag{11}$$

The main idea to calculate $P_{M\cup N}^{Open}$ is based on the fact that the vehicle is not likely to drive far away from his "preferred region". Thus, given an arbitrary location $\mathcal{L}$, the probability is the weighted sum of all its preferred region divided by a value related to the distance between $\mathcal{L}$ and the certain preferred region. The coefficient β is termed as *Lazy Factor*, which evaluates to what extent the vehicle is likely to go far away from his preferred region. The larger β is, the less possible for the vehicle to go far away, and vice versa. β is an experimental value, which is determined by $\mathcal{P}_{\mathcal{E}}$. Recall that the elements in $\mathcal{P}_{\mathcal{E}}$ are isolated points, and it can not only be used to evaluate the probability a vehicle is likely to "go out for open world" (termed as α) but also used to calculate to what extent this vehicle will "go out" (termed as β).

Now we reveal how to calculate $P(L = \mathcal{L}|\mathbf{S} = \mathbf{s})$. As mentioned previously, s is a random variable representing the vector of recent observations of the features from the vehicle's trip. Thus, we may decompose the variable $\mathbf{s}$ into:

$$\mathbf{s} = \begin{pmatrix} \boldsymbol{s} \\ \boldsymbol{t} \end{pmatrix} = \begin{pmatrix} s_0, s_1, ..., s_q \\ t_0, t_1, ..., t_q \end{pmatrix}$$

Where $\boldsymbol{s}$ is a vector recording the observed $q + 1$ locations and $\boldsymbol{t}$ is another vector denoting the observation time to the corresponding location. Now we want to find an "effective direction", which means the vehicle will go the which direction

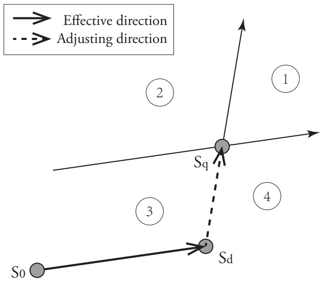


Fig. 5. An example on how to determine quadrant

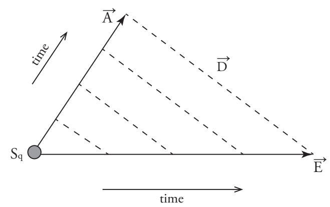


Fig. 6. An example on how to determine location in the certain time

potentially, according to s. An intuitive way is to use the Least Squares Method to find a fitting line. However, the basic idea is that all points in s are treated equally. The latest observed location may be more important to determine which direction the vehicle is tending toward.

Definition 3. *Given* s *we obtain the "tendency location" by:*

$$s_d = \frac{\sum_{i=1}^{q} s_i(t_i - t_0)}{\sum_{i=1}^{q}(t_i - t_0)}$$

We define "effective direction vector" to be $\overrightarrow{E} = s_0 \rightarrow s_d$*, and "adjusting direction vector" to be* $\overrightarrow{A} = s_d \rightarrow s_q$.

This definition is reasonable, because the latter observed locations enjoys more importance. The "effective direction vector" is the direction representing recent tendency, which includes the initial location and the "tendency location" while the "adjusting direction vector" can be understood as the fine tuning toward the destination. Now we show how to predict the location given s by Figure 5. We now translate these two vectors to the pass through s_q which is the latest observed location. Thus, we now obtain four quadrants considering s_q as the origin point. Then we assume the quadrant within these two vectors as the area that the vehicle will be in a short span of time. In Figure 5 scenario, the vehicle will most probably be in quadrant 1.

Now we have narrowed the predicted range. However, there are still a lot of locations, so now Equation (10) is incorporated into work. We define a vector $\overrightarrow{D} = \overrightarrow{A} - \overrightarrow{E}$. This vector propels over time parallel from s_q, as shown in Figure 6. Assume we should predict the location of the vehicle after a period of time Δt from the last observed time t_p, then we use the following equation to predict where $\overrightarrow{D}$ is:

$$distance(\overrightarrow{D}, s_q) = \Delta t \cdot \frac{\|s_q - s_0\|}{t_q - t_0}$$

There are several locations in $\overrightarrow{D}$. Now we use (10) to calculate which location in this line has the highest probability. Then, given s, we choose the location with the highest probability to be the predicted location after Δt, which is

$$\hat{\mathcal{L}} = \underset{\mathcal{L}_i \in \overrightarrow{D}}{\arg\max}\, P_{open}(L = \mathcal{L}_i) \tag{12}$$

We tested our vehicle location prediction algorithm on trips from collected GPS traces, the result is shown in Section V.

V. Performance Evaluation

In this section, we evaluate the performance of VDNet. We compare the communication overhead and vehicles' all-knowing time under different vehicle modes, namely, active mode, passive mode and dynamic switching between active and passive modes as mentioned in Section III-B. And we tested the communication overhead and all-knowing time after allowing drones to be relay nodes in a wireless multi-hop route. Also, we evaluate how the amount of vehicles and drones will affect the efficiency of the network. Finally, we evaluate the performance of VDNet's vehicle location prediction algorithm.

The experiment is based on a 1000m $\times$ 1000m rectangle street area based on a randomly selected region. The original map data is provided by OpenStreetMap [12]. Our simulation street layout is derived and normalized from the original map, removing unrelated topography, out-of-range streets, building and facility information, thus making the simulation map planar. Also, we replace the curves in the map with straight lines to boost running speed of the simulation.

We send 50 to 150 vehicles driving on the street layout. The initial location and heading direction is randomly chosen for each vehicle. The mobility pattern is generated similar to that of [18], that the velocity of each vehicle is a normal distribution with mean value 36km/h. Each vehicle generates new requests for sending its own data message to a random destination vehicle following Poisson distribution. Each source vehicle and intermediate vehicle will make their own judgment on how to select a next-hop or dispatch a drone by the strategies described in previous sections (Algorithm 2). A snapshot of simulation street layout is shown in Figure 7, and the detailed simulation setup is listed in Table II. Notice that as discussed in Section III-A, we set the drones' transmission range (projected to the street plane) as the same as vehicles.

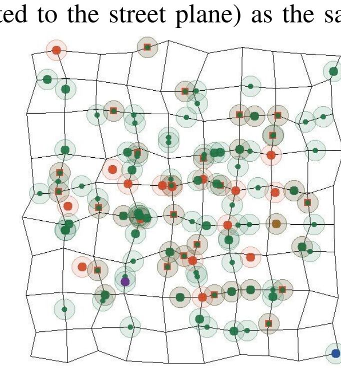


Fig. 7. Snapshot of the simulation area. Active vehicles have larger dots than passive vehicles. Green dots represent vehicles and red dots represent drones. Blue dots represent source vehicles, brown dots represent intermediate vehicle, purple dots represent destination. Dots that shaped in square represent a vehicle with on-board drones and the drone is not dispatched yet. Shaded circles around dots represent the transmission range.

A. All-knowing Time and Amount of Information

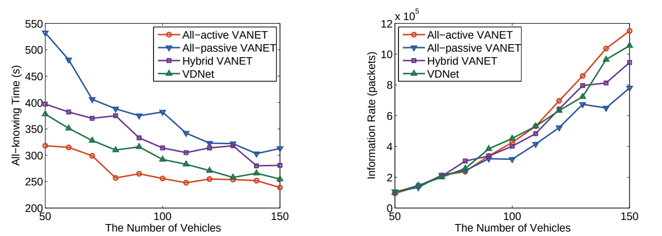
We denote the first moment that all vehicles' location information have been recorded in the location information

TABLE II
SIMULATION SETUP

Parameter	Value
Simulation area	1000m × 1000m
# of intersections	100
Vehicle velocity	Normal distribution with mean value of 36km/h
Transmission range	30m
# of vehicles	50—150
# of drones	0%—100% # of vehicles

database of every vehicle as the *all-knowing time*. Different amount of vehicles and their mode settings will affect the all-knowing time of the network. Figure 8(a) shows the all-knowing time as a function of the amount of the vehicle under three vehicle mode settings. If all vehicles are active, the all-knowing time will be small but at cost of high communication overhead (Figure 8(b)). All-passive setting causes the worst all-knowing time result. Hybrid of active and passive modes with dynamic switching (Section III-B) results in an intermediate performance on all-knowing time. And in VDNet where drones are participating in location information propagation, the performance on all-knowing time is better, but still slightly worse than all-active mode.

Figure 9 shows the probability distribution of all-knowing time in all-active, all-passive, hybrid vehicular networks, and VDNet. In order to compare the communication overhead,



(a) All-knowing time as a function of the amount of vehicles

(b) Unit time amount of information as a function of the amount of vehicles

Fig. 8. All-knowing time and unit time amount of information

we record every data message transmission during simulation. The amount of information in the network is measured by the observed number of data records. We evaluate the average amount of information during the all-knowing time, the result is shown in Figure 8(b). The amount of information increase with the number of vehicles in the network. Generally VDNet produces less amount of information than all-active scheme.

B. Data Message Delivery Efficiency

We also evaluate how drones will improve the data message delivery efficiency of the network by analyzing the distribution of the end-to-end delay and the delivery ratio during fixed time period. Figure 10 shows how the probability distribution of data message delivery time changes with drone-mounted vehicles' percentage in all vehicles. As the percentage rises, the data messages become more likely to be delivered in a shorter time. Figure 11 shows as drone owning rate increases, the average end-to-end delay of the system tend to decrease and the average delivery ratio tend to increase.

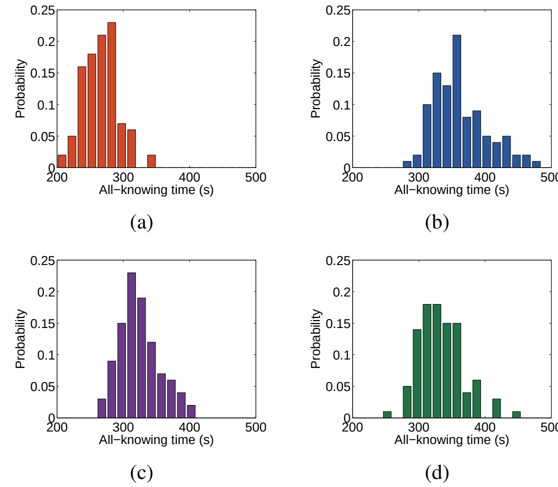


Fig. 9. PDF of all-knowing time in: (a) all-active VANET; (b) all-passive VANET; (c) hybrid VANET where all vehicles follows the rule of dynamic active/passive switching (Section III-B); (d) VDNet where drones are included into multi-hop routes and vehicles dynamically switch modes

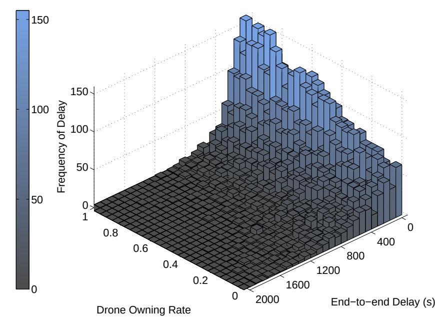


Fig. 10. The distribution of end-to-end delay of data message delivery with respect to how many percentages of the vehicles possess an on-vehicle drone

C. Vehicle Location Prediction Algorithm

We tested our vehicle location prediction algorithm on GPS traces collected in the same area. The GPS traces are provided by OpenStreetMap. We set the vehicles to drive following different traces repeatedly, and randomly add some "shift" to their trips. We tested our algorithm in three different modes:

- Closed-world model — This model uses simply a closed-world prior probability based on location information in the database before test time.
- Preferred-region model — This model uses Equation (11)

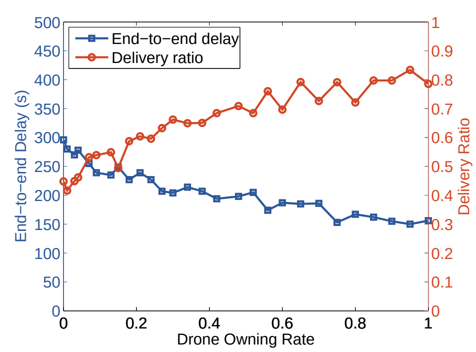


Fig. 11. Data message delivery efficiency, illustrated by average end-to-end delay and delivery ratio within fixed simulation time limit

and seek the location with the greatest probability to be the predicted location.

- Open-world model — This model uses Equation (12) to find the most probable location. This model has the best performance since recent directions and locations contribute to the prediction.

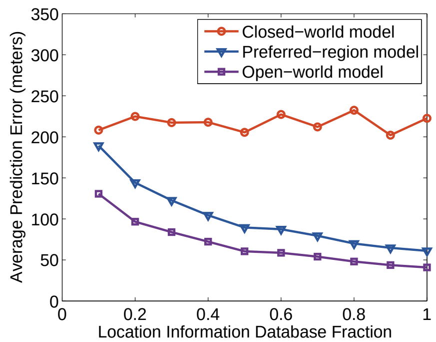


Fig. 12. The prediction error as a function of location information database usage fraction

Time sequence can be inferred from location information database fraction since the database accumulates location information over time. As shown in Figure 12, the average prediction error of preferred-region and open-world model drops with the usage fraction of location information database increases. However, the simple closed-world model is consistently poor in accuracy. The open-world model achieves better performance than preferred-region model.

VI. Conclusion

In this paper, we have devised VDNet, an infrastructure-less UAV-assisted VANET system which uses smart routing decision, drone dispatching and vehicle location prediction algorithm to provide high effectiveness of vehicle-to-vehicle data message delivery and low end-to-end delay, meanwhile control the communication overhead, compared to previous VANET systems. All algorithms and databases in VDNet are distributed. Vehicles and drones use observed history location to predict the current location of the destination vehicle, thus improves the efficiency of the system.

The accuracy of VDNet's prediction algorithm is growing by time, and it may not work if the GPS signal is blocked by crowded high buildings. Also, the effectiveness of VDNet will improve as more vehicles are equipped with an on-board drone, this will cost the drivers some money.

Future work should give the drones more freedom to dynamically multi-collect-multi-deliver messages. Also, complicated street patterns such as viaducts and tunnels need new schemes to achieve optimized utilization of vehicles and drones.

References

[1] Richard Bellman, *On a routing problem*, tech. rep., DTIC Document, 1956.

[2] Edsger W Dijkstra, "A note on two problems in connexion with graphs", in: *Numerische mathematik* 1.1 1959, pp. 269–271.

[3] Daniel Jiang and Luca Delgrossi, "IEEE 802.11 p: Towards an international standard for wireless access in vehicular environments", in: *IEEE Vehicular Technology Conference*, 2008, pp. 2036–2040.

[4] John Krumm, *Real time destination prediction based on efficient routes*, tech. rep., SAE Technical Paper, 2006.

[5] John Krumm and Eric Horvitz, "Predestination: Inferring destinations from partial trajectories", in: *UbiComp*, Springer, 2006, pp. 243–260.

[6] Dinesh Kumar, Arzad A Kherani, and Eitan Altman, "Route lifetime based optimal hop selection in VANETs on highway: an analytical viewpoint", in: *NETWORKING. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, Springer, 2006, pp. 799–814.

[7] Kevin C Lee, Uichin Lee, and Mario Gerla, "Survey of routing protocols in vehicular ad hoc networks", in: *Advances in vehicular ad-hoc networks: Developments and challenges* 2010, pp. 149–170.

[8] Ilias Leontiadis and Cecilia Mascolo, "Geopps: Geographical opportunistic routing for vehicular networks", in: *IEEE WoWMoM*, 2007, pp. 1–6.

[9] Fan Li and Yu Wang, "Routing in vehicular ad hoc networks: A survey", in: *IEEE Vehicular Technology Magazine* 2.2 2007, pp. 12–22.

[10] Rongxing Lu, Xiaodong Lin, Haojin Zhu, and Xuemin Shen, "SPARK: a new VANET-based smart parking scheme for large parking lots", in: *IEEE INFOCOM*, 2009, pp. 1413–1421.

[11] Hiroaki Oka and Hiroaki Higaki, "Wireless multihop communication in sparse Vehicular Ad-Hoc Networks", in: *International Journal of Internet Protocol Technology* 4.2 2009, pp. 115–125.

[12] OpenStreetMap, *OpenStreetMap*, [Online], URL: http://www.openstreetmap.org/about.

[13] Charles Perkins, Elizabeth Belding-Royer, Samir Das, et al., "RFC 3561-ad hoc on-demand distance vector (AODV) routing", in: *Internet RFCs* 2003, pp. 1–38.

[14] Mark Prigg, *Never get stuck in traffic again: Renault reveals concept car which comes with a DRONE that watches out for gridlock*, ed. by DailyMail.co.uk, [Online], 2014, URL: http://www.dailymail.co.uk/sciencetech/article-2553422/Release-drone-Renault-reveals-concept-car-traffic-spotting-quadcopter-built-roof.html.

[15] Anuj Puri, "A survey of unmanned aerial vehicles (UAV) for traffic surveillance", in: *Department of computer science and engineering, University of South Florida* 2005.

[16] Izhak Rubin and Runhe Zhang, "Placement of uavs as communication relays aiding mobile ad hoc wireless networks", in: *IEEE MILCOM*, 2007, pp. 1–7.

[17] Weihua Sun, Hirozumi Yamaguchi, Koji Yukimasa, and Shinji Kusumoto, "Gvgrid: A qos routing protocol for vehicular ad hoc networks", in: *IEEE IWQoS*, 2006, pp. 130–139.

[18] Nawaporn Wisitpongphan, Fan Bai, Priyantha Mudalige, Varsha Sadekar, and Ozan Tonguz, "Routing in sparse vehicular ad hoc wireless networks", in: *IEEE Journal on Selected Areas in Communications* 25.8 2007, pp. 1538–1556.

[19] Jie Wu, Shuhui Yang, and Fei Dai, "Logarithmic store-carry-forward routing in mobile ad hoc networks", in: *IEEE Transactions on Parallel and Distributed Systems* 18.6 2007, pp. 735–748.

[20] Qing Xu, Tony Mak, Jeff Ko, and Raja Sengupta, "Vehicle-to-vehicle safety messaging in DSRC", in: *ACM Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, 2004, pp. 19–28.

[21] Xue Yang, Jie Liu, NF Vaidya, and Feng Zhao, "A vehicle-to-vehicle communication protocol for cooperative collision warning", in: *IEEE MOBIQUITOUS*, 2004, pp. 114–123.

[22] Jing Zhao and Guohong Cao, "VADD: Vehicle-assisted data delivery in vehicular ad hoc networks", in: *IEEE Transactions on Vehicular Technology* 57.3 2008, pp. 1910–1922.