Privately Information Sharing with Delusional Paths for Data Forwarding in Vehicular Networks

Zhong Li, Cheng Wang, Lu Shao, Changjun Jiang

Abstract—We discuss how to efficiently forward data in vehicular networks. Existing solutions do not make full use of trajectory planning of nearby vehicles, or social attributes. The development of onboard navigation system provides drivers some traveling route information. The main novelty of our approach is to envision sharing partial traveling information to the encountered vehicles for better service. Our data forwarding algorithm utilizes this lightweight information under the delusional paths privacy preservation together with the social community structure in vehicular networks. We assume that data transmission is carried by vehicles and road side units (RSUs), while cellular network manages and coordinates relevant global information. The approximate destination set is the set of RSUs that are often passed by the destination vehicle. RSU importance is raised by summing encounter ratios of RSUs in the same connected component. We define a concept of space-time approachability which is derived from shared partial traveling route and encounter information. It describes the capability of a vehicle to advance messages toward destination. Then, we design a novel data forwarding algorithm, called approachability based algorithm, which combines the space-time approachability with the social community attribute in vehicular networks. We evaluate our approachability based algorithm on data sets from San Francisco Cabspotting and Shanghai Taxi Movement. Results show that the partially shared traveling information plays a positive role in data forwarding in vehicular networks. Approachability based data forwarding algorithm achieves a better performance than existing social based algorithms in vehicular networks.

Index Terms—vehicular networks, traveling information sharing, delusional paths, data forwarding

1 INTRODUCTION

As an important part of intelligent transportation system, vehicular networks have recently received increasing attention. Various intelligent vehicle applications have been developed to satisfy users’ different requirements, such as vehicle maintenance (a reminder for recall, fault notification in the vehicle etc.), community services (hiking, crowd sensing etc.), safety services (emergency alert, collision detection etc.). Therefore, an efficient data forwarding is important for vehicular networks to support these applications. There are generally two important factors in designing data forwarding schemes for vehicular networks, i.e., the communication architecture and the data forwarding methodology.

In the aspect of the communication architecture, a straightforward solution is to utilize the cellular networks to forward data in vehicular networks. However, there are some reasons to make such a solution impractical. First, the existing cellular networks already undertake various traffics, such as mobile telephone communication, then convincingly cannot provide enough capability for dealing with all data forwarding in a large-scale vehicular network. In addition, it is more impractical to build a dedicated cellular network for vehicular networks due to the high construction costs. Second, the direct point-to-point communication is often desired to improve the performance of some applications, such as real-time, privacy-aware and location-based social applications [1], [2].

To address these issues, the ad hoc mode is introduced, since it can not only offload the heavy data traffic from cellular networks but also seize the opportunity for vehicles to communicate directly with each other. Many representative studies contribute to the opportunistic data forwarding in the ad hoc way [3]–[8]. In these studies, some of them focusing on the mobile networks can also be applied to vehicular networks. It is worth noticing that some of them do not indicate clearly the architectures of their communication networks. The centralized coordination is always implicitly necessary for the implementation for their schemes. For example, in the milestone work BUBBLE RAP [5] contributed by Hui et al., the message holder needs to know whether it reaches a node within the same community as the destination node. Such information can only be obtained from the centralized coordinator. In other words, all those so-called ad hoc strategies cannot work without the centralized coordinator support. Naturally, the cellular network is brought about to serve as such centralized coordinator.

Then, a “paradox” rises as to why the cellular network does not undertake the data transportation as a matter of course since it has to come on the stage. While, we disagree that this is a paradox, and believe that the practicality of those strategies will not be destroyed due to the present of cellular networks. The crucial explanation is that the cellular network as the centralized coordinator only needs to transmit some small messages about the information of destination or some global control signals, rather than undertake a large amount of data forwarding, such as video transmission. This is basically consistent to the initial purpose of offloading the heavy workload in cellular networks. Therefore, in our work, we adopt such hybrid communication architecture. In this architecture, there are two layers as shown in Fig. [1] the top layer is the cellular network with base stations; the underlying layer is the ad

- Zhong Li, Cheng Wang, Lu Shao and Changjun Jiang are with the Department of Computer Science, Tongji University, Shanghai 201804, China.
- Correspondence Author: Changjun Jiang, E-mail: cjjiang@tongji.edu.cn.
hoc networks with vehicles and stationary road side units (RSUs). Specially, RSUs do not cover the whole city area and some of them are interconnected via wired or wireless links, i.e., there are several RSU connected components. RSUs allow passing vehicles to collect data from or deposit data into them according to the schemes designed in our work.

Here, we give three important additional explanations about the hybrid communication architecture used in this work. Firstly, with regard to the underlying network in the hybrid communication architecture, some studies about the data forwarding algorithms, like literature [9], [10], are also based on the underlying network with RSU/AP support. However, our network architecture is different from them: 1) we consider the role of base stations; 2) the RSUs/APs are not only used as road side solitary relays without connectivity; 3) the RSUs/APs have “brains” to decide whether to do data forwarding, i.e., they are not passive.

Secondly, in another cases, some studies usually make a simple assumption, in which a large amount of RSUs/APs are all connected with a backbone network in city area. This assumption indeed simplifies the network analysis [11]–[14]. In this kind of network environment, the data flooding along the backbone results in heavy workloads on all RSUs/APs. In fact, we can partition some RSU connected components according to some rules, such as the geographical area. In our work, for simplicity, we partition some RSU connected components based on the following steps: 1) system dispatches the sequence numbers (natural number) to each RSU; 2) according to ascending sequence number (from small to large sequence number), random numbers (2-5) of RSUs are grouped into different connected components in which RSUs are connected one by one (a chain). So, our underlying network with RSU support could not only guarantee a certain amount of copies in the network, but also avoid introducing a large scale data flooding.

Thirdly, the deployment of RSUs indeed impacts the experiment results. However, when we do comparison experiments in the work, we use the fair RSU copy strategy for all algorithms.

In the aspect of the forwarding methodology, it is crucial how to utilize the various information obtained from the vehicular network to design an efficient data forwarding scheme. Previous studies usually make full use of the map or road information to predict the future forwarding direction to forward data, such as some geography based methods [15]–[17] and trajectory based methods [9], [10]. However, the computation of the map or road information costs highly since the knowledge of GPS, road segment, vehicle speed etc. is often required. Then, researchers want to throw away these complex information and only use the vehicle social contact behaviors to predict the encounter probability to improve data forwarding, like some contact based methods [18], [19] and social based methods [5], [6], [8], [20]. However, we realize that this kind of research does not sufficiently consider the difference between the vehicular networks and common mobile networks in data forwarding. 1) Vehicles are moving on the physically constrained (i.e., road-restricted) areas. The inter contact time is limited by the traffic condition. 2) The shortest communication paths do not always match the physical shortest paths. So, we can see that the road information is indeed important for vehicles to forward data. So, on the basis of these efficient contact/social based methods, we hope the road information can return to the data forwarding design in vehicular networks. Then, we will face three challenges when we want to realize this hope.

Firstly, does there exist some lightweight road information that can reflect the geography/trajectory case? Second, how to obtain this kind of information in a security way? Third, how to formalize and use the lightweight road information together with the contact/social information to design an efficient data forwarding scheme? To address the aforementioned problems, in this work, we provide the following solutions.

Firstly, nowadays, many vehicles are equipped with navigators. So, given a destination, the traveling route can be calculated and adaptively adjusted according to the real-time traffic conditions. So, the vehicle can obtain some knowledge about the forthcoming traveling route easily. In vehicular networks, people usually like to join some activities to obtain good services. So, they could share the traveling route information more or less in this process. We can see, this traveling route information has two advantages. 1) It is lightweight (unlike the large map or road information) and can be easily exchanged between two vehicles without the help of the base stations. 2) It implies some road information, i.e., the forthcoming positions and the corresponding arrival time. So, the traveling information sharing satisfies our philosophy of exploiting the road information in a lightweight way. Therefore, we could take the traveling route information as an important factor to design the data forwarding algorithm for vehicular networks.

Secondly, when a vehicle shares the traveling information, it will worry about whether this behavior is security. As an information provider, some sharing information give to a strange information receiver is dangerous, since the traveling trace may be derived by the receiver. Thus, in our work, we provide a privacy preservation algorithm, called delusive paths, to ensure the trace safety when a vehicle shares its partial traveling information. The idea of delusive paths is when a vehicle wants to share new traveling information (including the position and the arrival time), delusive paths algorithm will check the number of paths between the last shared traveling information and the new traveling information. If the number of paths is more than or equals to 2, the algorithm will allow the new traveling information to be released. Otherwise, the new information is forbidden releasing. So, any two tuples of shared traveling information have 2\(^n\) paths, \(n \geq 1\), which
makes the information receiver can not guess the exact trace of the information provider.

Thirdly, in this work, we study the unicast session among vehicles under the hybrid communication architecture. We give a series of formalized definitions and measurements to use the lightweight road information together with the contact/social information to design an efficient data forwarding scheme.

1) We define approximate destination set to target the destination. According to the encounter ratio between RSUs and the destination, we use the median to filter RSUs and form the approximate destination set. The elements in the set are the RSUs that are often passed by the destination vehicle. If the message is first delivered to the RSUs contained in the approximate destination set, the destination vehicle will have more chances to directly receive the message from these RSUs. Besides, we also define a concept of RSU importance. For RSUs in the approximate destination set, if there exist some RSUs which are in the same RSU connected component, the RSU importance of this kind of RSU is defined as the sum of the encounter ratio (between the RSU and the destination) values of all RSUs in common connected components. Otherwise, the RSU importance is only defined as the encounter ratio (between the RSU and the destination) of single RSU.

2) We term the lightweight traveling information (the position and the arrival time of a vehicle) as a kind of relative deterministic information due to the stability of a map; meanwhile, we term the social attribute of the vehicles (e.g., community) as a kind of relative non-deterministic information due to the dynamics of social behaviors. With respect to the traveling information, we give the definition of space-time approachability which is adopted to measure the probability of a vehicle approaching to the approximate destination set. Space-Time approachability is calculated as a sum of the ratios for all shared traveling information at a vehicle’s one encounter. The numerator of the ratio is the sum of RSU importance for all RSUs within the communication range of one shared position. The denominator of the ratio is the difference value between the corresponding forthcoming arrival time and the current time. The lightweight traveling information of a vehicle gives us a kind of deterministic information to measure the approaching probability to the destination. Therefore, the larger space-time approachability a vehicle has, the faster the message will be carried to the approximate destination set. With respect to the social attribute, we use the community structure to characterize it in the vehicular networks.

3) Then, based on space-time approachability, we combine the social attribute to design an approachability based data forwarding algorithm in vehicular networks. In the algorithm, for different phases (vehicle→vehicle, vehicle→RSU, RSU→vehicle), the approachability based measurement is utilized in different ways. Besides, the RSUs in this algorithm do not passively send messages to all the passing vehicles. According to the approachability based measurement, they have the ability to decide how to forward data. We extensively evaluate the approachability based data forwarding algorithm on two data sets: San Francisco Cabspotting and Shanghai Taxi Movement. The results show that the approachability based data forwarding algorithm significantly outperforms several existing social based data forwarding algorithms applied to vehicular networks. In particular, we test the algorithm on the rush and non-rush hours of a day and see the power of traveling information sharing in data forwarding independently.

The rest of the paper is organized as follows. We review the related work in Section 2. We present the network model and some assumptions in Section 3. We give the method of the traveling information sharing under privacy preservation with delusive paths. In Section 4, we introduce the space-crossing community to characterize the social attribute of vehicular networks. In Section 5, we define the space-time approachability and design an approachability based data forwarding algorithm to show how to use the partial shared traveling information in data forwarding. In Section 6, we introduce two experiment data sets and state the problems of trace preprocessing, RSU deployment and contact extraction. We conduct extensive experiments and analyze our results in Section 7. The implementation of the approachability based algorithm in the hybrid communication structure is described in Section 8. Discussions about the problems of network overhead and incentive mechanism are provided in Section 9. Finally, we conclude the paper in Section 10.

2 Related Work

Recent reviews [21], [22] and [23] describe existing data forwarding algorithms in vehicular networks. Some studies that use the distance based way to forward data, called geography based methods [15]–[17]. However, the problem of dead-end road exists and there may not have vehicles in the prospective road [24]. Although some of them adjust the algorithms to address these problems, the computation of the large road information costs highly.

Some studies want to alleviate the dependency on the road information. Jeong et al. gave some representative trajectory based methods [9], [10]. In literature [9], each vehicle calculated the expected delivery delay to the closest RSU, and added it to its periodic beacon to inform neighbors. The expected delivery delay was calculated assuming constant vehicle speed, few wireless hops initially from RSU, and carrying messages until the next intersection. The algorithm sorted roads at intermediate intersections by geographically shortest paths, to establish forwarding priority. Thus, a message was forwarded to a vehicle with minimum expected delivery delay [21]. The similar work [10] was also done by Jeong et al. to study the data forwarding from RSU to vehicle. However, these algorithms rely on information such as speed, trajectory and direction to predict the time delay. These information are not stable, which will limit the execution of the methods.

We now review contact based methods. In Prophet [18], each node maintained the encounter history with other nodes, and the routing decision was made based on the encounter probability. Wu et al. [11] gave a novel work of infrastructure-assisted routing in vehicular networks by using Markov Chain to predict the future encounter probability. Specially, they considered the problem of buffer limits. In GeoMob [19], authors studied macroscopic mobility pattern and microscopic mobility pattern to predict the future region where a vehicle would go to. The message was given
to the vehicle which was most likely to go to the destination region. When the message was delivered to the destination area, the message was spread to all vehicles in this area.

Vehicle behaviors have some social characteristics, for example, community and centrality, that can be exploited. Some social based methods\cite{5, 8, 20} from mobile ad-hoc networks were also applied in vehicular networks. For example, the work ZOOM\cite{6} is the first paper that uses contact-level mobility (obtained by Markov Chain) together with the social-level mobility (obtained by computing the ego centrality) to forward data. The paper demonstrates that capturing social-level mobility as a complementary counterpart of contact-level priors can significantly improve the performance of opportunistic data forwarding.

Some studies related to the privacy preservation are specially reviewed in Section 4.2.

3 System Model and Assumptions

In this work, we mainly focus on the problem of data forwarding in the underlying network. So, we model this underlying network with RSU support as a dynamic graph which can be defined as a time sequence of network graph, denoted by \( G = \{ G_0, G_1, ..., G_t, ... \} \), where \( G_t = (V_t, E_t) \) represents a time dependent network snapshot recorded at time \( t \); \( V_t \) denotes the set of nodes, including the set of vehicles and the set of stationary RSUs; \( E_t = \{ (u, v) | u, v \in V_t \} \) denotes the edge set. Both node and edge sets change over time. The edges in the network are aggregated on the basis of a median based sliding window mechanism\cite{25}.

Let \( l(u,v,t) = 1 \) denote that there starts a contact between node \( u \) and \( v \) at time \( t \) (\( 0 \leq t < \infty \)). Then, we have \( \sum_{t=t_{\text{now}}-\Delta}^{t_{\text{now}}} l(u,v,t) \) denote the overall numbers of contacts between node \( u \) and \( v \) from time \( t_{\text{now}} - \Delta \) to \( t_{\text{now}} \), and have \( \sum_{t=t_{\text{now}}-\Delta}^{t_{\text{now}}} l_i(t) \) denote the overall numbers of contacts for all nodes from time \( t_{\text{now}} - \Delta \) to \( t_{\text{now}} \), \( 0 < \Delta < t_{\text{now}} \). We define the encounter ratio between node \( u \) and \( v \) at current time \( t_{\text{now}} \) as:

\[
e(u,v,t_{\text{now}}) = \frac{\sum_{t=t_{\text{now}}-\Delta}^{t_{\text{now}}} l(u,v,t)}{\sum_{t=t_{\text{now}}-\Delta}^{t_{\text{now}}} l_i(t)}
\]

Note that, we assume \( e(u,v,t_{\text{now}}) = e(v,u,t_{\text{now}}) \) by assigning the larger value to the other.

So, in the median based sliding window mechanism, according to the encounter ratios between any two nodes, we filter the edge between two nodes if the encounter ratio value of the two nodes is below the median of encounter ratio values among all nodes. Note that, the window length of sliding window mechanism is usually empirically determined\cite{5}.\cite{25}. Studying different window length (time granularity) would lead to more interesting findings\cite{27}.

---

1. In the paper, we take the length of sliding window \( \Delta \) as a constant, not a variable. We have \( \Delta = 6\times3600s \), since the social based comparison algorithms\cite{5, 6} used in the experiment are all set this value.
2. In the following sections, for brevity, denote the current time \( t_{\text{now}} \) by \( t \), without confusion.

4 Traveling Information Sharing under Privacy Preservation with Delusive Paths

4.1 Partial Shared Traveling Information

Nowadays, many vehicles are equipped with navigator. Intelligent navigator could give the driving routes with avoiding some congested intersections due to accidents, rush hours or municipal construction. Thus, each vehicle can easily obtain the knowledge of its forthcoming traveling route. Besides, in vehicular networks, each vehicle has the ability of joining in social interactions. According to the requirement of privacy preservation, vehicles can share some of them forthcoming traveling route information (position and time) with other vehicles to get the high quality routing service, of which the incentive mechanism is discussed in Section 10.2. The shared traveling information can be nonconsecutive, just random.

Assume that the vehicle will share some traveling route information when it meets another vehicle. Let sequence \( P_u^i = \{ p_u(1), t_u(1) >, < p_u(2), t_u(2) >, ..., < p_u(i), t_u(i) >, ..., < p_u(k), t_u(k) \} \) stand for the number of \( k \) partial shared traveling information of vehicle \( u \) at a encounter, where \( p_u(i) \) denotes the \( i-th \) key position that the vehicle will pass in the future, \( t_u(i) \) denotes the corresponding arrival time at \( i-th \) key position, \( 1 \leq i \leq k, k \in N^+ \). The value of \( k \) varies for different vehicles. It can be obtained through the following delusive paths privacy-preserving scheme. Note that, if a vehicle has the strong privacy demand, it can choose not to open its traveling route information totally. The partial shared information can be seen as a kind of deterministic information (absolute geographic trajectory) which is useful for data forwarding in vehicular networks.

4.2 Delusive Paths Privacy-preserving Scheme

In different applications and networking architectures, the requirements of privacy preservation are various. For example, the social relationship, the user behavior pattern, identification and some sensitive personal attributes can all become the objects of privacy preservation.

Before we give a privacy preserving scheme, we should investigate what kind of danger we may suffer and what kind of danger we can avoid. In previous studies, some applications use the popular C/S model, in which the centralized server collects and knows the global information. As an untrusted third-party, it may leak the information or deduce some sensitive personal information on the basis of the big data it owns. Some adversaries utilize the machine learning method or explore the identification correlation to get rules or models, and further derive the sensitive information. Then, the differentially private method\cite{28},\cite{29}, privacy check based data releasing method\cite{30}, generating dummy data method\cite{31}, k-anonymous technology\cite{32} and their varieties are often used to prevent the privacy leak. Besides, some applications are based on P2P model, like searching a friend’s approximate location, finding nearest friends. So, the aim of privacy preserving is to protect the adversary from profiling the users and exposing some sensitive information. Then, some multi-party based homomorphic encryption methods are developed\cite{33}-\cite{35}. Specially, with respect to the data forwarding, the conditional privacy-preserving authentication technique\cite{36} is used to protect
the data packets from being captured and analyzed by the adversary; and the bilinear paring technique is used to filter “junk” packets, like spam emails, when doing data forwarding in the network.

4.2.1 Trace Privacy
In our work, the vehicle will share partial traveling information when it meets another vehicle. The server in our communication architecture does not store the global information about the partial shared traveling information by each vehicle. So, some serious dangers/attacks from the server can be avoided. When sharing the partial traveling information, the vehicle (an information provider) requires to ensure the trace privacy. We say an information provider has trace security if the information receiver cannot recover the trace from a series of shared information (including the position and the arrival time) from the provider.

4.2.2 Delusive Paths Algorithm
Assume that the adversary (information receiver) has the knowledge of the city map. In our privacy-preserving scheme, we use the delusive paths algorithm to ensure the vehicle trace can not be recovered. The idea of the delusive paths algorithm is when the vehicle share a new traveling information tuple \( < p_u(j), t_u(j) > \), the algorithm will check the number of paths between the new traveling information \( < p_u(j), t_u(j) > \) and the last shared information \( < p_u(i), t_u(i) > \). If the number of paths exceeds one, the new traveling information is said to be safe. Otherwise, we prohibit the new information being released/shared to the encountered vehicle. The delusive paths algorithm guarantees any two shared traveling tuples have \( 2^n \) paths, \( n \geq 1 \), which makes the adversary can not guess the exact path the vehicle will pass by. We give an illustration of the delusive paths algorithm in Fig. 2.

The delusive paths algorithm is described in the following steps:

1. We have the new traveling information tuple \( < p_u(j), t_u(j) > \) and the last shared traveling information tuple \( < p_u(i), t_u(i) > \). The positions \( p_u(j) \) and \( p_u(i) \) are obtained through the GPS.
2. The road information is stored in the form of map node graph. Each node has its neighbors. The Euclidean distance between two nodes represents the length of a line in the map. The lines among some consecutive nodes constitute a road.
3. The candidate map nodes set for calculating the paths between tuple \( < p_u(j), t_u(j) > \) and \( < p_u(i), t_u(i) > \) is selected in a circle area in the map. The center of the circle is the middle point in the straight line between position \( p_u(j) \) and \( p_u(i) \). The diameter of the circle is set two times of the Euclidean distance between position \( p_u(j) \) and \( p_u(i) \). This limits the size of the map nodes set in our calculation and can omit some too long paths between the two tuples.
4. We find two map nodes \( node_i \) and \( node_j \) as the approximate/nearest nodes for position \( p_u(i) \) and \( p_u(j) \) respectively.
5. According to the maximal speed of a vehicle (in the experiment, we set it 120km/h), we calculate the maximal distance \( MD \) that a vehicle can drive between time \( t_u(i) \) and \( t_u(j) \).
6. Then, if the number of paths between tuple \( < p_u(j), t_u(j) > \) and \( < p_u(i), t_u(i) > \) does not exceed one.
7. From the begin node \( node_i \), we find one of its neighbors and use the recursive method to construct the path between the begin node \( node_i \) and the end node \( node_j \).
8. The recursion continues until the length of the path exceeds the maximal distance \( MD \) or the map node achieves the end node \( node_j \). All the qualified map nodes in finding the path in Step (7) are stored in the stack.
9. We use the hash table to store the number of paths between two shared traveling information tuples.

5 Space-Crossing Community Structure
In vehicle networks, due to daily activities of drivers (e.g., on and off duty, outing and shopping), vehicle movement behaviors show a characteristic of cluster or community. This social attribute reflects the law of object interactions in the underlying network clearly. The social community can be seen as a kind of nondeterministic information (node relative relationships) which has been proved useful in data forwarding in opportunistic networks. In different application scenarios, we can obtain different community structures through different community detection methods (see the recent review papers by Fortunato and Lancichinetti et al.). In our paper, we have three demands when doing the community detection: 1) the detection method can be used in a dynamic environment; 2) the detection method has the ability of handling the underlying structure with RSU support; 3) the detection results do help the data forwarding in opportunistic networks and can reflect the positive role of RSUs. So, based on above three requirements, we choose Space-Crossing Community Detection to tackle our vehicular network. Specially, for the third point, many previous community detection methods just simply treat RSUs as the common nodes (vehicles). These methods can not reflect the true communication connectivity capability among some long-distance nodes in far areas through RSUs. However, the space-crossing community detection method does well in this aspect.

Here, we briefly introduce the space-crossing community detection method. In this method, every mobile user and RSU can be viewed as an independent participant. First, due to frequent interactions (physical proximity) between nodes (including vehicles and RSUs), there will form some
dense groups with internal tight-knit communication nodes. Then, with the help of connectivity among some RSUs, a part of long-distance nodes that are in different physical proximity communities containing RSUs could have the strong capability to communicate with each other. Based on the fact of long-distance connectivity, RSUs execute combination criteria $S^a$ and $S^b$ [25] (details are moved in APPENDIX) on those physical proximity communities to form some groups across the geographical space, called space-crossing communities. The space-crossing community breaks the space limit, i.e., two long-distance vehicles may belong to the same community and share the common community attributes. This breakthrough change brings about the advantage in social based data forwarding, which has been demonstrated by Li et al. [25]. We give an illustration of space-crossing community structure in Figure 3.

6 The Power of Partial Shared Traveling Information in Data Forwarding

In this section, we will present how to use the shared traveling information to improve the data forwarding in community characterized vehicular networks.

From Section 3 and Section 4 we can obtain two kinds of guiding information when doing data forwarding in vehicular networks. First, if the vehicle is willing to share partial of its traveling route information, we can obtain some relative deterministic information about where the vehicle will go. Second, due to frequent vehicle interactions, through space-crossing community structure, we can obtain some nondeterministic information about where the vehicle will go.

In the paper, we study the unicast session among vehicles. First, based on the traveling route information, we give a definition of space-time approachability of a vehicle in Section 6.1. Then, based on space-crossing community structure, we use the node local activity and give the social approachability of a vehicle. These two approachability properties will predict a vehicle’s future movement from different perspectives. Finally, we design an approachability based data forwarding algorithm in vehicular networks.

6.1 Space-Time Approachability

**Definition 1 (Approximate Destination Set).** Assuming that, for a destination vehicle $d$, from last $\Delta$ length of sliding window to current time $t$, vehicle $d$ has passed several RSUs. Then, for every passed RSU $r$, if $e(r, d, t)$ is larger than the median of $\{e(r', d, t)|r' \in \{\text{all the passed RSUs by vehicle } d\} \land e(r', d, t) \neq 0\}$, we will put RSU $r$ into a new set, called approximate destination set $R_t(d)$.

**Definition 2 (RSU Importance).** Let $CC_t(r)$ denote the RSU connected component containing RSU $r$. Based on Definition 1 we define RSU importance for every $r \in R_t(d)$ as $w(r, d, t)$.

- If $|R_t(d) \cap CC_t(r)| = 1$, we will have $w(r, d, t) = e(r, d, t)$;
- If $|R_t(d) \cap CC_t(r)| > 1$, we will have $w(r, d, t) = \sum_{r' \in R_t(d) \cap CC_t(r)} e(r', d, t)$, where $e(r, d, t)$ and $e(r', d, t)$ denote the encounter ratio between $r - d$ and $r' - d$ at time $t$, respectively. For $r \notin R_t(d)$, we have $w(r, d, t) = 0$.

**Definition 3 (Space-Time Approachability).** For a vehicle $u$, it shares its $k$ partial traveling route information $P^k_u = \{ < p_u(1), t_u(1) >, < p_u(2), t_u(2) >, ..., < p_u(i), t_u(i) >, ..., < p_u(k), t_u(k) > \}$ at an encounter, as described in Section 4.1. Taking the $i$-th position information $p_u(i)$ as a center, we can find a RSU set $I_u(i)$ which denotes the nearby RSUs within the vehicle-RSU communication range. So, based on approximate destination set $R_t(d)$, we let $R_t(d) \cap I_u(i)$ denote the set of proximate destinations (RSUs) approached by vehicle $u$ in its $i-th$ shared position.

We define the space-time approachability for vehicle $u$ whose message destination is vehicle $d$ as $TSA(u, d)$, having

$$TSA_t(u, d) = \frac{\sum_{i=1}^{k} \sum_{r \in R_t(d) \cap I_u(i)} w(r, d, t)}{t_u(i) - t},$$

where $w(r, d, t)$ denotes the RSU importance of $r$ at current time $t$ with the related destination $d$, and we assume that $t_u(i) - t > 0$. Note that, we allow $TSA_t(u, d) = 0$, if vehicle $u$ rejects to share traveling route information with the encountered vehicles.

Here, we give an example of calculating space-time approachability in Fig 4. Supposing that, there are two vehicles, the vehicle $u$ and $v$ share their partial traveling route information as illustrated in the box in the picture. RSU $r_1, r_2, r_3, r_4$ are in the approximate destination set $R_t(d)$. So, we have

$$TSA_t(u, d) = \frac{w(r_1, d, t)}{t_u(1) - t} + \frac{w(r_2, d, t)}{t_u(2) - t};$$

$$TSA_t(v, d) = \frac{w(r_3, d, t)}{t_v(1) - t} + \frac{w(r_4, d, t)}{t_v(2) - t}.$$
Definition 6 (Social Approachability [25]). Given two activity vectors $A_t(u)$ of node $u$ and $A_t(v)$ of node $v$, we define social approachability between $u$ and $v$ at time $t$ as $SA_t(u,v)$, having

$$SA_t(u,v) = A_t(u) \cdot A_t(v),$$

where the operator $\cdot$ denotes the inner product of vectors.

Space-crossing community can reflect the node’s belonging property from the view of physical communication among nodes. Local activity can show the importance of a node in a certain space-crossing community. Based on the destination-oriented aim, a larger social approachability can not only guarantee to find a node that has similar distribution of the node’s belonging communities with the destination, but also can gain larger local activity components in the vector. Thus, a node having larger social approachability with the destination indicates that it has higher chance to approach to the destination.

6.3 Approachability Based Measurement

Definition 7 (Approachability Based Measurement). For a session, the message is delivered from node $u$ to destination node $d$, and now, node $u$ meets another node $v$. Then, we define a fair approachability based measurement $M_t(u,d)$ and $M_t(v,d)$ for node $u$ and node $v$ at time $t$.

If $TSA_t(u,d) \neq 0$, $SA_t(u,d) \neq 0$, $TSA_t(v,d) \neq 0$ and $SA_t(v,d) \neq 0$, we will have

$$M_t(u,d) = TSA_t(u,d) \times SA_t(u,d);$$
$$M_t(v,d) = TSA_t(v,d) \times SA_t(v,d).$$

If $TSA_t(u,d) \times TSA_t(v,d) = 0$, i.e., the vehicle(s) do(es) not like to share traveling route information, we will have $M_t(u,d) = SA_t(u,d); M_t(v,d) = SA_t(v,d)$.

If $SA_t(u,d) \times SA_t(v,d) = 0$, i.e., the social approachability between node $u/v$ and destination $d$ is totally irrelevant, we will have $M_t(u,d) = TSA_t(u,d); M_t(v,d) = TSA_t(v,d)$.

For other cases, $M_t(u,d) = M_t(v,d) = 0$.

6.4 Approachability Based Data Forwarding Algorithm in Different Phases

Phase 1: Vehicle → Vehicle

When a vehicle holds a message and it meets another vehicle, since each vehicle has its activity vector, if they like to share some partial traveling route information, they will calculate their approachability based measurement with the destination respectively. The message holder tries to send the message to a node which has larger approachability based measurement than itself and let the node send the message to the destination consecutively. If the encountered node has a smaller approachability based measurement than the message holder, the message holder will inquire the current neighbors of the encountered node. If the set of current neighbors contains the destination, then the message will be transmitted to the encountered node.

Phase 2: Vehicle → RSU

When a vehicle holds a message and it meets a RSU, first, the vehicle will detect whether the RSU is in the approximate destination set $R_t(d)$. If yes, the message is delivered to the RSU directly; if no, the vehicle will transmit the message to the encountered RSU with a larger approachability based measurement than the vehicle itself. Otherwise, the calculation is enabled by beacon messages sent by base stations periodically, so that the message holder can learn the knowledge of the destination.
message holder will inquire the current neighbors of the encountered RSU. If the set of current neighbors contains the destination, then the message will be transmitted to the encountered RSU.

**Phase 3: RSU → Vehicle**

When a RSU holds a message, it first delivers the message to other RSUs in its common connected components (repeated copies are not allowed in the same RSU). Then, the RSUs will use the approachability based measurement to deliver the message to the passing vehicles with larger approachability based measurement than the RSUs themselves. Otherwise, the message holder will inquire the current neighbors of the encountered vehicle. If the set of current neighbors contains the destination, then the message will be transmitted to the encounter vehicle.

In above three phases, after the message holder transmits the packet to the encountered node, the holder removes the packet from its buffer.

### 7 Vehicle Trace Data Analysis

For the purpose of our study, we use two large GPS based vehicle mobility traces. One is San Francisco Cabspotting in America, the other is Shanghai Taxi Movement in China.

#### 7.1 Data Sets

**San Francisco Cabspotting:** The data contains GPS coordinates of 536 taxis collected over a period of three consecutive weeks in the San Francisco Bay Area. Each taxi is equipped with a GPS receiver and sends a location-update (timestamp, identifier, geo-coordinates) periodically. The location-updates are quite fine-grained. The average time interval between two consecutive location updates is less than 60 sec [42].

**Shanghai Taxi Movement:** The data set was collected by our research group in Shanghai, China, approximately 3000 taxis from January to September, 2006. The taxi periodically sends reports back to the data collector via an on-board GPS-enabled device. Each taxi reports every 60 seconds. The information contained in the data set includes the vehicle ID, the latitude and longitude location, timestamp, onboard, vehicle moving speed and heading direction. The pattern of taxi mobility is diverse with a much larger spatial and temporal coverage. Shanghai Taxi Movement covers a large portion of Shanghai city.

#### 7.2 Trace Preprocessing

GPS accuracy might be affected by many factors (e.g., the signal multipath error and the device clock error). In above two vehicle data sets, there exist some noisy data (e.g. some data are in the sea). So, a noise suppression method which use map polygon clipping to filter the noisy data is excused on the vehicle data sets. By counting the number of GPS records of all taxis, the results of trace preprocessing are shown in the Fig.5 and Fig.6 respectively. Note that, a more precise method of dealing with GPS accuracy problem can be seen in literature [43].

![Fig. 5. Subfigure (a) shows the original vehicle data distribution of San Francisco Cabspotting. Subfigure (b) shows the preprocessing result after using noise suppression.](image)

![Fig. 6. Subfigure (a) shows the original vehicle data distribution of Shanghai Taxi Movement. Subfigure (b) shows the preprocessing result after using noise suppression.](image)

#### 7.3 Deploying RSUs in the Map

Of San Francisco Cabspotting and Shanghai Taxi Movement, 70 and 352 RSUs are deployed in the network, respectively. Efficiently deploying RSUs is crucial to improve packet forwarding efficiency in vehicular networks. For different network scenarios and research purposes, the optimal RSU deployment strategies are different [36], [44]–[46]. However, this problem is not the main focus of our paper.

In the paper, we discuss three configurations of RSU deployment. The first configuration has RSU locations selected in dense vehicle traffic area, like San Francisco northeastern area and Shanghai center area; the second configuration has RSU locations selected in sparse vehicle traffic area; the third configuration has RSU locations selected uniformly in the vehicular network. There are two illustrations of RSU deployment, shown in Fig.7.

#### 7.4 Contact Extraction and Results of Space-Crossing Community

Most of vehicle data sets contain the information of GPS, but do not give the contact records. We assume that any vehicle-vehicle pairs and vehicle-RSU pairs can communicate when they are in the communication transmission range. The shadowing effect of buildings is not considered. We use the approximate method described in [47] to get contacts among vehicles and RSUs.

Besides, the amount of data that can be transmitted between two devices depends on the contact durations and the communication technology (e.g., WiFi). For simplification, we assume that each interaction between two nodes can support a successful data delivery.

According to the extracted contacts in vehicle data sets, we can use the median based sliding window mechanism in
Fig. 7. The red triangle represents the RSU. Subfigure (a) shows the third configuration of RSU deployment in San Francisco. Subfigure (b) shows the first configuration of RSU deployment in Shanghai.

Fig. 8. Different color nodes group into different space-crossing communities. Subfigure (a) shows the space-crossing communities in San Francisco Cabspotting captured at 22:00:00, 17th May. Subfigure (b) shows the space-crossing communities in Shanghai Taxi Movement captured at 18:00:00, 24th August. Here, for the simplicity of representation, we only focus on the communities, i.e., the same node may be depicted repeatedly in different communities in the pictures.

Section 7 to form dynamic graphs. Then, the nodes execute the space-crossing community detection method based on the dynamic graph and obtain the result of communities. We capture different space-crossing community structure on San Francisco Cabspotting and Shanghai Taxi Movement datasets, shown in Fig.8.

8 EVALUATION
8.1 Simulation Setup

We use an open-source simulator ONE [48] for simulation. We import vehicle traces and the road map for node mobility. The source and destination pairs are chosen randomly among all vehicles. Each simulation is repeated 20 times with different random seeds. Except the parameters described in Section 7.1, the other simulation settings are summarized in Table 1.

| Parameter                  | Settings                  |
|----------------------------|----------------------------|
| vehicle transmission range | 100m                       |
| RSU transmission range     | 300m                       |
| V2V transmission speed     | 250KBps                    |
| V2R transmission speed     | 1MBps                      |
| vehicle buffer size        | 5MB                        |
| RSU buffer size            | 5MB                        |
| packet generation interval | 200s-300s randomly         |
| packet size                | 50KB-100KB randomly        |
| sliding window size        | 6*3600s                    |
| simulation time            | Cabspotting: 2071531s; Shanghai: 2073599s |

8.2 Experiment Results and Analysis

8.2.1 Compared Algorithms

In this section, we compare our algorithm with two popular social based data forwarding algorithms (BUBBLE RAP, ZOOM) in vehicular networks.

- BUBBLE RAP [5] provides a hierarchical forwarding strategy. A node first bubbles the message up the hierarchical ranking tree using the global centrality. When the message reaches the community of the destination node, local centrality is used instead of the global centrality.

- ZOOM [6] considers the factors of contact-level mobility (captured by Markov Chain) and social-level mobility (captured by ego betweenness) in vehicular networks. First, it uses the inter contact time to predict the next contact time. The forwarding criterion is that the message holder chooses a candidate node with minimal contact time to the destination as a relay. When this criterion fails to show the priority between two encountered nodes, ZOOM will use the ego betweenness to choose the next relay.

8.2.2 Comparison Fairness

Note that, for the sake of fairness, we select settings or parameters which bring about the best performances for above two comparison algorithms. Additionally, since the comparison algorithms are not based on the underlying network with RSU support, we use the fair RSU strategy (spreading the messages in RSU connected component) for above comparison algorithms. That is to say, ZOOM and BUBBLE RAP also can be used in the environment with RSUs support. ZOOM and BUBBLE RAP both have the same copies strategy on RSUs with our approachability based algorithm.

8.2.3 Metrics

The performance of the proposed Approachability Based Algorithm is evaluated in the following metrics: delivery ratio, overhead ratio and average latency.

- Delivery Ratio: the ratio of the number of successfully delivered messages to the total number of created messages.

- Average Latency: the average messages delay for all the successful sessions.

- Overhead Ratio: the proportion of the difference between the number of relayed messages and successfully delivered messages out of the successfully delivered messages.

8.2.4 General Comparison Experiment

Fig.9–Fig.14 show the delivery ratio, overhead ratio and average latency of Approachability Based Algorithm, ZOOM and BUBBLE RAP in San Francisco Cabspotting and Shanghai Taxi Movement for three kinds of RSU deployments. We can see, on average, the delivery ratio of Approachability Based Algorithm achieves best among those compared algorithms while the overhead ratio and average latency are lowest.

In terms of delivery ratio, the peak value of Approachability Based Algorithm is higher than ZOOM and BUBBLE RAP and the time of emerging the peak value of Approachability Based Algorithm is later than ZOOM and
Algorithm outperforms ZOOM with 8 percent and 9.63 percent, respectively in Fig.12 (a) and Fig.13 (a).

In terms of overhead ratio and average latency, Approachability Based Algorithm keeps a low overhead ratio and average latency than compared algorithms in both data sets. Together with the delivery ratio, we can see, Approachability Based Algorithm does not use a long delay to exchange for a good delivery ratio. Therefore, the traveling route information indeed help messages quickly delivered to the approximate destination set with low overhead ratio and delay. Note that, in San Francisco Cabspotting, the overhead of ZOOM is smaller than BUBBLE RAP, but in Shanghai Taxi Movement, the opposite is true. This is because the contacts in Shanghai...
Taxi Movement are denser than San Francisco Cabspotting, which results in ZOOM, who first uses the contact-level mobility to forward data, generating many relays (hops).

8.2.5 Rush and Non-Rush Hours Comparison Experiment

Especially, we do experiments to verify the power of the partial shared traveling information independently. In vehicle trace data sets, we observe a phenomenon of the vehicular network: the traffic is different in different time period of a day, i.e., existing the rush hours and non-rush hours, as illustrated in Fig.15(a) and Fig.16(a).

Fig.15(a) and Fig.16(a) reflects the distribution of the number of contacts captured in San Francisco Cabspotting (26th MAY, 2008) and Shanghai Taxi Movement (5th AUGUST, 2006), in which the time of 8:00-9:00, 11:00-13:00, 16:00-20:00 are the rush hours approximately. The number of contacts in rush hours is more than that in non-rush hours. Similarly, the distribution of the number of space-crossing communities also shows the rush and non-rush hours characteristic. Fig.15(b) and Fig.16(b) reflect the distribution of the number of space-crossing communities in San Francisco Cabspotting (26th MAY, 2008) and Shanghai Taxi Movement (5th AUGUST, 2006).

In the rush hours, the social communities and the partial shared traveling information all work in data forwarding. While, in the non-rush hours, due to the decrease of social communities, the data forwarding mainly relies on the shared traveling information. Fig.15(c) and Fig.16(c) are the results of delivery ratio test on rush and non-rush hours in San Francisco Cabspotting (26th MAY, 2008) and Shanghai Taxi Movement (5th AUGUST, 2006). In order to avoid the cumulative effect of time and packet generation, each hour is independently tested. From the results, we can see clearly that the delivery ratio do not fluctuate largely on rush and non-rush hours from time 5:00-20:00. This can demonstrate that the shared traveling information indeed helps the data forwarding in vehicular networks. For time of 0:00-5:00 and 20:00-24:00, since the number of contacts is few, the social communities and the shared traveling information both play a little role in data forwarding.

8.2.6 Different Parameters Comparison Experiment

For further performance study, we investigate an important factor–buffer size. We see how the performance of data forwarding reacts to the buffer size in vehicular networks.
Subfigure (b) gives a distribution of the number of space-crossing communities on 5th AUGUST, 2006. Subfigure (c) shows the hourly delivery ratio of on 5th AUGUST, 2006.

Fig. 16. Simulation Results on Shanghai Taxi Movement. Subfigure (a) gives a distribution of the number of contacts on 5th AUGUST, 2006.

APPENDIX Algorithm 1 and 2. Through neighbor to neighbor way, each node obtains the members in its belonging community(ies) within a limited delay. The detailed procedures are provided in APPENDIX.

The density function criterion in APPENDIX is a monotonically increasing function, it can control the size of community, i.e., the community would not be too large. As the number of nodes increases, the threshold for forming a community also becomes strict. Thus, to some extent, this control can alleviate the delay produced by acquiring the information of nodes within a community.

The RSU locations are selected in the dense traffic area. Due to space limit, we omit other cases. In San Francisco Cabspotting, setting TTL as 720s, we vary the buffer size from 5MB to 1MB. Fig 17 (a) shows the result. When the buffer size decreases, all algorithms have low packet delivery ratio. However, at every freezing value of buffer size, our algorithm achieves better comparing with ZOOM and BUBBLE RAP. It demonstrates the superiority of our algorithm. In Shanghai Taxi Movement, setting TTL as 600min, we also vary the buffer size from 5MB to 1MB. Fig 17 (b) shows the result. The similar trend also appears in Shanghai Taxi Movement.

The destination vehicle adopts the median value to obtain a set of RSUs having a relative higher encounter ratio with the destination in a time window. So, the calculation of ADS can be obtained in a distributed way by the destination itself. Second, each node knows the encounter ratio between any two nodes in its belonging space-crossing community(ies) through neighbor to neighbor way within a limited delay. So, the local activity of each node can be obtained. Similar to the ADS, the calculation of the activity vector for a destination vehicle can be done by itself. In the phase of data forwarding, the base stations are also used to manage two jobs: 1) receiving the reports about the ADS and activity vector of the destination vehicle periodically; 2) sending small/beacon messages about the ADS and activity vector of the destination to the source. Finally, on the basis of the information about the destination, the space-time approachability and the social approachability can be calculated by a node itself. Therefore, approachability based data forwarding algorithm can be done by vehicles and RSUs.

9 IMPLEMENTATION

To begin with, a node has perfect knowledge of its neighbors and some local approximation knowledge captured by its neighbors. Some required information can be transferred through neighbor to neighbor way, which is also used in Hui’s distributed community detection [49].

So, about the space-crossing community detection, based on the knowledge that a node can learn, the space-crossing community detection can be done by node itself in a distributed way. The detailed procedures are provided in APPENDIX Algorithm 1 and 2. Through neighbor to neighbor way, each node obtains the members in its belonging space-crossing community(ies) within a limited delay. In the phase of community detection, the base stations are used to manage two jobs: 1) executing the median based sliding window mechanism to obtain the dynamic graphs; 2) receiving the reports about the belonging community(ies) from nodes periodically.

Then, about the data forwarding, first, the approximate destination set (ADS) can be obtained by the destination vehicle itself. The destination vehicle moves in the vehicular network. It collects the information about its encountered RSUs in recent time window (the length of the time window is set six hours, which is same as BUBBLE RAP and ZOOM). The destination vehicle adopts the median value to obtain a set of RSUs having a relative higher encounter ratio with the destination in a time window. So, the calculation of ADS can be obtained in a distributed way by the destination itself. Second, each node knows the encounter ratio between any two nodes in its belonging space-crossing community(ies) through neighbor to neighbor way within a limited delay. So, the local activity of each node can be obtained. Similar to the ADS, the calculation of the activity vector for a destination vehicle can be done by itself. In the phase of data forwarding, the base stations are also used to manage two jobs: 1) receiving the reports about the ADS and activity vector of the destination vehicle periodically; 2) sending small/beacon messages about the ADS and activity vector of the destination to the source. Finally, on the basis of the information about the destination, the space-time approachability and the social approachability can be calculated by a node itself. Therefore, approachability based data forwarding algorithm can be done by vehicles and RSUs.

10 DISCUSSION

10.1 Overhead

Here, we give a discussion about the comparison of network overhead between our algorithm and other relevant algorithms.

For geographical based algorithms, they like to use information of the vehicle speed, direction, road segment and GPS to predict a good routing path by base stations in a centralized way. However, in our routing scheme, the base stations do not need to know and utilize the GPS information to find routing paths in a centralized way. This is the main difference between our data forwarding algorithm and other geographical based algorithms. In our algorithm, the data forwarding is done by vehicles and RSUs in an opportunistic and distributed way. The base stations are only used to send some messages about the destination to source nodes to assist them forwarding data.
Besides, some contact based methods like to use the history contacts to predict the next contact time with the destination. They put much emphasis on the node next contact. This kind of method is prone to lose many potential useful contacts which may be not directly with the destination. In our approachability based algorithm, first, the partial shared traveling information provides the deterministic information of the future contacts; second, the space-crossing community structure can provide the nondeterministic information of the future contacts; Our algorithm only concerns the probability of the next relay approaching to the destination, not the specific contact, which leads to a wider range of contacts than the previous contact based methods.

Our algorithm indeed introduces the overhead in doing space-crossing community detection. Besides the base stations also cost overheads in doing some extra global assistant work. However, we think the vehicle (unlike the sensor) has enough power to do the distributed community detection, and base stations offload the data forwarding to the vehicles and RSUs, which is worth for achieving a good data forwarding performance.

10.2 Incentive Mechanism

We provide an appropriate incentive mechanism that can attract more vehicles to participate and well perform in the routing service.

Our data forwarding can be seen a kind of cooperative routing service. Be similar to the crowdsourcing, our cooperative routing service also require the incentive mechanism to maintain the service. It also takes advantage of vehicles’ willing to collaborate toward a continuous data harvesting process. Be different from crowdsourcing, the cooperative routing service does not like the fractional crowdsourcing applications, such as Yahoo! Answers [50] for knowledge sharing, Amazon Mechanical Turk [51] for using human intelligence to perform tasks, CreekWatch [52] for tracking pollution levels in water resources, Sensorly [53] for making cellular/WiFi network coverage maps and PMG [54] for map generation, in which all of them require a centralized server to compute and publish a unified result as a service to all the participants. Our cooperative routing service is provided to each participating vehicle in a decentralized way, not through a centralized server, like “dis-crowdsourcing”.

The philosophy of “all for one, one for all” runs throughout the incentive mechanism. All the participating vehicles can enjoy the convenient routing served by other vehicles. At the same time, if vehicles want to obtain this service, they also need to provide some utilizable information for other vehicles and get the corresponding reputation scores. More concretely, the server sets the reputation scores based on the traveling information shared by each vehicle. Fig. 18 gives an illustration of the working process of the incentive mechanism.

The service provider (information provider) and the routing service demander (information receiver) both pay attention to the other reputation score on each side. So, the data forwarding scheme stated in Section 8 can be extended to a new scheme with considering the reputation score. In the new scheme, the information provider will see the score of the routing service demander to decide whether to provide the traveling information. And the routing service demander will combine the approachability based measurement and the score of the information provider to decide whether to deliver the packet to the information provider. A vehicle with a higher reputation score will be given a higher chance to participate in this cooperative service and receive more payments upon the completion of a task. The further relevant study can be done in the future work.

11 Conclusion

In this paper, we investigate how to use a vehicle’s lightweight traveling information (including the forthcoming positions and corresponding arrival time) to forward data in a social community characterized vehicular networks. We propose the concept of space-time approachability which is used to measure the capability of a vehicle approaching to the destination. Then, we describe a high efficient approachability based data forwarding algorithm in the vehicular network with hybrid communication architecture. Through comprehensive simulations, we demonstrate that our algorithm outperforms other social based algorithms applied in vehicular networks. For our future work, we will further explore the incentive mechanism when sharing traveling information in vehicular networks.

References

[1] M. Bakht, M. Trower, and R. H. Kravets, “Searchlight: won’t you be my neighbor?” in Proc. ACM MobiCom 2012.

[2] N. Cheng, N. Lu, N. Zhang, X. S. Shen, and J. W. Mark, “Vehicular WiFi offloading: Challenges and solutions,” Vehicular Communications, vol. 1, no. 1, pp. 13–21, 2014.

[3] E. Daly and M. Haahr, “Social network analysis for routing in disconnected delay-tolerant manets,” in Proc. ACM MobiHoc 2007.
APPENDIX

SPACE-CROSSING COMMUNITY DETECTION AND RELEVANT ALGORITHMS IN [25]

In Space-Crossing community detection method, every mobile user and RSU can be viewed as an independent participant.

At initial network snapshot, based on initial network graph defined in Section 3 the locating community phase of algorithm AFOSC [20] is applied to obtain initial set of physical proximity communities \( PP_1 \). In locating community phase of algorithm AFOSC, for a node \( u \), it first chooses one of its neighbors \( v \). Then, the two nodes and their common neighbors form an initial structure \( C \). Then, a density function criterion

\[
|C_{in}| \geq \left( \frac{|C|}{2} \right)^{1 - \frac{1}{(\frac{2}{3})^D}}
\]

is used to decide whether the structure \( C \) is a physical proximity community, where \( |C_{in}| \) and \( |C| \) denote the set of edges having both endpoints in \( C \) and the number of nodes in \( C \).

Then, based on \( PP_1 \), we use combination criterion \( S^a \) which is presented in Algorithm 1 to obtain the initial set of space-crossing communities \( SC_1 \). In \( S^a \), a certain RSU \( r \) will first check its connected RSU \( r' \) that the serial number is smaller than RSU \( r \). If they are not in the same physical proximity communities and the RSU \( r' \) has not done \( S^a \) combination, we will combine the physical proximity communities containing them into a new space-crossing community. Otherwise, the RSU \( r \) will check another connected RSU \( r'' \) that the serial number is larger than RSU \( r \) and decide whether to form a new space-crossing community.

For subsequent time slot \( t (2 \leq t < \infty) \), based on \( PP_t \) the adaptive finding community phase of algorithm AFOSC [20] is applied to obtain the set of physical proximity communities \( PP_t \). In the adaptive finding community phase of algorithm AFOSC, dynamic network changes are classified into four simple actions: adding new nodes, adding edges, removing nodes and removing edges.

Based on above \( PP_t (2 \leq t < \infty) \), we apply combination criterion \( S^b \) which is presented in Algorithm 2 to obtain the set of space-crossing communities \( SC_t (2 \leq t < \infty) \). In \( S^b \), when a certain RSU \( r \) whose mark of combination criterion \( S^a \) is done finds that the size of its belonging physical proximity communities changes. The RSU \( r \) will first check its connected RSU \( r' \) that the serial number is smaller than RSU \( r \). If they are not in the same physical proximity communities and the RSU \( r' \) has not done \( S^b \) combination, we will combine the new physical proximity communities containing them and update a new space-crossing community. Otherwise, the RSU \( r \) will check another connected RSU \( r'' \) that the serial number is larger than RSU \( r \) and decide whether to update a new space-crossing community.

Algorithm 1 Combination Criterion \( S^a \)

**Input:**
The physical proximity community structure \( PP_1 \) at time slot \( t_1 \).

**Output:**
The space-crossing community structure \( SC_1 \) at time slot \( t_1 \).

1. Define an array \( OCA \), the length of \( OCA \) is the number of RSUs;
2. For \( (k = 0; k < |OCA|; k + +) \) do
3. Set \( OCA[k] = 0 \);
4. /\* \( x = 0 \) denotes the \( k + 1 - \) th RSU have not done \( S^a \) combination /\*
5. End for
6. \( SC_1 \leftarrow PP_1 \);
7. For a RSU \( r - th \);
8. If \( (r-r-th and r-1-th RSUs are not in the same physical proximity communities & the r-1-th RSU has a communication link with the r-th RSU & OCA[r-2] \neq 1) \) then
9. Let \( LA \) denote the set of the labels of physical proximity communities containing the \( r - th RSU \);
10. Let \( LB \) denote the set of the labels of physical proximity communities containing the \( r - 1 - th RSU \);
11. For \( (i = 0; i < |LA|; i + +) \) do
12. For \( (j = 0; j < |LB|; j + +) \) do
13. \( OC \leftarrow \) combine the physical proximity communities containing \( LA[i] - th RSU \) and the physical proximity communities containing \( LB[j] - th RSU \);
14. \( SC_1 \leftarrow SC_1 \setminus \{ \text{the physical proximity communities containing} \ \text{LA[i] - th RSU} \} \cup \{ \text{the physical proximity communities containing} \ \text{LB[j] - th RSU} \} \cup \{ C \} \);
15. End for
16. End for
17. \( OCA[r - 1] = 1 \) and \( OCA[r - 2] = 1 \);
18. Else
19. The same operations (Step 8 - Step 13) are done for \( r+1 - th RSU \);
20. End if
21. End if
22. Update \( SC_1 \);

Algorithm 2 Combination Criterion \( S^b \)

**Input:**
The physical proximity community structure \( PP_1 \) at time slot \( t_1 \).

**Output:**
The space-crossing community structure \( SC_1 \) at time slot \( t_1 \).

1. \( SC_t \leftarrow PP_t \);
2. Define an array \( OCB \), the length of \( OCB \) is the number of RSUs;
3. For \( (k = 0; k < |OCB|; k + +) \) do
4. Set \( OCB[k] = 0 \);
5. /\* \( x = 0 \) denotes the \( k+1 - \) th RSU have not done \( S^b \) combination /\*
6. End for
7. If \( (the size of physical proximity communities containing r-th RSU changes in \( PP_1 \) & OCA[r-1]=1) \) then
8. If \( (r-r-th RSU and r-1-th RSU has a communication link & they are not in the same physical proximity communities & OCB[r-2] \neq 1) \) then
9. Combine the physical proximity communities containing \( r - th RSU \) and the physical proximity communities containing \( r-1-th RSU \) to form new space-crossing communities;
10. Set \( OCB[r - 1] = 1 \) and \( OCB[r - 2] = 1 \);
11. Else
12. The same operations (Step 8 - Step 13) are done for \( r+1 - th RSU \);
13. \( OCB[r] = 1 \) and \( OCB[r-1] = 1 \);
14. End if
15. End if
16. Update \( SC_t \);