2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications

VehiCloud: Cloud Computing Facilitating Routing In Vehicular Networks

Yang Qin and Dijiang Huang
Arizona State University
Tempe, Arizona, USA
{yang.qin.1, dijiang}@asu.edu

Xinwen Zhang
Huawei Research Center
Santa Clara, CA USA
xinwen.zhang@huawei.com

***Abstract*—Establishing reliable routing among highly mobile vehicles is a challenging problem in vehicular networks. Towards this issue, we present *VehiCloud*, a novel cloud computing architecture that leverages emerging cloud computing technologies to deal with unreliable inter-vehicle communications and extend the restricted computational capabilities of mobile devices. A way-point information framework (WIF) is devised within the VehiCloud architecture, aiming to provide routing service for vehicular network, where each vehicle serves as a mobile service node and predicts its future locations by generating *way point* messages, which describe the trajectory of the vehicle's movement. A decision module in VehiCloud collects vehicles' way points and makes routing decisions for inter-vehicle communication. Selected paths of the routing are globally optimized in terms of message delivery ratio by respecting the constraints of end-to-end delay and communication cost. Our implementation of VehiCloud and real-road experiments demonstrate that it is practical and efficient to address fundamental routing problems for vehicular networks.**

***Index Terms*—vehicular networks; routing; mobile cloud;**

I. INTRODUCTION

Traditional vehicular networks are mostly established based on pure ad hoc communication. Consequently, inter-vehicle communication is unreliable because of the high mobility of vehicles. Moreover, a vehicle's communication and computation capabilities are constrained by the limited resources of its on-board devices. On the other side, the development of wireless communication technologies makes inter-vehicle communication systems heterogeneous. For example, with pervasive roadside infrastructures, vehicles can often access the Internet via Vehicle-to-Infrastructure (V2I) channels. In addition, with advanced wireless access technologies such as 3/4G, LTE, and WiMax, vehicles can access the Internet with longer communication distance than traditional short-distance and low-speed wireless communication technologies. Thus, we envision that vehicular network will eventually become a mobile extension of the Internet. Following this, mobile cloud computing (a.k.a., MobiCloud) is an emerging technique [1] [2] that has been fundamentally changing the research and development of mobile applications. With the light of this trend and maturing related technologies, we devise *VehiCloud*, a novel mobile cloud infrastructure for vehicular communication system.

In VehiCloud, each vehicle is treated as a Mobile Service Node (MSN), which monitors its own movement as well as predicts its future locations. As a result, the cloud can predict future traffic information by collecting the trajectory information of MSNs. Moreover, with the awareness of the dynamic network topology, VehiCloud helps reduce the routing and communication uncertainty, which is especially critical in vehicular network with high mobility. More than this, the cloud is also capable of utilizing MSNs' mobility and resource to carry and forward messages.

Routing in vehicular networks has been considered as a challenging task, with consideration of various factors, including vehicles' mobility patterns, the ability of vehicles to carry messages to different locations, and the availability of Internet connections at roadside infrastructure. Moreover, variant communication channels (e.g., ad hoc wireless channel and the Internet connection) have different levels of reliability in a location, which leads to various end-to-end delays and communication costs. A good routing strategy should take all these for an optimized and dynamic decision. Uniquely with cloud-based architecture, VehiCloud introduces a way-point information framework (WIF) to enable routing service for highly mobile and dynamic topologies in vehicular networks. In WIF, an MSN generates *way points* by anticipating its future locations. A cloud decision module (DM) collects the way points of MSNs through terminals (i.e., RSUs and vehicles with Internet access). With this information, the DM constructs a time-space link graph (TSLG) to model the entire network. The TSLG is not only a snapshot of the network topology. Instead, all topology changes within a certain time are captured by the TSLG. Assisted by the TSLG, the DM is then able to make globally optimal routing decisions for inter-vehicle communication in temporal and dynamic manner, empowered with the elastic computing resources of underlying cloud platforms.

This paper has the following contributions:

1) We design and implement VehiCloud, transforming traditional vehicular networks into a service-oriented architecture, and provide a potential platform to address several fundamental problems in vehicle networks including reliable routing. VehiCloud maximally takes the advantage of each vehicle in the system by utilizing cloud computing technologies;

2) We model the highly dynamic network topology as a time-space link graph (TSLG). A TSLG includes three types of communication links in vehicular networks: the ad hoc wireless channels, the "carry-and-forward" links (i.e., the operation of a vehicle carrying a message for some time and

978-0-7695-4745-9/12 $26.00 © 2012 IEEE
DOI 10.1109/TrustCom.2012.16

then forwarding it), and the Internet connections. Compared to a conventional network connectivity graph which is just a snapshot of a network topology, TSLG includes a time dimension. That is, the topology changes of the network within a certain duration are all mapped into a single graph;

3) We address the fundamental routing problem in vehicular networks with VehiCloud, by modeling routing as a linear program. The routing decision is globally optimized in terms of the message delivery ratio by considering the constraints on end-to-end delay and communication cost.

The rest of the paper is organized as follows. We describe the VehiCloud architecture in Section II, and present the details of the cloud routing service in Section III. In Section IV, we present experiment details and analyze the results. We present related work in Section V, and conclude this paper at the end. .

II. VEHICLOUD ARCHITECTURE OVERVIEW

We assume that every vehicle has an on-board unit (OBU), including a built-in navigation system (NS) (e.g., GPS), with maps and locations of available roadside units (RSUs). The NS is also capable of anticipating the approximate locations of the vehicle at certain points of time in the future, based on path calculation and driving history mining.

Both OBUs and RSUs have wireless communication devices. The RSUs are sparsely distributed in the transportation system. Some of the OBUs have broadband wireless communication capabilities, e.g., 3/4G cellular communication devices, WiMax, and they can always connect to the Internet. Some of the OBUs only have short range communication capabilities, e.g., WiFi or Wireless Access in Vehicular Environment (WAVE). These OBUs usually can communicate with each other or with RSUs within the range of up to 1,000 meters depending on the devices' capability and channel quality. In this hybrid vehicular networking environment, some public vehicles (e.g. buses, police cars, etc.) are equipped with broadband wireless devices. Other vehicles can use these public vehicles and RSUs as access points to access the Internet. These access points are called "terminals" in this article. The public vehicles are the "mobile terminals" while the RSUs are the "fixed terminals".

As we can see in Fig.1, OBUs and RSUs jointly form a vehicular ad hoc network. The terminals serve as access points to the Internet, upon which VehiCloud is established with the heterogeneous communication mechanisms. Each MSN, i.e., a vehicle, serves as a mobile agent, which predicts its own moving trajectory, monitors certain conditions in certain area, and performs other lightweight tasks such as message transmission as needed. For instance, the MSNs generate their "way points" by anticipating their future locations at certain future time points. On the other hand, these MSNs are free of computation and resource intensive tasks, such as data mining, machine learning, and large-scale optimization. These tasks are executed by a decision module empowered by traditional cloud computing clusters.

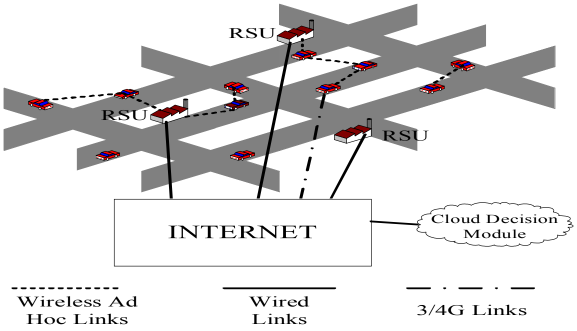


Fig. 1. Basic VehiCloud Architecture.

TABLE I
A SAMPLE WAY POINT

vid	gt	ft	fp	pb
123ABC	12:00pm	12:15pm	(x_1, y_1)	1.0

III. VEHICLOUD ROUTING SERVICE

In this section, we present how to establish the routing service in VehiCloud. To achieve our goal, we devise a way-point information framework (WIF) as follows.

A. Way points

A way point is essentially a pair of values: a given time and its corresponding position for a vehicle. The time can be the current time or a future time. The value of the position is typically a geographic location. That is, the way points tell the vehicle's current and future locations.

We formally define a way point as a tuple of five attributes: vehicle ID (vid), generation time (gt), future time (ft), future position (fp) and probability (pb). A sample way point is given in Table I. This way point is generated by the vehicle "123ABC" at 12:00pm. It describes that the vehicle will be at the location (x_1, y_1) at time 12:15pm and the probability for this prediction to be true is 1.

A vehicle usually generates a sequence of way points in a batch, namely a way-point vector (W-Vector). The way points in a W-Vector are generated for the (future) time points with a given time interval (usually a fixed time interval). Multiple way points may be generated for the same time, which means the vehicle would be at multiple possible locations at that time.

We say that a way point *expires* when its "future time" has passed. A way point becomes *inactive* when it expires, otherwise it is considered *active*. Two way points w_i and w_j are considered to be *adjacent* if and only if they are from the same vehicle, and for any way point w_k generated by the same vehicle, we have $w_k.ft \leq \min\{w_i.ft, w_j.ft\}$ or $w_k.ft \geq \max\{w_i.ft, w_j.ft\}$. That is, there is no way points generated by the same vehicle for the future time (ft) during the future time interval $[w_i, w_j]$.

After generating the way points (or a W-Vector), a vehicle (performed by its OBU) sends them to its nearest terminals. The way points can be directly sent to an encountered mobile terminal, or to a nearby fixed terminal using geographic routing. Here, we assume that every NS is aware of the exact locations of fixed terminals, as mentioned in Section II. Upon receiving a W-Vector, the terminal forwards it to the VehiCloud decision module, which will be presented in the next section.

B. Service Decision Module

Given the way points collected from vehicles and the locations of all fixed terminals, the cloud decision module (the DM, see Fig.1) is responsible for making the routing decision by constructing a time-space link graph (TSLG).

1) The Time-Space Link Graph: If the network is static, the network topology can easily be mapped to a simple link graph, with vertices representing the network nodes and edges representing the communication channels between these nodes. However, vehicular networks are not static. Vehicles usually move in high speed and the network topologies keep changing as time elapses. The TSLG $G = (V, E)$ is a unique way of handling time lapse and mobility in vehicular networks. It is designed to map the changing network topology (over a certain time period) onto a single graph. That is, the TSLG actually adds the time dimension into a normal network connectivity graph. In addition, the TSLG needs to model all possible information-transfer links as its edges, including “carry-and-forward” links (i.e., a vehicle carries the information from one location to another), ad hoc wireless links, and Internet connections. To illustrate the TSLG, we present an example in Fig.2, in which multiple vertices are created to represent a single entity (e.g., a vehicle) at different points of time (the vertices connected by vertically directional edges in Fig. 2). These vertices have completely different edges associated, reflecting the topology changes due to mobility at different time points. For example, two vertices representing two vehicles at time t_1 are disconnected, since at that time these two vehicles are not within each other’s communication range; however, the two vertices representing the same two vehicles at time t_2 are connected (by directional edges in both direction), which means the two vehicles meet each other and an ad hoc wireless channel between them exists at that time. On the other hand, all vertices representing a RSU should have edges associated which stand for the Internet connection available at all time.

In the following presentation, we describe the construction of the TSLG. In TSLG, a physical entity can be represented as multiple vertices in the graph. The edges between vertices that represent different entities stand for ”can-flow”, i.e., a message can flow from one physical entity to a different entity, and the edges between vertices that represent the same entity stand for ”can-carry”, i.e., the same physical entity can carry a message for a certain time period.

The vertex set V is comprised of several subsets:

$$V = \{v_c\} \cup M_s \cup M_r \cup W_O \cup W_T \cup F \cup I,$$

where v_c is a single vertex representing the cloud decision module (DM). Each entity (vehicle or fixed terminal) m has two vertices: $v_{m_s} \in M_s$ and $v_{m_r} \in M_r$, which stand for the entity as a sender and receiver, respectively. We call v_{m_s} the sender vertex and v_{m_r} the receiver vertex of m. Each way point w corresponds to a vertex $v_w \in W_O$. If w is generated by a mobile terminal, it also has a corresponding vertex $v'_w \in W_T$. In addition, at the graph construction time, the DM enumerates all different values of the active way points’ “future times”[1]. And for each future time point t, a vertex $v_t \in I$ is created to represent the Internet at that time. Similarly, a vertex $v_{t,f} \in F$ represents the instance of the fixed terminal (RSU) f at the future time point t.

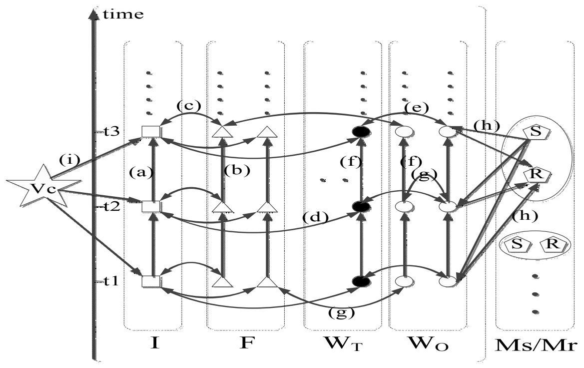


Fig. 2. The time-space link graph.

In TSLG, every vertex v has a probability p_v, which is defined as follows:

$$p_v \triangleq \begin{cases} 1 & \forall v \in V/W_O; \\ w.pb & \forall v_w \in W_O. \end{cases}$$

That is, the probabilities of vertices in W_O (representing the way points) are determined by the way points’ probability values. All other vertices have the probabilities of 1.

Each edge $e \in E$ has the following attributes attached: the probability (p_e), delay (d_e) and cost (k_e). First, we define that *the probability of any edge equals to the probability of its source (starting) vertex*:

$$p_{(v,w)} \triangleq p_v.$$

The edge set E is constructed according to the following rules. Please refer to Fig. 2 for examples of each rule.

(a) Let $I = \{v_{t_1}, v_{t_2}, ..., v_{t_n}\}$ where $t_1 < t_2 < ... < t_n$. We have

$$\{(v_{t_i}, v_{t_{i+1}}) | v_{t_i}, v_{t_{i+1}} \in I, 1 \leq i \leq n-1\} \subseteq E.$$

Every edge $e = (v_{t_i}, v_{t_{i+1}})$ is directed, pointing from v_{t_i} to $v_{t_{i+1}}$. This means that there is no backward link from a later point of time to an earlier time. In other words, a message cannot be sent back in time. For the delay and cost, we have $d_e = t_{i+1} - t_i$ and $k_e = 0$, respectively.

(b) Let $\{v_{t_1,f}, v_{t_2,f}, ..., v_{t_n,f}\} \subseteq F$ be the vertices representing the fixed terminal f at different points of time, where $t_1 < t_2 < ... < t_n$. We have

$$\{(v_{t_i,f}, v_{t_{i+1},f}) | v_{t_i,f}, v_{t_{i+1},f} \in F, 1 \leq i \leq n-1\} \subseteq E.$$

Data flow going through these edges means that the data is carried by the terminal (RSU) for a certain period. Thus, the

---

[1] Please refer to Section III-A for the definition of active way points.

edge delay is the interval between the two points of time, and the cost is 0: $d_e = t_{i+1} - t_i, k_e = 0$.

(c) There are two symmetric edges $e = (v_{t,f}, v_{t'})$ and $e^r = (v_{t'}, v_{t,f})$ between vertex $v_{t,f} \in F$ and vertex $v_{t'} \in I$ if and only if $t = t'$. That is,

$$\{(v_{t,f}, v_{t'}), (v_{t'}, v_{t,f}) | v_{t,f} \in F, v_{t'} \in I, t = t'\} \subseteq E.$$

These edges represent the Internet connections at RSUs. The edge delay and cost are determined by the properties of the local Internet connections: $d_e = d_{e^r} = D_I, k_e = k_{e^r} = K_I$, where we use D_I and K_I to denote the delay and cost of the local Internet connection, respectively. Though we use the same notations, the values of them can vary at different RSUs.

(d) Similar to RSU, each mobile terminal has Internet access. Thus, there are also two symmetric edges $e = (v_w, v_t)$ and $e^r = (v_t, v_w)$ between vertex $v_w \in W_T$ and vertex $v_t \in I$ if and only if $w.ft = t$, where $w.ft$ is the future time of the way point w. Then, we have:

$$\{(v_w, v_t), (v_t, v_w) | v_w \in W_T, v_t \in I, w.ft = t\} \subseteq E.$$

The edge delay and cost are determined by the quality of the Internet connections: $d_e = d_{e^r} = D_I, k_e = k_{e^r} = K_I$. Similarly, D_I and K_I may have different values for different moving terminals. For example, different ISPs may offer different prices for the 3/4G connections.

(e) Recall that every way point w generated by a mobile terminal has two corresponding vertices, $v_w \in W_O$ and $v'_w \in W_T$. These two vertices have two symmetric edges between themselves:

$$\{(v_w, v'_w), (v'_w, v_w) | v_w \in W_O, v'_w \in W_T\} \subseteq E.$$

Obviously, the edge delay and cost are both 0 in this case: $d_e = 0, k_e = 0$.

(f) A moving vehicle can carry a message from one spot to another and then forward it to others. Such "carry-and-forward" links are modeled by the edges between adjacent way points[2]. For any two vertices $v_{w_i}, v_{w_j} \in W_O$ or $v_{w_i}, v_{w_j} \in W_T$, there is an edge $e = (v_{w_i}, v_{w_j})$ from v_{w_i} to v_{w_j} if and only if w_i and w_j are adjacent and $w_i.ft \leq w_j.ft$. Note that the two adjacent way points may be generated within a single W-Vector or two different W-Vectors with the same vehicle ID (vid). That is,

$$\{(v_{w_i}, v_{w_j}) | v_{w_i}, v_{w_j} \in W_O \textit{ or } v_{w_i}, v_{w_j} \in W_T,$$
$$w_i.ft \leq w_j.ft, w_i \textit{ and } w_j \textit{ are adjacent}\} \subseteq E.$$

In this case, the delay of the edge is set to the time duration between the future times of w_i and w_j, and the cost of the edge is set to 0: $d_e = w_j.ft - w_i.ft, k_e = 0$.

(g) Two vehicles located within each other's communication range can have a direct link between them. Therefore, there should be two symmetric edges e and e^r between two vertices in one of the following two cases: (i) the two vertices correspond to two way points generated by two different vehicles, they have the same future time, and the geographic distance between their future positions is less than R; (ii) one of them represents a RSU instance at a certain point of time, the other corresponds to a way point of the same future time, and the distance between the way point's future position and the RSU is less than a communication threshold R. Thus, we have

$$\{(v_{w_i}, v_{w_j}), (v_{w_j}, v_{w_i}) | v_{w_i}, v_{w_j} \in W_O, w_i.vid \neq w_j.vid,$$
$$w_i.ft = w_j.ft, |w_i.fp, w_j.fp| < R\} \subseteq E,$$

and

$$\{(v_w, v_{t,f}), (v_{t,f}, v_w) | v_w \in W_O, v_{t,f} \in F,$$
$$w.ft = t, |w.fp, pos_f| < R\} \subseteq E,$$

where pos_f denotes the position of the RSU f. In these two cases, the edge delay is set to the transmission delay (D_H) of a direct wireless channel, and the edge cost if set to 0: $d_e = d_{e^r} = D_H, k_e = k_{e^r} = 0$.

(h) For each vehicle m (whose ID is vid_m), there are edges from the sender vertex $v_{m_s} \in M_s$ to all its way points, and edges from all its way points to the receiver vertex $v_{m_r} \in M_r$:

$$\{(v_{m_s}, v_w) | v_{m_s} \in M_s, v_w \in W_O \cup W_T,$$
$$w.vid = vid_m\} \subseteq E,$$
$$\{(v_w, v_{m_r}) | v_{m_r} \in M_r, v_w \in W_O \cup W_T,$$
$$w.vid = vid_m\} \subseteq E.$$

Similarly , for each fixed terminal f, there are edges from the sender vertex $v_{f_s} \in M_s$ to all its vertices in F, and edges from all its vertices in F to the receiver vertex $v_{f_r} \in M_r$:

$$\{(v_{f_s}, v_{t,f}) | v_{f_s} \in M_s, v_{t,f} \in F\} \subseteq E,$$
$$\{(v_{t,f}, v_{f_r}) | v_{f_r} \in M_r, v_{t,f} \in F\} \subseteq E.$$

These edges have 0 delay and cost: $d_e = 0, k_e = 0$.

(i) Lastly, there is an edge e from the vertex of the DM (v_c) to every vertex representing the Internet:

$$\{(v_c, v_t) | v_t \in I\} \subseteq E.$$

The delay and cost of the edge depends on the property of the Internet connection (as described in (c) and (d)): $d_e = D_I, k_e = K_I$. Since the DM does not provide data transmission service and it only disseminates routing decisions to vehicles, there are no reversed edges back to the DM.

The TSLG construction strategy described above assumes that the message transmission delay (over an Internet connection or wireless channel) has negligible impact on the network topology. For example, if vehicle A is directly connected to RSU_a when vehicle B is directly connected to RSU_b, we assume that if A sends a message to RSU_a who then sends the message to RSU_b via Internet, B can still receive the message from RSU_b because it has not moved out of RSU_b's communication range. The assumption is usually sound, considering the vehicles' moving velocity and communication

---

[2] See Section III-A for the definition of adjacent way points

distance, and the transmission delay for short messages. For example, transmitting a message of 1024 bytes via an 11mbps wireless channel normally takes less than 1 millisecond. A vehicle in the speed of 80mph can only move less than 0.04 meters in 1 millisecond, which is definitely negligible since the communication range of a vehicle is usually larger than 100 meters.

We need to emphasize that the structure of the TSLG changes as time elapses. When old way points expire, the corresponding vertices and the associated edges need to be removed from the graph. The new way points also need to be incorporated into the graph as soon as they are collected. Note that the set of active way points also affect the subset of vertices representing the Internet and RSUs (set I and F).

2) Optimal Routing: Once the TSLG G is constructed, the DM can model routing as a path-searching problem. First, the DM needs to determine the source vertex s and destination vertex t according to the actual information source and destination of the requested session. If the actual information source is the entity m (m can be a vehicle or a fixed terminal), the source vertex s must be $v_{m_s} \in M_s$; similarly, if the actual destination is the entity m', the destination vertex t must be $v_{m'_r} \in M_r$. Given the s and t, the size of the data to be transferred denoted as $size$, the maximum acceptable end-to-end delay denoted as D_{max}, and the budget (maximum acceptable cost) denoted as K_{max}. Our goal is to find a $s-t$ path on G such that the path has the highest message delivery ratio and satisfies the given delay and cost constraints. In order to model this routing problem, we need to use the following definitions:

Message Delivery Ratio: The message delivery ratio $dratio_r$ of a path $r = (e_1, e_2, ..., e_n)$ is the joint probability of all edges on the path:

$$dratio_r = \prod_{i=1}^{n} p_{e_i}.$$

Here, we assume all edges are independent from each other.

End-to-end Delay: The end-to-end delay $delay_r$ of a path $r = (e_1, e_2, ..., e_n)$ is the summation of the delay of all edges on the path:

$$delay_r = \sum_{i=1}^{n} d_{e_i}.$$

Path Cost and Message Delivery Cost: The cost $cost_r$ of a path $r = (e_1, e_2, ..., e_n)$ is the summation of the cost of all edges on the path. The cost of delivering a message m via the path r, say $cost_{m,r}$ is the product of the path cost and the size of the message to be delivered:

$$cost_r = \sum_{i=1}^{n} k_{e_i}, \qquad cost_{m,r} = cost_r \times sizeof(m).$$

In addition, given that the logarithm function is monotonically increasing, we know that maximizing $dratio_r$ is equal to maximizing $\log(dratio_r)$. Hence, we can model the complex vehicular network routing problem as a constrained shortest path problem, i.e., to find a $s-t$ path $r = (e_1, e_2, ..., e_n)$ that:

$$minimizes \ \sum_{i=1}^{n} -\log(p_{e_i}), \quad s.t.$$

$$delay_r = \sum_{i=1}^{n} d_{e_i} \le D_{max};$$

$$cost_r = \sum_{i=1}^{n} k_{e_i} \le K_{max}/size.$$

We can use a linear program solver to solve this constraint shortest path problem in polynomial time. The resulting $s-t$ path is a globally optimized route in terms of the message delivery ratio, with a bounded end-to-end delay D_{max} and message delivery cost K_{max}. The path may contain "carry-and-forward" links, wireless ad hoc links, as well as Internet connections.

IV. EXPERIMENTS AND RESULTS

We conduct real-world experiments to demonstrate how the VehiCloud architecture can be used to address the fundamental routing problem. We first present the experiment setup and then analyze the results.

A. Experiment Setup

1) Environment: The experiment is conducted in a residential area. The street map is shown in Fig. 3. The roads taken by our vehicles are highlighted by red lines. In the map, each of three horizontal streets is approximately 400 meters in length, and the distance between them is about 100 meters. Between streets, the residential area is packed by buildings. The local traffic within this experiment area is not very high, thus our experimental vehicles can move in a relatively constant speed.

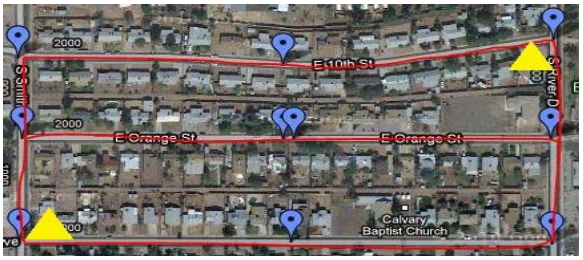


Fig. 3. Experiment Environment.

2) Vehicles: There are 10 vehicles involved in our experiment. Their starting positions are highlighted using blue tags (round shape) in the map. Each of the vehicles carried an Android phone, on which we implemented both the WIF and a position based routing (PBR) scheme [3]. These phones are all capable of receiving GPS signals to obtain their locations.

TABLE II
EXPERIMENT SCENARIOS

Scenario	A	B	C	D	E	F
Vehicle Speed (mph)	20	20	20	30	30	30
Communication Range (m)	250	200	150	250	200	150

3) Network setup: The vehicular network is comprised of the 10 MSNs (i.e., vehicles) and 2 fixed nodes (RSUs, emulated by 2 stationary phones). The positions of the two RSUs are marked as yellow triangles in the map. When WIF is in use, one of the 10 vehicles is randomly selected as a mobile terminal. The RSUs and the mobile terminal had Internet access, so they could directly talk with the cloud decision module (DM), which is not shown in the map. The ad hoc communication channels are emulated through 3G connections via Android phones. At a certain point of time, if and only if the distance between two nodes was less than the current setting of the communication range, a direct ad hoc link was assumed to exist between them. But when they actually communicated with each other, the messages were transmitted via the 3G network. All these nodes receive commands from a centralized server and report their performance data, e.g., packet drop, to the server for post-experiment analysis.

4) Experiment Scenarios: We conduct a comprehensive set of experiments including 6 different scenarios to evaluate our solution. These scenarios are summarized in Table II.

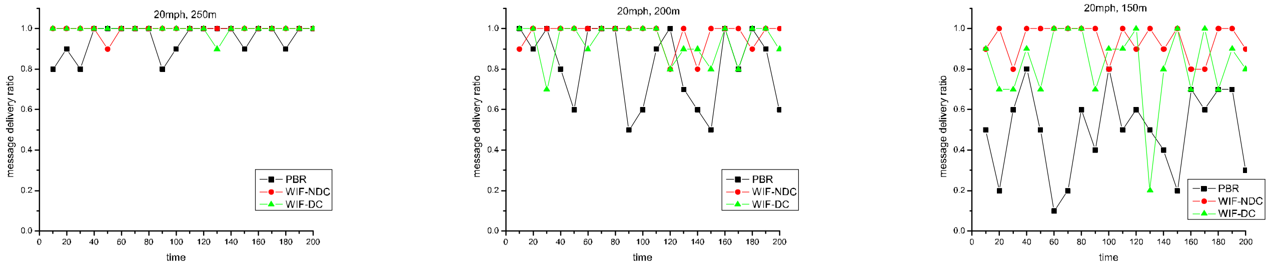
Each scenario has three rounds, and each round lasts for 5 minutes. For each scenario, PBR is used in the first round and WIF-based strategies are used in the other two rounds. In the second round, no delay constraint was applied (denoted as WIF-NDC) when making routing decisions; while in the third round, a 30-second delay constraint was applied (denoted as WIF-DC). For each round, data transmission is started from the 50th second and ended at the 250th second. During the data transmission period, for each second, we (at the central server) randomly choose a vehicle to initiate a message and send it to another randomly chosen vehicle. Note that at the beginning of each round, the vehicles start from their initial locations highlighted in the map, and follow the same route in the same moving speed to minimize the experimental inconsistency.

5) WIF settings: In the implemented WIF, the time interval between two adjacent way points in a W-Vector is 5 seconds. These way points are computed based on the vehicle's moving speed and its predetermined moving trajectory. Without losing generality, cost was not considered as a constraint in either WIF-NDC or WIF-DC for simplicity.

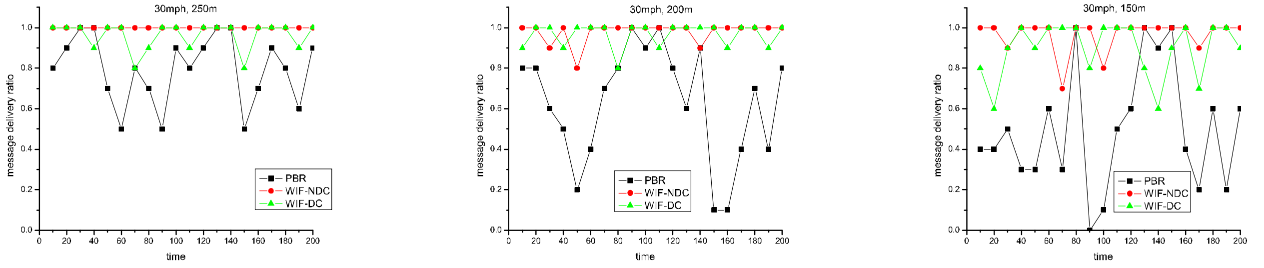
B. Experiment Results

The experimental results are depicted in Fig. 4 and Fig. 5. Fig. 4 compares the message delivery ratios of the three different routing strategies (PBR, WIF-NDC, and WIF-DC) in different scenarios, over the 200-second period of data transmission (i.e., 50th -250th second). The message delivery ratios are calculated (and marked in the figures) for every 10 seconds. First of all, we can clearly see that, both WIF-NDC and WIF-DC outperform PBR significantly in all cases. PBR performs well when the communication range is large (250 meters) and the vehicle speed is low (20mph) as shown in Fig. 4(a). However, PBR is too sensitive to both the two factors, especially the communication range. For example, when the communication range decreases at 150 meters (see Fig. 4(c) and Fig. 4(f)), the performance of PBR is extremely poor; the message delivery ratio is lower than 60% for most of the time and the behavior is totally unpredictable. This is because PBR highly relies on the connectivity of the network. When the communication decreases and the vehicle speed increases, the network connectivity cannot be easily maintained, and vehicles can get disconnected more frequently. As a result, PBR often fails due to vehicles not being able to find the next hop. On the contrary, we find that WIF-NDC is the most reliable routing strategy. The message delivery ratio of WIF-NDC is usually greater than 80% and up to 100%. The reason for such behavior is that WIF-NDC has no delay constraint when making routing decisions. Thus, the routes selected have more considerations on the route reliability. For example, no matter how fast the vehicles move and how small the communication range is, WIF-NDC will still find the chances for two vehicles to communicate directly with each other when they meet on the street. Moreover, the source vehicle can also leave the message to an RSU when it passes by the RSU, which can then deliver it to the destination at a later time. These behaviors can hardly be affected by the vehicle speed and communication range. In the case of WIF-DC, we add a 30-second delay constraint when making routing decisions, which makes it more sensitive to vehicles' communication range than WIF-NDC (but still much better than PBR). For instance, comparing Fig. 4(a), 4(b) and 4(c), we can see that the message deliver ratio of WIF-DC has a notable drop when vehicles' communication range decreases. Likewise, Fig. 4(d), 4(e) and 4(f) demonstrate the same behaviors. The reason for this phenomenon is straightforward: since a 30-second delay constraint is applied, highly reliable "carry-and-forward" links cannot be used as much as in WIF-NDC. In this case, WIF-DC has to use multi-hop ad hoc links to forward messages within the delay constraint. Nonetheless, we claim that WIF-DC is still reliable since at most of time, its message delivery ratio is still as high as 80%. We also notice that, as shown in Fig. 4(c) and Fig. 4(f), the decreasing of the communication range has less impact on WIF-DC's performance when the vehicles move faster. The reason is that an encounter between two vehicles appears fasters when the vehicles move faster, and so the delay constraint has less impact.

In Fig. 5, we present the results about the average hop counts in different scenarios. It is necessary to point out that, we ignore the hops of the "query" phase but only focus on the hops taken by the actual message transfer. Using PBR, every node, including the source and relay nodes, has to send a query to a location server to get the location information of the destination. However, in WIF, only the source needs to send a query to the DM to get the route to the destination, which can dramatically reduce the overhead during the route query phase. However, since there are many different implementations of



(a) Scenario A (b) Scenario B (c) Scenario C



(d) Scenario D (e) Scenario E (f) Scenario F

Fig. 4. Message Delivery Ratio

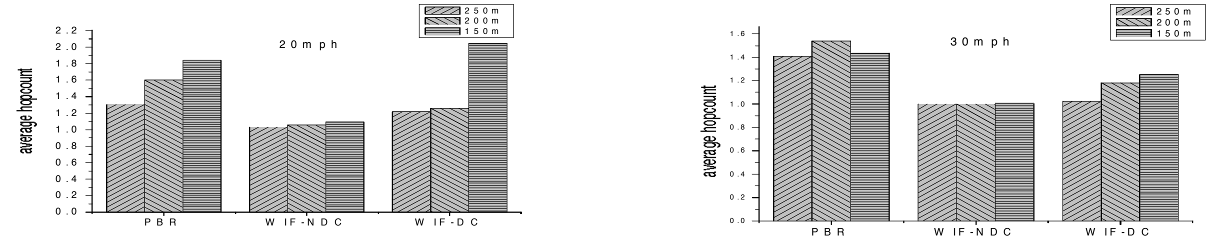
location services, it is not fair to compare the query phase using some specific strategy for PBR. Thus, in our comparative study of hop counts, we only focus on the message transfer, in spite of the location query (PBR) and path query (WIF). We observe that the hop count of WIF-NDC usually equals to 1 regardless the different settings of vehicles' moving speed and communication range.

WIF-NDC always selects shorter paths. This is because a path with less number of hops has higher reliability. For instance, if vehicle A wants to send a message to vehicle B, it is always better for A to carry the message until it meets with B rather than send the message to B via relay nodes, since the probability of "carry-and-forward" links is equal to 1 while that of the ad hoc links is less than 1. However, WIF-DC will select longer paths due to the delay constraint. This is because a vehicle cannot carry the message for a long time, and thus it has to send the message to the destination via available relay nodes to satisfy the delay requirements. Hence, the average hop count of WIF-DC increases notably when the vehicles' communication range decreases. In Fig. 5(a), we can see that the average hop count of WIF-DC is the highest among the three routing strategies when the communication range is decreased to 150 meters. In comparison, we do not notice such phenomenon in Fig. 5(b), which represents vehicles moving in a higher speed. This is because the high moving speed makes vehicles meet sooner, which reduces the impact of the delay constraint. Additionally, we observe that in Fig. 5(b), PBR has a lower hop count when the communication range decreases to 150 meters, which is contradictory to Fig. 5(a). That is actually because the average hop count is calculated for the messages that are successfully delivered. In Scenario F (30mph, 150m), PBR has a very lower message delivery ratio (see Fig. 4(f)). The low hop count tells us that PBR cannot survive in multi-hop message transfer in this case.

V. RELATED WORK

Recent development of mobile cloud computing [1], [2] constructed a new service oriented framework that recruits mobile devices as service providers to build a sensing-based new application platform. In such a framework, each mobile device (usually an embedded device) is a service provider. An embedded device senses its surrounding information, such as wireless communication channel status, neighboring nodes information, environmental information (e.g., CO2 and pollution levels, etc.), personal information (e.g., medical and health information using bio sensors), etc. Mobile cloud computing is an emerging research area. How to construct a mobile cloud system to assist vehicular communication is still an unexplored area.

Existing information dissemination strategies in vehicular networks can be classified into three categories: restricted flooding, "carry-and-forward", and geographic routing. In order to alleviate the overhead incurred by flooding, improved schemes such as [4], [5] and [6] have been proposed to restrict the flooding to certain extents. The "carry-and-forward" strategy is commonly used for delay tolerant information propagation. Generally, messages are carried by an intermediate vehicle until it finds the opportunity to forward the messages further. By utilizing the predictable mobility of vehicles, the messages will ultimately be transmitted to the destination. Proposals such as MDDV [7], VADD [8], SADV



(a) 20mph

(b) 30mph

Fig. 5. Average Hopcount

[9], Ferry [10] and ZebraNet [11] all applied the similar carry-and-forward idea. However, these mechanisms can only send messages to fixed destinations such as roadside units, but not moving vehicles; and they usually cause large end-to-end delay. Geographic routing or position based routing (PBR) is capable of disseminating information to mobile destinations when their locations are known. As presented in [3], [12], [13] and [14], PBR does not rely on the network topology. Routing decisions are made at each hop by selecting the next hop that is closest to the destination in terms of the geographic distance. The prerequisite of applying geographic routing is that the source vehicle and all the relays have to know the exact geographic location of the destination in real-time. This is hard to be satisfied because knowing the location of the destination is difficult in vehicular networks. Protocols such as DLS, SLS and RLS presented in [15] provide location services by periodical location information exchange either proactively or reactively across the entire network. Other protocols as GLS [16] use grid hierarchy to maintain location information in real-time. However, all these solutions can hardly be applied to fast moving vehicles. They usually incur high communication overhead when generating large amount of location updates, and experience enormous number of location lookup failures.

VI. CONCLUSION

In this paper, we presented VehiCloud, a cloud assisted architecture for inter-vehicle routing. Our approach transforms the conventional vehicular communication system into a service-oriented architecture by extending the concept of mobile cloud into vehicular networks. VehiCloud treats vehicles as both the mobile service nodes (MSNs) of the cloud and end users of cloud services. We elaborate a concrete cloud-based routing service to address the fundamental routing problem for inter-vehicle communication. Our real-world experiment demonstrates that, the VehiCloud routing service outperforms previous solutions significantly, in terms of the route reliability. In addition, the experiment also shows the ability of VehiCloud on making the routing decision by considering variant constraints.

ACKNOWLEDGEMENT

This research is sponsored by ONR YIP. Some of the work is also partially sponsored by NSF CNS-1029546 and DUE-0942453. The research outcomes do not necessarily reflect the views of sponsors.

REFERENCES

[1] D. Huang, “MobiCloud: A Secure Mobile Cloud Computing Platform,” *E-Letter of Multimedia Communications Technical Committee (MMTC), IEEE Communications Society (invited paper)*, 2011.

[2] D. Huang, X. Zhang, M. Kang, and J. Luo, “Mobicloud: A secure mobile cloud framework for pervasive mobile computing and communication,” in *Proceedings of 5th IEEE International Symposium on Service-Oriented System Engineering*, 2010.

[3] B. Carp and H. Kung, “GPSR: Greedy Perimeters Stateless Routing for Wireless Network,” *Proceedings of ACM/IEEE Mobicom*, pp. 243–254, 2000.

[4] R. Yadumurthy, A. CH, M. Sadashivaiah, and R. Makanaboyina, “Reliable MAC broadcast protocol in directional and omni-directional transmissions for vehicular ad hoc networks,” *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, pp. 10–19, 2005.

[5] G. Korkmaz, E. Ekici, F. Özgüner, and Ü. Özgüner, “Urban multi-hop broadcast protocol for inter-vehicle communication systems,” *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pp. 76–85, 2004.

[6] C. Maihöfer, T. Leinmüller, and E. Schoch, “Abiding geocast: time–stable geocast for ad hoc networks,” *Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, pp. 20–29, 2005.

[7] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, “MDDV: a mobility-centric data dissemination algorithm for vehicular networks,” *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pp. 47–56, 2004.

[8] J. Zhao and G. Cao, “VADD: Vehicle-assisted data delivery in vehicular ad hoc networks,” *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1910–1922, 2008.

[9] Y. Ding, C. Wang, and L. Xiao, “A static-node assisted adaptive routing protocol in vehicular networks,” *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*, pp. 59–68, 2007.

[10] W. Zhao, M. Ammar, and E. Zegura, “A message ferrying approach for data delivery in sparse mobile ad hoc networks,” *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pp. 187–198, 2004.

[11] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, “Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet,” *ACM SIGPLAN Notices*, vol. 37, no. 10, pp. 96–107, 2002.

[12] R. Jain, A. Puri, and R. Sengupta, “Geographical routing using partial information for wireless ad hocnetworks,” *Personal Communications, IEEE [see also IEEE Wireless Communications]*, vol. 8, no. 1, pp. 48–57, 2001.

[13] Y. Ko and N. Vaidya, “Location-Aided Routing (LAR) in mobile ad hoc networks,” *Wireless Networks*, vol. 6, no. 4, pp. 307–321, 2000.

[14] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward, “A distance routing effect algorithm for mobility (DREAM),” *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 76–84, 1998.

[15] T. Camp, J. Boleng, and L. Wilcox, “Location information services in mobile ad hoc networks,” *Communications, 2002. ICC 2002. IEEE International Conference on*, vol. 5, 2002.

[16] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris, “A scalable location service for geographic ad hoc routing,” *Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 120–130, 2000.