Community-based Bus System as Routing Backbone for Vehicular Ad Hoc Networks

Fusang Zhang*†‡, Hai Liu‡, Yiu-Wing Leung‡, Xiaowen Chu‡ and Beihong Jin*
*State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences
† University of Chinese Academy of Sciences
‡Dept of Computer Science, Hong Kong Baptist University

Abstract—**Low delivery latency and high delivery ratio are two key goals in the design of routing schemes in Vehicular Ad Hoc Networks (VANETs). The existing routing schemes utilize real-time information (e.g., geographical position and vehicle density) and historical information (e.g., contacts of vehicles), which usually suffer from a long delivery latency and a low delivery ratio. Inspired by the unique features of bus systems such as wide coverage, fixed routes and regular service, we propose to use the bus systems as routing backbones of VANETs. In this work, we present a Community-based Bus System (CBS) which consists of two components: a community-based backbone and a routing scheme over the backbone. We collect real traces of 2515 buses in Beijing and build a community-based backbone by applying community detection techniques in the Beijing bus system. A two-level routing scheme is proposed to operate over the backbone. The proposed routing scheme performs sequentially in the inter-community level and the intra-community level, and is able to support message delivery to both mobile vehicles and specific locations/areas. Extensive experiments are conducted on the real trace data of the Beijing bus system and the results show that CBS can significantly lower the delivery latency and improve the delivery ratio. CBS is applicable to any bus-based VANETs.**

Keywords—VANETs; bus systems; backbone; routing;

I. INTRODUCTION

A vehicular ad hoc network (VANET) consists of a set of mobile vehicles equipped with dedicated short-range communication (DSRC) devices, which enable inter-vehicle communications and the communications between vehicles and roadside units (RSUs). Routing in VANETs is a very challenging task due to high-speed mobility and dynamic network topologies. Extensive work has been done in the design of routing schemes in VANETs. The existing work could be classified into three categories. The work in the first category is to deliver messages from source vehicles to specific geographical locations (i.e., vehicle → location), which can support various location-based applications such as geographic advertising [1], delivery of parking information [2] and tourist points of interest [3]. The work in the second category is to deliver messages from source vehicles to destination vehicles (i.e., vehicle → vehicle), which is often used for data collection and information sharing [4]. The work in the third category is to disseminate messages, e.g., emergency messages and traffic alert messages [5], in a specified area. A broadcast operation is usually performed in the routing schemes in this category and the challenge is to tackle the broadcast storm problem [6] [7].

Low delivery latency and high delivery ratio are two key design goals of routing schemes in VANETs. In the existing solutions, delivering a message from one vehicle to another is usually determined based on either real-time information [8] [9] [10] [11] [12] or historical information [13] [14] [15] [16] [17]. With the former strategy, a vehicle holding a message selects its next-hop relay vehicle based on the real-time information such as geographical position, vehicle density and moving direction. This strategy performs well in dense VANETs but suffers from a long delivery latency and a low delivery ratio in sparse networks due to the lack of global optimization. The latter strategy utilizes historical information of vehicles to estimate the occurrences of their contacts in future. A vehicle delivers its message to the relay vehicle with the largest chance in contact with the destination vehicles. Notice that the contacts of vehicles are not on a regular/routine basis but random in practice. Two vehicles that contacted previously may not contact again in the near future. Thus, this strategy could result in a long delivery latency and failure of message delivery.

To tackle the aforementioned problems, some work [10] [18] proposed to deploy RSUs at road intersections and bus stops so as to provide message relay for vehicles. However, their routing efficiencies are limited by the number and locations of RSUs and it incurs considerable cost in the deployment and management of the RSUs [19]. In this work, we propose to utilize bus systems as routing backbones of VANETs without RSUs. We study the bus system in Beijing, China, where there are 21293 buses of 989 bus lines in total. We collect real traces of 2515 buses (from 1 Mar 2013 to 31 Mar 2013) and conduct extensive analysis of the traces. We find that there are several advantages in using bus systems as routing backbones of VANETs.

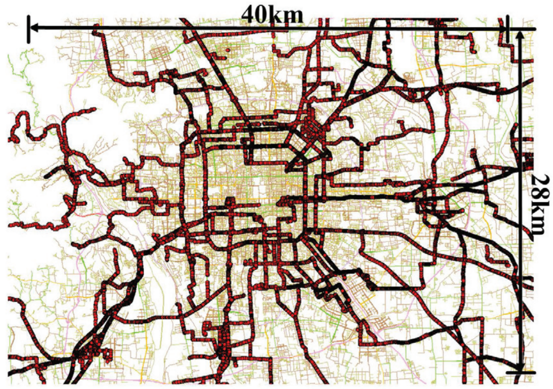


Fig. 1: One-day traces of 2515 buses in Beijing. The bold lines denote the aggregated traces of the buses.

- *Wide coverage*. We plot the traces of the 2515 buses in Beijing in Figure 1. It is clear to see that the traces form a backbone of Beijing city. Therefore, it is feasible to use bus systems as routing backbones for message delivery in VANETs.

- *Fixed routes*. Compared with routes of other vehicles (e.g., taxies), the routes of buses are normally fixed. This unique feature of bus systems enables us to map a specific location/area to fixed routes of buses. For example, the route of bus line No. 944 passes by the Beijing Olympic Stadium (i.e., the Bird's Nest). The messages destined for the Bird's Nest area can be delivered by the buses of line No. 944.
- *Regular service*. The service of a bus line is regular. For example, bus line No. 988 starts and stops its service at 5am and 10pm, respectively, in Beijing. If service hours and fixed routes of two bus lines overlap, the contact of the buses from these two bus lines is very likely to occur and thus message delivery among these buses is highly predictable.

In this work, we propose a Community-based Bus System (CBS) as routing backbone of VANETs. The idea of CBS originates from our analysis of real traces of 2515 buses in Beijing. Specifically, CBS is composed of a community-based backbone and a routing scheme over the backbone. We first build a contact graph which shows the closeness relation of bus lines. We notice that some buses are "closer" than others in terms of the frequency of contacts and some bus lines are "more active" in connecting other bus lines. Inspired by the concept of social networks, we apply community detection techniques in the contact graph to build a community graph which identifies potential communities of the Beijing bus system. A backbone graph is derived from the community graph by mapping the fixed routes of bus lines to the real map. Based on the community-based backbone, a two-level routing scheme is proposed to deliver messages to either a mobile vehicle or a specific location. The two-level routing scheme operates sequentially in the inter-community level and the intra-community level on the backbone. Our proposed solution is applicable to any bus-based VANETs.

The main contributions of this work are summarized as follows.

- We analyze the real traces of 2515 buses in Beijing and discover a strong community structure in the bus lines of these buses.
- We propose to utilize bus system as a routing backbone of VANETs and build a community-based backbone by applying community detection techniques of social networks.
- We propose a two-level routing scheme that operates on the community-based backbone. The proposed routing scheme is able to support message delivery to both mobile vehicles and specific locations/areas.
- We conduct extensive experiments on the real bus traces. The experimental results show that our proposed solution CBS can significantly lower the delivery latency and increase the delivery ratio, compared to the existing solutions.

The rest of this paper is organized as follows. Related work is reviewed in Section II. We analyze the traces of the Beijing bus system in Section III. We apply community detection techniques to build a community-based backbone in Section IV, and propose a two-level routing scheme in Section V. The experimental results are presented in Section VI. Finally, we conclude our work in Section VII.

II. RELATED WORK

VANET is a kind of mobile ad hoc network (MANET) and is essentially a delay tolerant network (DTN). We first review existing routing schemes in VANETs and then discuss relevant work in MANETs and DTNs. The differences between our solution and the existing solutions are summarized in Table I.

A. Routing Schemes in VANETs

Basically, there are two strategies in the design of routing schemes in VANETs. The first strategy is to use real-time information of vehicles such as geographical position, vehicle density and moving direction. GSR [8] is a typical position based greedy routing scheme in which a vehicle sends messages to a neighboring vehicle that is closer to the destination than itself. A similar idea was used in GPCR [9]. It chooses a neighboring vehicle whose geographical position is at the intersection or closest to the destination, and forwards the message to this neighbor. In addition to the geographical position, traffic information are also considered in existing routing schemes. For example, VADD [11] proposes a stochastic model based on vehicular traffic information which aims to minimize the message delivery latency. TBD [12] utilizes traces of vehicles and the traffic information (e.g., vehicle speed and vehicle density) to improve the performance of data forwarding. A localized algorithm is presented to compute the expected data delivery delay (EDD) at individual vehicles to an access point. The computed EDD is shared with neighboring vehicles and the vehicle with the smallest EDD is selected as the next carrier.

The other strategy is to utilize historical information including contacts and traces of vehicles. MaxProp [13] builds a bus network in the UMass Amherst campus and estimates the delivery likelihood, i.e., the probability of contact between buses. However, the testbed of MaxProp is composed of 30 buses only. BLER [14] studies contact length between different bus lines where the contact length is defined as the length of overlapping routes of these bus lines. A routing path is computed from one bus line to another such that the sum of contact length of the path is maximized. Similar to BLER, R2R [15] calculates the frequencies of contacts between bus lines based on historical traces and then utilizes them to decide the routing paths. A recent work, ZOOM [16], considers the contact-level mobility and the social-level mobility in the message delivery. A message is relayed by the vehicle with the shortest contact delay to the destination of the message. If information of contact delay to the destination is not available, the message will be delivered to a popular vehicle that has high centrality in the social level, because the popular vehicle can contact more vehicles and can have more opportunities for message forwarding. Another recent work called GeoMob [20] determines routes based on the traces of vehicles. GeoMob captures the traffic volumes in different regions and uses the K-means clustering method to construct clustered regions. The route is selected to pass through the regions with high traffic volumes. The routing from one region to another is determined based on the mobility patterns of individual vehicles. The

TABLE I: Differences between our solution and the existing solutions

		Consider msg delivery in bus systems	Build backbones on bus systems	Use real traces of vehicles	Apply community detection techniques of social networks	Support msg delivery to a mobile vehicle	Support msg delivery to a specific area
Our Solution (CBS)		√	√	√	√	√	√
VANETs	GSR/GPCR [8] [9]	×	×	×	×	×	√
	VADD/TBD [11] [12]	×	×	×	×	×	√
	MaxProp [13]	√	×	√	×	√	×
	BLER/R2R [14] [15]	√	×	√	×	√	×
	ZOOM [16]	×	×	√	√	√	×
	GeoMob [20]	×	×	√	×	×	√
MANETs	GeoTORA [21]	×	×	×	×	×	√
	GeoGrid [22]	×	×	×	×	×	√
	SimBet [23]	×	×	×	√	×	×
	BUSNet [24]	√	√	×	×	√	×
DTNs	BUBBLE [25]	×	×	×	√	×	×
	RCM [17]	√	×	√	×	√	×
	WiFi-enabled DTN [18]	√	√	×	×	×	√

message is delivered to the vehicles with higher probabilities of going to the destination region. However, the traffic volumes in the clustered regions may vary significantly at different time of a day due to the unpredictable behaviors of vehicles (e.g., taxies). Thus, the clustered regions are dynamic and should be updated frequently which incurs considerable overhead of maintenance.

B. Routing Schemes in MANET and DTN

There are several relevant routing schemes designed for mobile ad hoc networks (MANETs). GeoTORA [21] and Geogrid [22] are proposed to route messages to a specific location. Notice that the moving speeds of nodes in MANETs are usually slow. It is feasible to update the network topology upon each of routing requests. In contrast, vehicles move with high speeds in VANETs and the network topologies change rapidly. Thus, GeoTORA and Geogrid are not suitable for VANETs. SimBet [23] is a social based routing scheme in MANETs, which employs the similarity and betweenness centrality of nodes. Messages are routed to most central nodes until a node with higher similarity is met. Then the packet is routed within the community until the destination is reached. BUSNet [24] builds a M-GRID mobility model in vehicular MANETs, and compares its performance with the random-waypoint model. However, it lacks a detailed routing scheme in [24].

In delay tolerant networks (DTNs), RCM [17] assumes that nodes are with specific cyclic motion and contact patterns, and uses the Markov decision process to compute a route such that the expected latency of the route is minimized. The experiments are conducted on the same campus bus system as in [13]. However, the urban bus system has more complicated contact relation, and individual buses do not follow a strict cyclic motion. BUBBLE [25] investigates social-based message forwarding based on human mobility traces in a social DTN. BUBBLE utilizes betweenness centrality to identify communities of the social DTN. Messages are routed to the nodes with higher betweenness centralities. In WiFi-enabled DTNs [18], bus stops are utilized as relays for message exchange among buses. It deploys wireless communication units at the bus stops so as to provide bus-stop communications. Messages could be delivered to and stored at the bus stops, and then be forwarded to other buses. In this scheme, two buses do not directly exchange information even if there is an encounter between the two buses.

III. TRACE ANALYSIS OF BEIJING BUS SYSTEM

In order to understand the characteristics of bus systems in typical metropolitan areas, we collect the GPS traces of 2515 buses in Beijing during 1-31 March 2013. Each bus in service submits a GPS report every 20 seconds. Table II illustrates a sample GPS report of a bus which includes information of timestamp, bus ID, bus line number, current location (i.e., “Latitude” and “Longitude”), moving speed, moving direction, etc.

TABLE II: Sample GPS report of a bus

Item	Value
$Timestamp$	2013-03-01 21:49:36
$BusID$	BJ G31279
$Bus\ line\ number$	939
$Longitude$	116.494115
$Latitude$	40.057195
$Speed\ (km/h)$	29
$Moving\ direction$	192
$Next\ stop\ number$	14

We first study geographic distribution of bus traces by aggregating the GPS traces of the 2515 buses. In Figure 2, we plot four instantaneous geographic distributions of bus traces at 7am, 12pm, 3pm and 8pm, respectively. The aggregated bus traces cover an area of 1120 km^2. We have the following observations.

- The instantaneous geographic distribution of the bus traces covers the whole city. The aggregated trace of the 2515 buses indeed forms a backbone of the city, which enables message delivery over the backbone.
- The aggregated trace is stable against the time. Because the bus routes are fixed, the backbones formed by the aggregated traces at different time are more or less the same. It implies that the design of routing schemes is independent of the backbone.

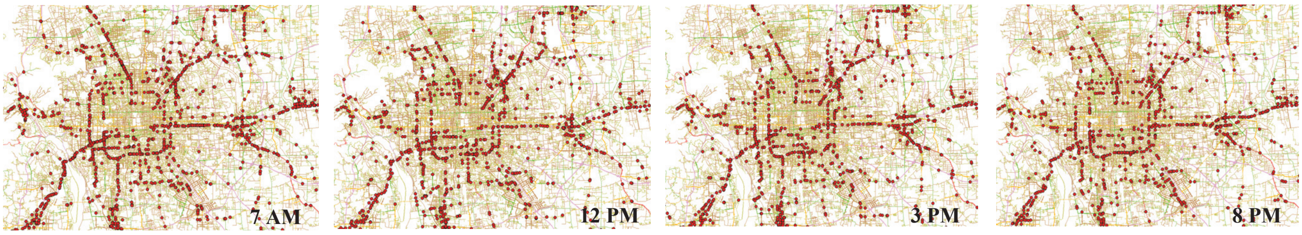


Fig. 2: GPS traces of 2515 buses in Beijing. Each dot represents a GPS report of a bus.

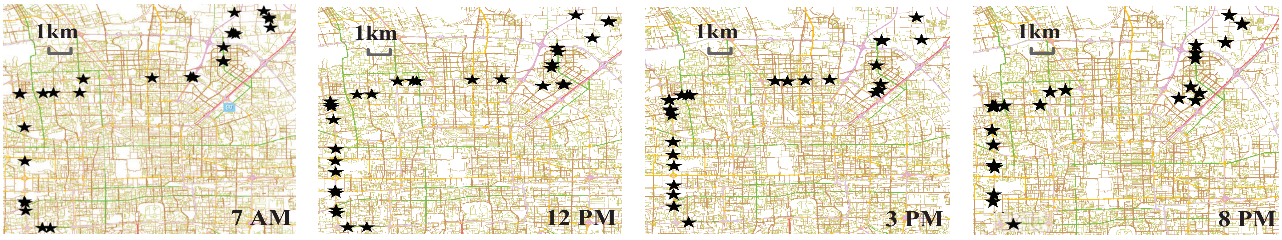


Fig. 3: GPS traces of buses of line No. 944. Each asterisk represents a GPS report of a bus of No. 944.

We plot instantaneous geographic distribution of bus traces of line No.944 in Figure 3, where position of a bus is indicated by an asterisk. We can see that the buses of line No.944 are distributed along their fixed route. Notice that the communication range of vehicles is normally smaller than $1000m$ [26]. These buses cannot form a connected route but several connected components instead. This important feature enables multi-hop delivery of messages in these connected components, which is exploited in the intra-community level routing of CBS (Details are given in Section V.B).

Though the backbone is stable and can be determined by aggregating the traces of all buses, there are many challenges to be addressed so as to realize routing over the backbone. For example, how to explore and utilize the relation between the buses and thus build a structured backbone? How to determine an efficient route on the backbone? In the following, we propose a Community-based Bus System (CBS) to answer these questions. CBS is composed of two components: a community-based backbone (Section IV) and a routing scheme over the backbone (Section V).

IV. COMMUNITY-BASED BACKBONE CONSTRUCTION

The construction of the community-based backbone consists of three steps. Firstly, we build a *contact graph* which shows the closeness relation of bus lines. Secondly, we model the bus system as a social network and apply community detection techniques in the contact graph to compute a *community graph*. Finally, a *backbone graph* can be derived by mapping the community graph to the real map. The three steps are detailed as follows.

A. Constructing Contact Graph

We assume that two buses could exchange a message (called a *contact*) if the distance between their locations is within a given communication range. In the Beijing bus system, trace reports of buses are discretely generated and each bus sends a GPS report every 20 seconds. Taking the time drift into account, we treat two GPS reports that are generated within 20 seconds as the simultaneously-generated reports. We have the following definitions.

***Definition** 1 (**Contact**): There is a contact between two buses if the following two conditions are satisfied. 1) There exist two trace reports which are generated within 20 seconds by the two buses, respectively. 2) Their distance is not greater than the communication range at the time when the two trace reports are generated.*

***Definition** 2 (**Frequency of Contacts**): Frequency of contacts between two bus lines is defined as the number of contacts between any two buses from these two bus lines, respectively, in a unit of time.*

Because all buses of the same line number follow the same fixed route and service schedule, the contact relation is essentially the relation between two bus lines, instead of two individual buses. By analyzing the GPS reports of all buses, we present the definition of the contact graph as follows.

***Definition** 3 (**Contact graph**): A contact graph is defined by a weighted graph $G_{ct} = (V, E)$, where each node $v \in V$ denotes a bus line representing all the buses of the same line number. For $u, v \in V$, there is an edge $e_{uv} \in E$ between bus lines u and v if there exists a contact between two buses from u and v, respectively. Each edge e_{uv} is associated with a weight $w_{uv} = 1/f_{uv}$, where f_{uv} represents the frequency of contacts between u and v.*

Figure 4 shows an instance of the contact graph, which is built by analyzing the one-hour GPS reports of all the 2515 buses in the Beijing bus system. The communication range is set to be $500m$. We can see that the contact graph is connected which implies that the communication between any two bus lines is feasible. In Figure 4, there are 120 bus lines (i.e., nodes) with 516 contacts (i.e., edges). The network diameter of the contact graph is 8, in terms of the number of hops. Notice that the contact graph is a weighted graph. For instance, the

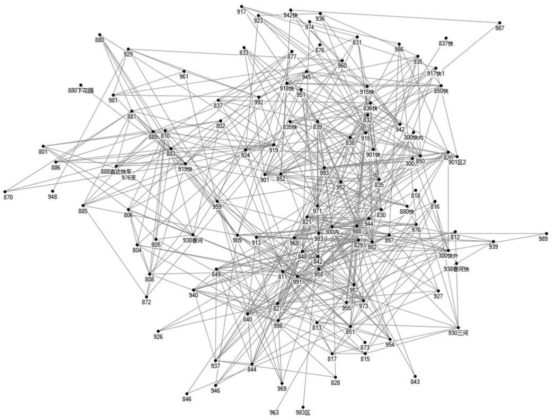


Fig. 4: Contact graph of 120 bus lines (i.e., 120 nodes).

weight of the edge between bus lines No. 955 and No. 988 is 1/393, where 393 is the frequency of contacts between the two bus lines in a unit of time (i.e., one hour in Figure 4). The weights of edges are not specified in Figure 4 due to the large size of the graph.

With the contact graph, a route (e.g., the shortest path) can be easily computed between any pair of nodes in the graph. However, the contact graph is unstructured and the relation among the nodes are not fully exploited. It has been shown that the social relation of the nodes can be exploited for more efficient routing in MANETs [23] and DTNs [25]. In the next section, we study the social structure of the contact graph to build a community graph.

B. Constructing Community Graph

In the contact graph, we notice that some bus lines (i.e., nodes) are strongly connected in terms of the frequency of contacts (i.e., weight of edges) and some bus lines are "active" in connecting more neighboring bus lines. In the discipline of social networks, a *community* is a subset of nodes with stronger connections among them than towards other nodes [27]. Inspired by this concept, we apply community detection techniques in the contact graph to identify potential communities of the bus lines. It is equivalent to partitioning the contact graph into several communities, each of which is a group of strongly connected bus lines. We define the community graph as follows.

***Definition** 4 (**Community Graph**): A community graph is derived from the contact graph G_{ct} and is defined by a weighted graph $G_{cm} = (V, E)$, where each node $v \in V$ denotes a community of bus lines. For $u, v \in V$, there is an edge $e_{uv} \in E$ between communities u and v if there is an edge between two bus lines from u and v, respectively, in G_{ct}. Each edge e_{uv} is associated with a weight which is the minimum weight of edges between any two intermediate bus lines (defined below) in communities u and v, respectively.*

***Definition** 5 (**Intermediate Bus Line**): Based on the contact graph and the community graph, an intermediate bus line is defined as the bus line that belongs to a community, say u, and is connected to a bus line in another community, say v, ($u \neq v$). That is, different communities are connected by the intermediate bus lines.*

Two community detection algorithms are applied in the contact graph to build the community graph. The first is the pioneer and well-known Girvan-Newman algorithm (GN for short), presented by Girvan and Newman [27]. The second is the Clauset-Newman-Moore algorithm (CNM for short) which is the fastest approximation algorithm for large-scale networks, presented by Clauset, Newman and Moore [28]. Both algorithms use *edge betweenness* to measure the influence of an edge. Specifically, edge betweenness of an edge is defined as the number of shortest paths between pairs of nodes that go through this edge in the graph. If a graph contains communities that are loosely connected by a few inter-community edges, then all shortest paths between different communities must go through these inter-community edges. Thus, an edge with a high betweenness is a bridge-like connector between two communities, and the removal of this edge may separate the communities from one another.

For completeness, we describe the basic idea of the GN algorithm [27] in the following. For the CNM algorithm, readers may refer to [28] for details. The GN algorithm first calculates the edge betweenness for all edges in the graph. Each time, the edge with the highest betweenness is removed and the betweenness of the remaining edges are recalculated. The above process is repeated until all edges are removed. The result of the GN algorithm is a reverse tree structure of all nodes in the graph. To determine the number of communities in the graph, Newman defined a quality function called *modularity* [29] as follows.

$$Q = \frac{1}{2m} \sum_{vw} [A_{vw} - \frac{k_v k_w}{2m}] \delta(c_v, c_w), \tag{1}$$

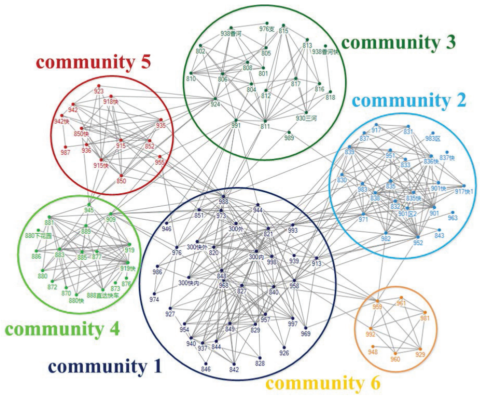
where m is the number of edges in the graph and k_v/k_w is the degree of vertex v/w. Vertices v and w belong to communities c_v and c_w, respectively. δ-function $\delta(c_v, c_w)$ is 1 if $c_v = c_w$ and 0 otherwise. A_{vw} is 1 if vertices v and w are connected and 0 otherwise.

Modularity Q measures the fraction of the within-community edges (i.e., the edges that connect vertices which belong to the same community) minus the expected value of the same quantity in a randomized network. Q is equal to 0 if the number of within-community edges is no different from what we expect for the randomized network. Value of Q approaching 1 (i.e., the maximum value) indicates a strong community structure of the network. In practice, a value above 0.3 is a good indicator of significant community structure of a network, and the typical value of Q is in the range from 0.3 to 0.7 [29].

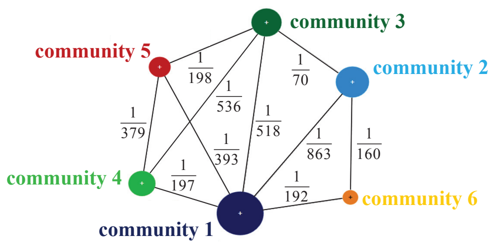
We apply both the GN algorithm and the CNM algorithm to the contact graph in Figure 4 to compute the community graph with the maximum modularity value. The modularity value is maximized when the number of communities equals 6 in both algorithms. The optimal modularity value of the GN algorithm is $Q = 0.576$ while that of the CNM algorithm is $Q = 0.53$. The comparison of the results is given in Table III, where each number in the second and the third columns denotes the number of bus lines contained in the corresponding community that is computed by the algorithms. The fourth column "Common" indicates the number of common bus lines in the same community identified by the two algorithms. There are more than 93% overlap between the communities computed by the two algorithms, which implies that the both algorithms output a similar community structure. Since the modularity value of the GN algorithm is larger, we adopt the GN algorithm in constructing the community graph which is shown in Figure 5, where the 120 bus lines are partitioned into 6 communities in Figure 5(a) and an abbreviated graph is shown in Figure 5(b). The community graph is a weighted graph. For instance, the weight of the edge between community 3 and community 5 is 1/198 which is the minimum weight of edges between any two intermediate bus lines in these two communities, respectively.

TABLE III: Number of bus lines in communities

	GN	CNM	Common
$Community$ 1	37	32	32
$Community$ 2	24	25	24
$Community$ 3	21	19	19
$Community$ 4	18	18	17
$Community$ 5	13	16	13
$Community$ 6	7	10	7



(a)



(b)

Fig. 5: Community graph. Six communities are identified.

C. Constructing Backbone Graph

Once the community graph is available, we can construct the backbone graph by mapping the fixed routes of the bus lines to the corresponding locations in the real map. In this way, the bus lines, the communities and geographic locations/areas are logically connected so as to support geographic routing. In other words, given a geographic location/area, the backbone graph can determine a community containing a bus line whose service (e.g., route) covers this location/area. The backbone graph is defined as follows.

***Definition** 6 (**Backbone Graph**): The backbone graph is derived from the community graph by mapping the fixed routes of the bus lines to the corresponding roads in the map. The backbone is partitioned into communities and different communities may overlap due to the overlapping routes of the bus lines in these communities.*

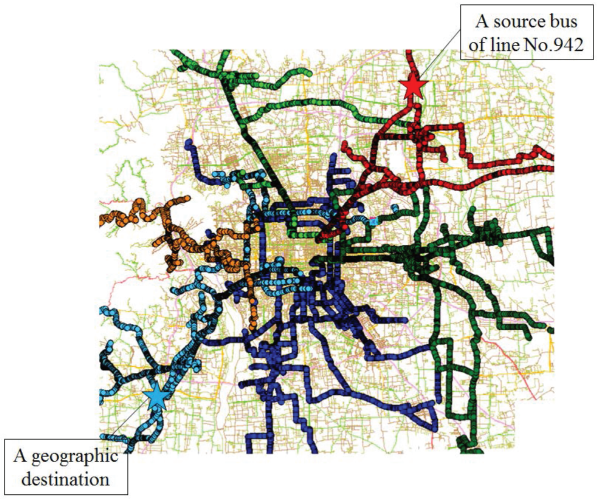


Fig. 6: Backbone graph. Six communities are in different colors.

Figure 6 shows the backbone graph derived from the community graph in Figure 5. The community-based backbone is partitioned into 6 regions (i.e., communities) in different colors. Notice that the construction of the backbone graph is a one-off operation and the graph can be preloaded at all buses once it is computed. The backbone graph needs to be updated only when there is a change of bus service (e.g., change of bus routes) which does not frequently occur in practice.

V. ROUTING OVER THE BACKBONE

In this section, we propose a two-level routing scheme of CBS which operates over the community-based backbone. The routing scheme can support message delivery to both mobile vehicles (i.e., vehicle $\rightarrow$ vehicle case) and specific locations/areas (i.e., vehicle $\rightarrow$ location case). Notice that a routing scheme that supports the vehicle $\rightarrow$ location case is also applicable to the vehicle $\rightarrow$ vehicle case as long as the trace of the destination vehicle is known. The basic idea is as follows. The routing message could be delivered to the bus whose route overlaps the trace of the destination vehicle. When the bus travels in the overlapping route, it can broadcast the message to any vehicles that it encounters. If the bus misses the destination vehicle, it could send the message to other buses which serve in the same overlapping route. In this way, the destination vehicle can eventually receive the message with a high probability. We focus on the routing from a source vehicle to a geographic destination (i.e., vehicle $\rightarrow$ location case) in this work.

The source vehicle could be any general vehicle, e.g., a taxi and a private car. Because the backbone covers the whole city, we assume that a routing message can be easily delivered from the source vehicle to one of its nearby buses traveling in the backbone. Therefore, the routing actually starts with a bus who first gets the routing message. For simplicity, we assume that the source vehicle is a bus. We assume that the community graph and the backbone graph are preloaded at all buses. Each bus periodically broadcasts HELLO messages so that it can be aware of neighboring buses within the communication range. The routing scheme of CBS operates in two levels, namely, the inter-community level (i.e., macro view) and the intra-community level (i.e., micro view). The inter-community routing is to determine a route from a source community to a destination community in the community graph, while the intra-community routing is to compute an efficient route within each community.

A. Inter-community Routing

Given a source vehicle (i.e., bus) and a geographic destination, the routing scheme of CBS performs in the following three steps.

1) Identifying source and destination communities: The source and the destination are mapped to the corresponding communities in the backbone graph. For the source bus, it is easy to find the community which contains the corresponding bus line of the source. For the destination, it determines the community that includes the bus line whose route covers the destination. Taking Figure 6 as an example, the source is a bus of line No. 942 and the destination is located in the down-left corner of the map. Combining the backbone graph in Figure 6 and the community graph in Figure 5, we can determine that bus line No. 942 belongs to community 5 and the route of bus line No. 837 in community 2 goes through the destination. We call community 5 *the source community* and community 2 *the destination community*.

2) Computing route between source and destination communities: In the community graph (i.e., the weighted graph shown in Figure 5, the shortest path is computed from the source community to the destination community: community 5 $\rightarrow$ community 1 $\rightarrow$ community 2, which is called *an inter-community route*. If the source and the destination belong to the same community, the intra-community routing (presented in next subsection) is invoked immediately.

3) Identifying intermediate bus lines of communities: According to Definition 5, different communities are connected by the intermediate bus lines. For each community in the route, we determine the intermediate bus line that is connected to the downstream community in the route. If there are more than one intermediate bus lines, it selects the one with the smallest weight (the most stable connection) in connecting the downstream community.

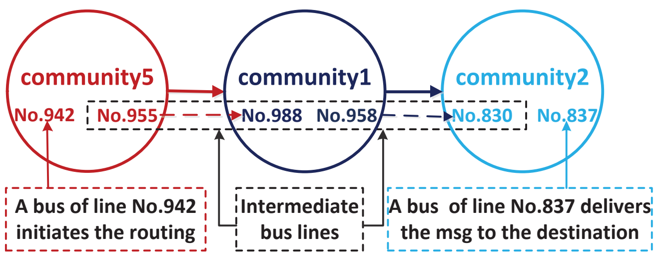


Fig. 7: Inter-community routing.

Figure 7 shows the inter-community route determined in the inter-community routing. We can see that bus line No. 955 belongs to community 5 and it is the intermediate bus line that connects bus line No. 988 in community 1.

B. Intra-community Routing

Once an inter-community route is computed, the intra-community routing is applied in each of the communities in the route. The intra-community routing is to route the message from a bus of the source bus line carrying the message to the bus of the intermediate bus line within the same community. In each of the communities, the intermediate bus line serves as the destination bus line in the intra-community routing in that community. It is believed that a larger frequency of contacts between bus lines implies a more reliable communication relation and a higher delivery ratio. Since the bus lines in the same community are strongly connected with larger frequencies of contacts, it is better to limit the message delivery within the community. The intra-community routing is described as follows.

1) Computing intra-community route: Given a community, the source bus line and the destination bus line (i.e., the intermediate bus line in this community), we delete all the bus lines that do not belong to this community and the corresponding edges in the contact graph. The resulting graph is a subgraph of the contact graph induced by the bus lines in this community. Then, the shortest path is computed from the source bus line to the destination bus line in this subgraph. Figure 8 shows the intra-community routing in the three communities which are calculated in Figure 7. Taking community 5 as an example, the optimal intra-community route is: No. 942 (the source bus line in community 5) $\rightarrow$ No. 918K $\rightarrow$ No. 915 $\rightarrow$ No. 955 (the destination bus line in community 5).

2) Exploiting multi-hop transmission: According to the observation made in Figure 3, buses of the same bus line usually form several connected components in the route. It implies that multi-hop transmission is feasible in these connected components. Thus, a copy of the message will be delivered to the buses of the same bus line whenever the multi-hop transmission is possible. This strategy can significantly save the carrying time of buses and can increase the delivery ratio. Since the number of buses under a single bus line is limited (e.g., a typical number is 20 in the Beijing bus system), the overhead of duplicated messages is acceptable.

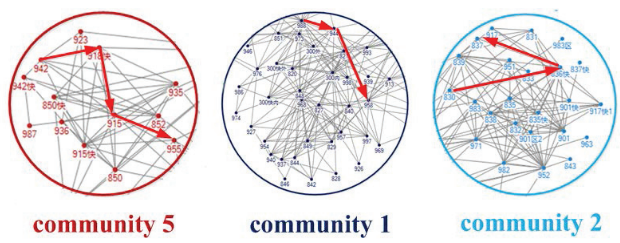


Fig. 8: Intra-community routing.

After the intra-community routing is applied in the three communities in Figure 8, the final route from the source bus line (No. 942) to the destination bus line (No. 837) is determined as follows: No. 942 (5) $\rightarrow$ No. 918K (5) $\rightarrow$ No. 915 (5) $\rightarrow$ No. 955 (5) $\rightarrow$ No. 988 (1) $\rightarrow$ No. 944 (1) $\rightarrow$ No. 958 (1) $\rightarrow$ No. 830 (2) $\rightarrow$ No. 836K (2) $\rightarrow$ No. 837 (2), where each number in the parenthesis denotes the community that the bus line belongs to. There are 9 hops in the route.

VI. PERFORMANCE EVALUATION

A. Experimental Setup

We conduct extensive trace-driven simulations to study the performance of CBS. In the simulations, CBS is compared with two routing schemes in bus-based VANETs [14] [15] and one routing scheme in general VANETs [16]. The routing schemes selected for comparison are briefly described as follows.

- BLER [14]. It first builds a graph in which each node denotes a bus line and each edge of two nodes indicates at least one contact between these two nodes (i.e., bus lines). The weight of an edge is the contact length which is defined as the length of overlapping routes of the corresponding bus lines. The routing path to the destination is computed in the graph such that the sum of contact length of the route is maximized.
- R2R [15]. This scheme uses a similar graph as BLER except that the weight of an edge is the frequency of contacts between two bus lines.
- ZOOM [16]. In this scheme, each vehicle estimates its delay to the destination vehicle based on historical contact information. All vehicles are grouped into communities by the Louvain algorithm [30] and ego-betweenness is adopted to measure the centrity of the vehicles. A vehicle, say u, delivers a message to a relay vehicle, say v, if one of the following three rules is satisfied. 1) v is the destination vehicle of the message. 2) v has a shorter estimated delay to the destination vehicle. 3) v has a larger ego-betweenness than u if both u and v have no information about the destination vehicle. The vehicles in ZOOM include taxis and buses and the estimated delay of next contact between two vehicles follows a specific probability distribution. In contrast, the buses considered in this work normally have fix routes/schedules and the probability model developed in [16] is not applicable to the bus system. Therefore, we adopt only rules 1) and 3) of ZOOM and name this modified scheme ZOOM-like in our simulations.

We consider the following metrics in evaluating the performance of the routing schemes.

1) Delivery ratio. It is defined as the ratio of the number of successfully-delivered messages to the total number of messages during the operation of the bus system.

2) Delivery latency. It is defined as the time it takes to deliver the message from the source to the destination. It is applicable to only successfully-delivered messages.

3) Number of hops. It is defined as the number of bus lines it takes to deliver the message from the source to the destination.

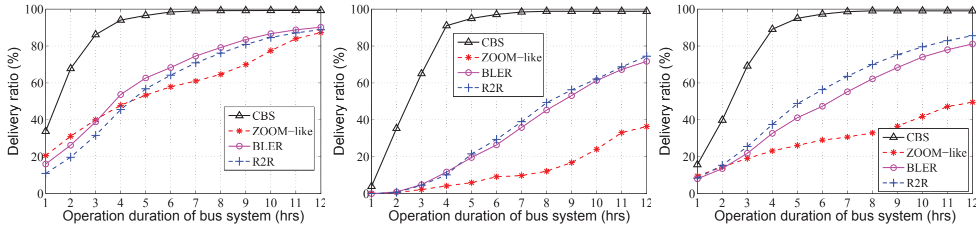
4) Message carrying time. It is defined as the average duration that each relay vehicle carries the message.

In the simulations, we set the communication range to be $500m$ and adopt the real GPS traces of 2515 buses of 120 bus lines in Beijing to build the graphs needed by the four routing schemes (i.e., CBS, BLER, R2R, and ZOOM-like). Since CBS, BLER and R2R depend on the relations of bus lines and the contact relations of the bus lines are relatively stable, we use one-hour traces to generate the graphs in CBS, BLER and R2R. For CBS, we build the contact graph in Fig. 5 according to Definitions 1-3. Then, the GN algorithm [27] is applied to generate the community graph in Fig. 6. The backbone graph is finally computed in Fig. 7. BLER and R2R build the contact graphs in a similar way to CBS. The weight of each edge is set to be the contact length and the frequency of the contact, respectively, in BLER and R2R. ZOOM stems from mining the relations of individual vehicles (e.g., taxies). Considering that our data set includes only the traces of buses and daily traces of individual buses normally are similar, we use one-day traces of the buses to generate the graph of ZOOM-like in which 49 communities of the 2515 buses are identified.

We study three cases of routing requests: 1) short distance case, 2) long distance case, and 3) hybrid case. The source and the destination are within a same community in the short distance case while they are located in different communities (i.e., the routing has to cross multiple communities) in the long distance case. The hybrid case is the mixture of both the short distance case and the long distance case. 6000 routing requests are generated in the first 6000 seconds of the experiment, i.e., a new routing request is generated in every second of the 6000 seconds. For each routing request, the source vehicle (i.e., the source bus) is randomly selected from the 2515 buses. In the short distance case, the destination location is randomly selected from the joint routes of all the bus lines which belong to the same community of the source bus. A bus whose route covers this destination location acts as the destination bus. In the long distance case, the destination bus is selected in a similar way but the destination is located outside the community of the source bus. The destination location could appear in any place of the backbone in the hybrid case. The experiment lasts 12 hours, i.e., the operation of the bus system lasts 12 hours. A message that can be delivered to the destination bus within 12 hours is counted as a successfully-delivered message in calculation of the delivery ratio.

B. Delivery Ratio

Figure 9 shows the delivery ratio versus the operation duration of the bus system in the three cases. From Figure 9(a) (i.e., the short distance case), we observe that CBS gives the highest delivery ratio among all the schemes. For instance, CBS successfully delivers 94% messages within 4 hours, while the corresponding delivery ratios of BLER, R2R, and ZOOM-like are 54%, 46% and 48%, respectively. In the long distance case shown in Figure 9(b), the curves of all the schemes drop because it takes a longer time to deliver a message to the destination bus when the routing distance becomes larger. Nevertheless, CBS significantly outperforms the other three schemes in the long distance case. A similar conclusion can be drawn in the hybrid case in Figure 9(c). Figure 10 shows the relation between the delivery ratio and the communication range. We can see that the delivery ratio of CBS retains stable at a high level, regardless of the change of the communication range. For the other three schemes, their delivery ratios improve as the communication range increases. In particular, a significant increase is observed when the communication range increases from $100m$ to $200m$. The



(a) Short distance case (b) Long distance case (c) Hybrid case

Fig. 9: Delivery ratio vs. operation duration of bus system.

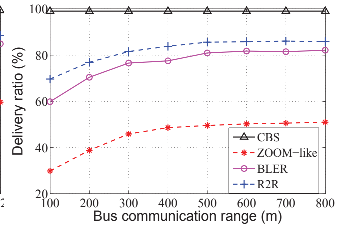
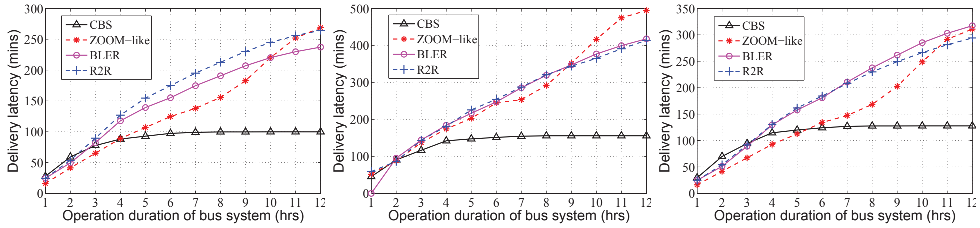


Fig. 10: Delivery ratio vs. communication range.



(a) Short distance case (b) Long distance case (c) Hybrid case

Fig. 11: Delivery latency vs. operation duration of bus system.

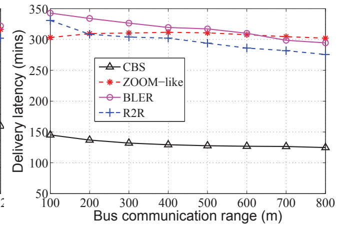


Fig. 12: Delivery latency vs. communication range.

delivery ratios of BLER, R2R, and ZOOM-like stabilize when the communication range is greater than $500m$.

C. Delivery Latency

Figure 11 shows the delivery latency versus the operation duration of the bus system in the three cases. We can see that CBS gives the shortest latency when the bus system lasts over five hours. Its delivery latency increases slightly at the first four hours and remains stable thereafter. Specifically, the delivery latency of CBS is under 100 minutes, 155 minutes and 127 minutes, respectively, in the short, long, and hybrid distance cases. In contrast, the delivery latencies of the other three schemes increase rapidly. The reasons are given in the following. Since BLER is very similar to R2R, we take R2R as an example for explanation. R2R simply computes the routing path in the contact graph such that the sum of weights (i.e., frequency of contacts) in the path is maximized. This simple strategy may miss those crucial links which bridge different groups (i.e., communities) of bus lines. As a result, R2R may output the path containing an unreliable link with a small frequency of contacts, which consequently increases its delivery latency. ZOOM utilizes the relation of individual vehicles for routing. So the routing efficiency of ZOOM is highly dependent on the number of contacts of the vehicles. However, we find that a bus can contact only 5% of all buses in the Beijing bus system. The delivery latency of ZOOM-like is high with this limited number of contacts.

Figure 12 shows the delivery latency versus the communication range in the three cases. It can be seen that all the four schemes experience a decrease as the communication range increases. Obviously, a large communication range facilitates the contact of buses and increases the probability of multi-hop forwarding. When the communication range increases from $100m$ to $800m$, the delivery latencies of CBS, BLER, and R2R decrease by 14.5%, 14%, and 16.6%, respectively. In contrast, the decrease in the delivery latency of ZOOM-like is not significant. Once again, CBS gives the shortest delivery latency among all the schemes, which demonstrates the superiority of routing over the backbone of the bus system.

D. Number of Hops and Message Carrying Time

Figure 13 shows the average number of hops of the successfully-delivered messages. The hops are measured between the bus lines. In CBS, it needs about 6 hops on average to deliver a message to the destination, which is larger than the other three schemes. Notice that the delivery latency in VANETs is mainly due to the latency in carrying the messages instead of the latency in forwarding the messages. CBS gives the highest delivery ratio as well as the shortest delivery latency among all the schemes, regardless of its larger number of bus lines in delivering the messages.

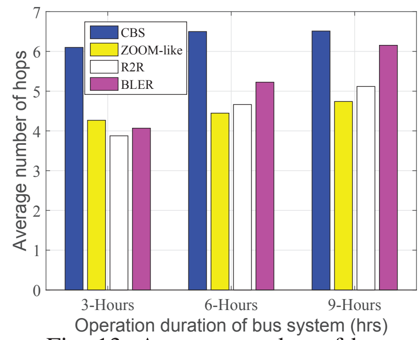


Fig. 13: Average number of hops

Figure 14 shows the average carrying time of each message in each bus line before it is delivered to the destination. The average carrying time of CBS is below 20 minutes which is the lowest among all the schemes. This is because CBS exploits multi-hop transmissions among the buses for fast message delivery. The average carrying time of ZOOM-like increases significantly after nine hours elapse in the experiment. In BLER and R2R, without the multi-hop transmissions, buses have to carry messages for a long time until they find suitable relay buses.

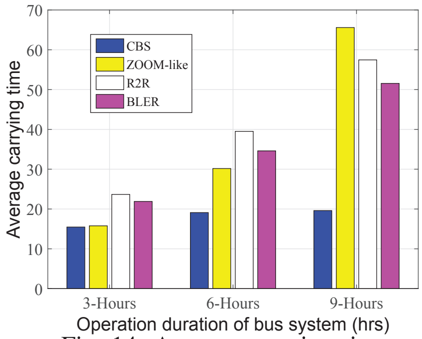


Fig. 14: Average carrying time

VII. Conclusion

This work is based on real GPS traces of 2515 buses in Beijing, China. By analyzing these GPS traces, we have found that the bus system can serve as a routing backbone of VANETs. We have proposed a Community-based Bus System (CBS) as routing backbone to support efficient message delivery in VANETs. CBS consists of a community-based backbone and a routing scheme over the backbone. We have built a community-based backbone by applying community detection techniques in the Beijing bus system. We have presented a two-level routing scheme that operates over the backbone. The proposed routing scheme is able to deliver messages to both mobile vehicles and specific locations/areas. Extensive experiments have been conducted on the real trace data of the Beijing bus system. The experimental results have shown that CBS can significantly lower the delivery latency and improve the delivery ratio, compared to the existing solutions.

Acknowledgment

This work is supported in part by General Research Fund of Hong Kong [No. GRF211513], Research Grants of Hong Kong Baptist University [No. FRG2/12-13/055 and FRG2/13-14/091], and National Natural Science Foundation of China [No. 61472408].

References

[1] D. Zhang and C. K. Yeo, "Enabling efficient wifi-based vehicular content distribution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 3, pp. 479–492, 2013.

[2] R. Lu, X. Lin, H. Zhu, and X. S. Shen, "SPARK: a new VANET-based smart parking scheme for large parking lots," *Proc. IEEE INFOCOM*, pp. 1413–1421, 2009.

[3] J. Bao, Y. Zheng, and M. F. Mokbel, "Location-based and preference-aware recommendation using sparse geo-social networking data," *Proc. ACM SIGSPATIAL*, 2012.

[4] J. Qin, H. Zhu, Y. Zhu, L. Lu, G. Xue, and M. Li, "POST: Exploiting dynamic sociality for mobile advertising in vehicular networks," *Proc IEEE INFOCOM*, pp. 1761–1769, 2014.

[5] M.-C. Chuang and M. C. Chen, "Deep: Density-aware emergency message extension protocol for vanets," *IEEE Transactions on Wireless Communications*, vol. 12, no. 10, pp. 4983–4993, 2013.

[6] F. J. Ros, P. M. Ruiz, and I. Stojmenovic, "Acknowledgment-based broadcast protocol for reliable and efficient data dissemination in vehicular ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 33–46, Jan. 2012.

[7] M. Slavik and I. Mahgoub, "Spatial distribution and channel quality adaptive protocol for multihop wireless broadcast routing in VANET," *IEEE Transactions on Mobile Computing*, vol. 12, no. 4, pp. 722–734, Apr. 2013.

[8] C. Lochert, H. Hartenstein, J. Tian, H. FiiRler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," *Proc. IEEE Intelligent Vehicles Symposium*, pp. 156–161, 2003.

[9] C. Lochert, M. Mauve, H. Fler, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 12, no. 4, pp. 69–72, 2005.

[10] D. Yong and X. Li, "Sadv: Static-node-assisted adaptive data dissemination in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 5, pp. 2445–2455, 2010.

[11] 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.

[12] J. Jeong, S. Guo, Y. Gu, T. He, and D. Du, "TBD: Trajectory-based data forwarding for light-traffic vehicular networks," *Proc. IEEE ICDCS*, pp. 231–238, 2009.

[13] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: routing for vehicle-based disruption-tolerant networks," *Proc IEEE INFOCOM*, pp. 1–11, 2006.

[14] M. Sede, X. Li, D. Li, M.-Y. Wu, M. Li, and W. Shu, "Routing in large-scale buses ad hoc networks," *IEEE WCNC*, pp. 2711–2716, 2008.

[15] L. Li, Y. Liu, Z. Li, and L. Sun, "R2R: data forwarding in large-scale bus-based delay tolerant sensor networks," *Proc. IET WSN*, pp. 27–31, 2010.

[16] H. Zhu, M. Dong, S. Chang, Y. Zhu, M. Li, and S. Shen, "ZOOM: Scaling the mobility for fast opportunistic forwarding in vehicular networks," *Proc IEEE INFOCOM*, pp. 2832–2840, 2013.

[17] C. Liu and J. Wu, "Practical routing in a cyclic mobispace," *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, pp. 369–382, 2011.

[18] U. G. Acer, P. Giaccone, D. Hay, G. Neglia, and S. Tarapiah, "Timely data delivery in a realistic bus network," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 1251–1265, 2012.

[19] V. D.C., M. Patra, and C. S. R. Murthy, "Joint placement and sleep scheduling of grid-connected solar powered road side units in vehicular networks," *Proc. IEEE WiOpt*, pp. 534–540, 2014.

[20] L. Zhang, B. Yu, and J. Pan, "GeoMob: a mobility-aware geocast scheme in metropolitans via taxicabs and buses," *Proc. IEEE INFOCOM*, 2014.

[21] Y.-B. Ko and N. H. Vaidya, "Geotora: a protocol for geocasting in mobile ad hoc networks," *Proc. IEEE ICNP*, pp. 240–250, 2000.

[22] W.-H. Liao, Y.-C. Tseng, K.-L. Lo1, and J.-P. Sh, "Geogrid: A geocasting protocol for mobile ad hoc networks based on grid," *Journal of Internet Technology*, vol. 1, pp. 23–32, 2000.

[23] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," *Proc of ACM MOBIHOC*, pp. 32–40, 2007.

[24] K. J. Wong, B. sung Lee, B. chong Seet, G. Liu, and L. Zhu, "BUSNet: Model and usage of regular traffic patterns in mobile ad hoc networks for inter-vehicular communications," *Proc. ICT Conf*, p. 273C289, 2003.

[25] P. Hui, jon Crowcroft, and E. Yoneki, "BUBBLE Rap: social-based forwarding in delay-tolerant networks," *IEEE/ACM Transactions on MOBILE COMPUTING*, vol. 10, no. 11, pp. 1576–1589, 2011.

[26] J. Guo and N. Balon, "Vehicular ad hoc networks and dedicated short-range communication," *Technical Report, University of Michigan - Dearborn*, June 26, 2006.

[27] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences(PNAS)*, p. 99:7821C7826, 2002.

[28] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E 70*, 2004.

[29] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E 69*, p. 026113, 2004.

[30] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech.*, 2008.