Exploiting Active Sub-areas for Multi-copy Routing in VDTNs

Bo Wu, Haiying Shen and Kang Chen
Department of Electrical and Computer Engineering
Clemson University, Clemson, South Carolina 29634
{bwu2, shenh, kangc}@clemson.edu

***Abstract*—In Vehicle Delay Tolerant Networks (VDTNs), current routing algorithms select relay vehicles based on either vehicle encounter history or predicted future locations. The former method may fail to find relays that can encounter the target vehicle in a large-scale VDTN while the latter method may not provide accurate location prediction due to traffic variance. Therefore, these methods cannot achieve high performance in terms of routing success rate and delay. In this paper, we aim to improve the routing performance in VDTNs. We first analyze vehicle network traces and observe that i) each vehicle has only a few active sub-areas that it frequently visits, and ii) two frequently encountered vehicles usually encounter each other in their active sub-areas. We then propose Active Area based Routing method (AAR) which consists of two steps based on the two observations correspondingly. AAR first distributes a packet copy to each active sub-area of the target vehicle using a traffic-considered shortest path spreading algorithm, and then in each sub-area, each packet carrier tries to forward the packet to a vehicle that has high encounter frequency with the target vehicle. Extensive trace-driven simulation demonstrates that AAR produces higher success rates and shorter delay in comparison with the state-of-the-art routing algorithms in VDTNs.**

***Index Terms*—VDTN, Active area, Routing algorithm**

I. INTRODUCTION

Recently, the problem of providing data communications in vehicle networks (VNETs) has attracted a lot of attention. Vehicle Delay Tolerant Networks (VDTNs) create a communication infrastructure composed by vehicle nodes, which offers a low cost communication solution without relying on base stations. In this paper, we focus on routing algorithms VDTNs for data communications.

Current routing algorithms in VDTNs can be classified to three categories: contact [1–4], centrality [5–7] and location [8–11] based routing algorithms. Based on the fact that vehicles which encountered frequently in the past tend to encounter frequently in the future, contact based routing algorithms relay packets according to the encounter history. In centrality based algorithms, a packet carrier forwards the packet to the vehicle with the highest centrality, i.e., the vehicle that can encounter more vehicles. However, in the contact and centrality based algorithms, a packet carrier may fail to find relays that can encounter the target vehicle in a large-scale VDTN with thousands of vehicles on a very large area, leading to low routing efficiency.

Location based routing algorithms predict the future locations of vehicles, find the shortest path from the source vehicle to the target vehicle, and select the vehicles with trajectories on the shortest path as relay vehicles. These algorithms require highly accurate prediction so that relay vehicles and the packet carrier will be close to each other in a certain short distance (e.g., less than 100 meters). However, it is difficult to achieve accurate prediction because vehicles have high mobility and vehicle trajectories are greatly influenced by many random factors such as the traffic and speed of vehicles. Also, since the shortest path is determined without considering the traffic, there may be few vehicles on the path. Therefore, if the selected relay vehicle is missed due to low prediction accuracy, it is difficult to find other relay candidates, which leads to low routing efficiency.

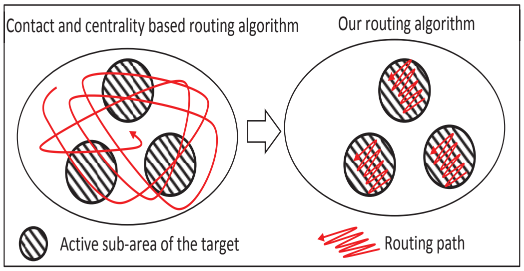


Fig. 1: Current routing algorithm vs. AAR.

Therefore, current routing algorithms cannot achieve high performance in terms of routing success rate and delay. In this paper, we aim to improve the routing performance in VDTNs. We first analyze real Vehicle NETwork (VNET) *Roma* [12] and *SanF* [13] traces and gain the following two observations: i) each vehicle has only a few active sub-areas in the entire VDTN area that it frequently visits, and ii) two frequently encountered vehicles usually have high probability to encounter each other in their active sub-areas, while have very low probability to encounter each other in the rest area on the entire VDTN area. We then propose Active Area based Routing method (AAR) which consists of two phases based on the two observations correspondingly.

As shown in Figure 1, unlike the contact and centrality based routing algorithms that search the target vehicle in the entire VDTN area, AAR constrains the searching areas to the active sub-areas of the target vehicle, which greatly improves the routing efficiency. AAR first distributes a packet copy to each active sub-area of the target vehicle, and then in each sub-area, each packet carrier tries to forward the packet to a vehicle that has high encounter frequency with the target vehicle.

Specifically, AAR consists of the following two algorithms for these two steps.

Traffic-considered shortest path spreading algorithm. It jointly considers traffic and path length in order to ensure there are many relay candidates in the identified short paths to efficiently distribute multiple packet copies. Figure 2 shows an example of the basic idea of our traffic-considered shortest path spreading algorithm. Current location based routing algorithms relay the packet from road intersection a to b through the shortest path (i.e., the dotted line) but fail to consider whether there are enough relay vehicles in the path. If the shortest path only consists of small roads with less traffic, it leads to a long time for a packet to reach the target sub-area. In our spreading algorithm, the packet is routed along the circuitous path (i.e., the solid line) which consists of main roads that are full of traffic. Then, the packet can easily find next hop relay vehicle and reach b faster in spite of the longer length of the path.

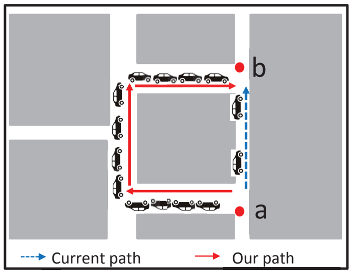


Fig. 2: An example of the traffic-considered shortest path spreading algorithm.

Contact-based scanning algorithm. It restricts each packet copy in its responsible active sub-area to find relay vehicles with high encounter frequencies with the target vehicle. Specifically, the packet copy is forwarded to vehicles traveling in different road sections so that it can evenly scan the sub-area.

By avoiding searching the non-active sub-areas of the target vehicle as in the contact and centrality based routing algorithms, AAR greatly improve routing efficiency. Instead of pursuing the target vehicle as in the location based routing algorithms, each packet copy in an active sub-area of the target vehicle is relayed by vehicles with high encounter frequency with the target vehicle, thus bypassing the insufficiently accurate location prediction problem in location based routing algorithms.

To sum up, the main contributions of this paper are as follows:

1) We measure two real VNET *Roma* and *SanF* traces, which serves as the foundation for our proposed routing algorithm for VNETs.
2) We propose a traffic-considered shortest path spreading algorithm to spread different copies of a packet to different active sub-areas of the target vehicle efficiently.
3) We propose a contact based scanning algorithm in each active sub-area of the target vehicle to relay the packet to the target vehicle.

The rest of this paper is organized as follows. Section II presents the related work. Section IV measures and analyzes the pattern of vehicles' trajectories and the distance among different encounter locations of pairs of vehicles in two real VNET traces. Section V introduces the detailed design of AAR. In Section VI, the performance of AAR is evaluated by trace-driven experiments in comparison with the state-of-the-art routing algorithms. Section VII summarizes the paper with remarks on our future work.

II. RELATED WORK

Current routing algorithms in VDTNs can be classified to three categories: contact based [1–4], centrality based [14, 5–7] and location based routing algorithms [9, 11, 8, 10].

In the category of contact based routing algorithms, PROPHET [1] simply selects vehicles with higher encounter frequency with target vehicles for relaying packets. PROPHET is improved by MaxProp [2] with the consideration of the successful deliveries history. R3 [3]considers not only the encounter frequency history, but also the history of delays among encounters to decrease the routing delay performance. Zhu *et al.* [4] found that two consecutive encounter opportunities drops exponentially and based on the observation, improved the prediction of encounter opportunity by Markov chain to design the routing algorithm in vehicle networks.

In the category of centrality based routing algorithms, PeopleRank [5] is inspired by the PageRank algorithm, which calculates the rank of vehicles and forwards packets to the vehicles with higher ranks. SimBet [6] identifies some bridge nodes as relay nodes which can better connect the VNETs by centrality characteristics to relay packets. Instead of directly forwarding packets to target nodes, Bubble [7] clusters the nodes to different communities based on encounter history and still utilizes the bridge nodes to forward packets to the destination community. However, though vehicles with high centrality can encounter more vehicles, they may not have a high probability of encountering the target vehicle. Also, the main problem in both contact and centrality based routing algorithm is that packets may hardly encounter suitable relay vehicle due to the low encounter frequencies among vehicles in a large-scale VDTN.

In the category of location based routing algorithms, GeOpps [8] directly obtains the future location of the target vehicle from GPS data and spreads packets to certain geographical locations for routing opportunities through shortest paths. GeoDTN [9] encodes historical geographical movement information in a vector to predict the possibility that two vehicles become neighbors. Wu *et al.* [10] exploited the correlation between location and time in vehicle mobility when they used trajectory history to predict the future location of the target vehicle in order to improve the prediction accuracy. Instead of predicting exact future location, DTN-FLOW [11] divides the map to different areas and predict the future visiting area of vehicles, which improves the routing performance since it is much easier to predict the future visiting areas than exact future locations. However, as indicated previously, the location based algorithms may lead to low routing efficiency due to insufficiently accurate location prediction due to traffic and vehicle speed variance.

A number of multi-copy routing algorithms have been proposed. Spyropoulos *et al.* [15] introduced a "spray" family of routing schemes that directly replicate a few copies by source vehicle into the network and forward each copy independently toward the target vehicle. R3 [3] simply adopts the “spray” routing schemes based on its own single-copy routing. Bian *et al.* [16] proposed a scheme for controlling the number of copies per packet by adding an encounter counter for each packet carrier. If the counter reaches the threshold, then the packet will be discarded by the packet carrier. Uddin *et al.* [17] minimized the energy efficiency by studying how to control the number of copies in a disaster-response applications, where energy is a vital resource. However, in current multi-copy routing algorithms, different copies of each packet may search the same area on the entire VDTN area, which decreases routing efficiency. AAR spreads different copies of each packet to different active sub-areas of the target vehicle.

III. IDENTIFICATION OF EACH VEHICLE’S ACTIVE SUB-AREAS

Current routing algorithms search the target vehicle in the entire VDTN map, which leads to low routing efficiency since some routing paths may be outside of the active sub-areas of the target vehicle. Using multi-copies to search in different active sub-areas of the target vehicle can improve routing efficiency. Because our trace measurement uses the concept of each vehicle’s active sub-areas, we first introduce our method of identification of each vehicle’s active sub-areas in this section before we introduce our trace measurement. In the entire VDTN map, a *road section* is the road part that does not contain any intersections and it is denoted by the two IDs of intersections on its two ends such as ab in Figure 2. Then, instead of searching target vehicle v on the entire VDTN map, we only direct packets to search in the road sections where the target vehicle v visits frequently. We call these road sections the *active road sections* of vehicle v. To be more specific, we define the set of active road sections of vehicle v (denoted by S_v) by:

$$S_v = \{\forall s \in S | f(s, v) > r\} \tag{1}$$

where S is the set of all road sections, $f(s, v)$ is the frequency that vehicle v visits road section s; and r is a visit frequency threshold. A smaller threshold r leads to more road sections in the S_v and a larger routing area and vice versa. In this paper, we set $r = 7$ and $r = 5$ in *Roma* and *SanF* traces, respectively.

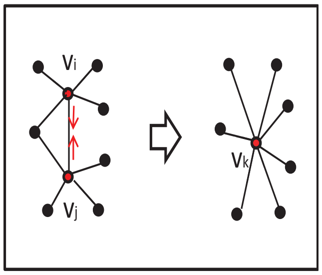


Fig. 3: Active sub-area identification.

Sending a packet copy to each active road section of the target vehicle generates many packet copies and high overhead. Actually, sending a packet copy to a set of connected active road sections is sufficient because the copy can be forwarded to vehicles travelling along all these road sections to search the target vehicle. We define an active sub-area of a vehicle as a set of connected active road sections of the vehicle. We propose a method to create the active sub-areas of each vehicle by following rules.

(1) Each sub-area of a vehicle consists of connected road sections of the vehicle so that a packet copy can scan the entire sub-area for the target vehicle without the need of traveling on the inactive road sections.
(2) Each sub-area of a vehicle should have similar number of road sections so that the load balance on the size of scanning sub-areas of multiple copies can be guaranteed.

Specifically, our active sub-area identification algorithm works as follows:

(1) First, we transform the entire VDTN map to a graph. We consider each road section as a node and connect two nodes if the corresponding two active road sections share the same road intersection. Also, we tag each node with weight 1. Then, the areas division problem is translated to a graph partition problem.
(2) Next, as shown in Figure 3, we continually select a directly connected nodes with the smallest sum of weights, remove the edges between these two nodes, merge them to one node, and set its weight to the sum. If all pairs of directly connected nodes have equal sum of weights, we randomly select a node pair to merge.
(3) We repeat step (2) until the number of nodes equals the number of active sub-areas required. Then, the corresponding road sections in one node constitute an active sub-area.

In the above process, two disconnected nodes cannot be merged, which guarantees that the road sections in each active sub-area are connected. Also, since the weight of a node represents the number of road sections corresponding to the node, merging two nodes with the smallest sum of weights can constrain the difference between the number of road sections in different active sub-areas. As a result, the above two rules are followed, which facilitates the execution of our proposed routing algorithm. The active sub-areas of each vehicle and the entire VDTN map are stored in each vehicle in VDTN. When a vehicle joins in the VDTN, it receives this information.

IV. TRACE MEASUREMENT

In order to design a new routing algorithm to improve the performance of current routing algorithms, we first need to better understand the pattern of vehicles’ trajectories and the relationship between vehicle contact and location. Therefore, we analyze the real-world VNET *Roma* and *SanF* traces gathered by taxi GPS in different cities, referred to as *Roma* [12] and *SanF* [13]. The *Roma* trace contains mobility trajectories of 320 taxies in the center of Roma from Feb. 1 to Mar. 2, 2014. The *SanF* trace contains mobility trajectories of approximately 500 taxies collected over 30 days in San Francisco Bay Area. Our analysis focuses on following two aspects:

1) *Vehicle mobility pattern.* We expect to find out whether the movement of each vehicle exhibits a certain pattern. If each vehicle frequently visits a few sub-areas in the

entire VDTN area, then our routing algorithm only needs to search these sub-areas of a target vehicle in order to improve routing efficiency.

2) *Relationship between contact and location.* Contact and centrality based routing algorithms search vehicles that have high encounter frequency with the target vehicle in the entire VDTN area. If we can identify the locations that the vehicles frequently meet the target vehicle, we can reduce the relay search area to improve the routing efficiency. Therefore, we expect to find out if such locations can be identified.

A. Vehicle Mobility Pattern

In order to measure the pattern of vehicles' mobility on the entire VDTN area, we normalize the total driving time of each vehicle to 100 hours and normalize its real visiting time on each road section by:

$$\bar{t}(s_i, v_i) = \frac{100 \times t(s_i, v_i)}{t_{v_i}} \tag{2}$$

where s_i denotes road section s_i, v_i denotes vehicle i, $\bar{t}(s_i, v_i)$ is the normalized visiting time of vehicle v_i on road section s_i, t_{v_i} is the real total driving time of vehicle v_i and $t(s_i, v_i)$ is the real visiting time of vehicle v_i on road section s_i. We then calculate the deviation of visiting time of vehicle v_i (D_{v_i}) by:

$$D_{v_i} = \frac{1}{|S|} \sum_{i \in S} \left(\bar{t}(s_i, v_i) - \bar{t}_{v_i}\right)^2 \tag{3}$$

where S is the set of all the road sections in the entire VDTN map and $\bar{t}_{v_i}$ is the average visiting time of vehicle v_i per road section. Since the total visiting time of each vehicle is normalized to 100 hours and S is fixed, $\bar{t}_{v_i}$ is a fixed value $\frac{100}{|S|}$ for any vehicle v_i. Figure 4 shows the distributions of the deviation of visiting time of vehicles in the *Roma* and *SanF* traces. The high deviations of most vehicles indicate that these vehicles' trajectories are unevenly distributed among road sections, and they frequently visit a few road sections.

Next, we measure the percentage of time of vehicles spent on their active sub-areas. Figure 5 shows the distribution of the percentage of time of vehicles spent on active area. As we can see, most vehicles spent more than 90% of time on their active sub-areas. Also, as shown in Figure 6, Our measurement shows that usually the total size of active sub-areas of each vehicle is smaller than 10% of the size of the entire VDTN area. From Figures 4 and 5, we conclude our first observation (**O1**) as follows:

O1: *Each vehicle has its own active sub-areas which are usually very small comparing to the entire VDTN map.*

Based on this observation, we can constrain the areas of searching the target vehicle to its active sub-areas. Then, routing of packets on inactive areas of the target vehicle can be avoided and the routing efficiency can be improved.

B. Relationship between Contact and Location

First, we define a pair of vehicles as frequently encountered pair of vehicles if they encounter more than 10 times. Then, for any frequently encountered pair of vehicles v_i and v_j that frequently meet each other, we calculate the average distance $\overline{d(v_i, v_j)}$ by:

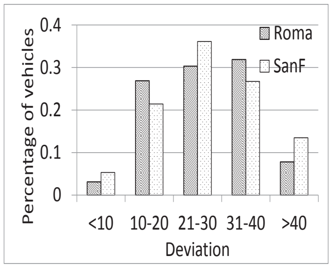


Fig. 4: Deviation of visiting time of vehicles.

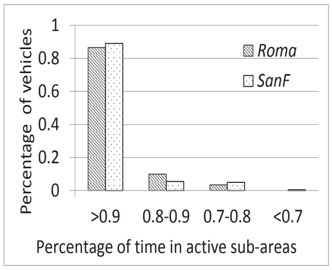


Fig. 5: Percentage of time spent on active sub-areas.

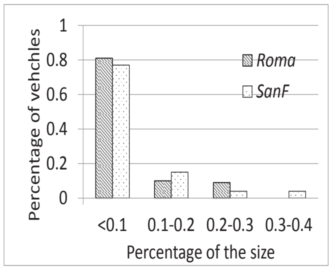


Fig. 6: Percentage of size of the entire map.

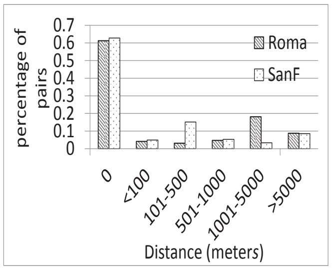


Fig. 7: Distance from encounter locations to active sub-areas.

$$\overline{d(v_i, v_j)} = \frac{\sum_{i=1}^{n} d_i(v_i, v_j)}{|n|} \tag{4}$$

where $d_i(v_i, v_j)$ is the shortest distance between the ith encounter location and the shared active sub-areas of vehicles v_i and v_j, and n is the number of encounters happened between these two vehicles. Figure 7 shows the distribution of the average distances of pairs of vehicles that encountered frequently in the *Roma* and *SanF* traces. We find that for most pairs of vehicles, their encounter locations are near by their active sub-areas. Actually, most encounters are happened in their active sub-areas (i.e., encounter locations with average distance 0). Therefore, we conclude our second observation (**O2**) as follows:

O2: *The frequently encountered vehicles usually encounter each other in their active sub-areas.*

Based on this observation, we can use contact based routing in each active sub-area of the target vehicle rather than the entire VDTN map, which will greatly improve the routing efficiency.

V. ACTIVE AREA BASED ROUTING METHOD

Before introducing the detailed design of AAR, we first give an overview of the routing process for a packet in AAR.

1) In the traffic-considered shortest path spreading algorithm, the source vehicle spreads different copies of the packet to the target vehicles' active sub-areas through paths with short distance and more traffic, as shown in the left part of Figure 8. Different from current location

based routing algorithms, we identify the spreading paths with the consideration of not only the physical distance of the paths but also the traffic condition in order to have enough relay vehicle candidates in spreading, which improves the spreading efficiency.

2) In the contact-based scanning algorithm, a packet copy in an active sub-area continually scans the sub-area until it encounters the target vehicle or a vehicle that can encounter the target vehicle more frequently as a relay vehicle, as shown in the right part of Figure 8. Since the scanning focuses on the active sub-areas of the target vehicle that it visits frequently and also encounters its frequently encountered vehicles, the routing efficiency is improved.

In the following, we introduce these two algorithms. As the work in [18], we assume that each road intersection is installed with a road unit. The road unit can send information to and receive information from nearby vehicles and store information. The road units help to calculate traffic, receive and forward packets.

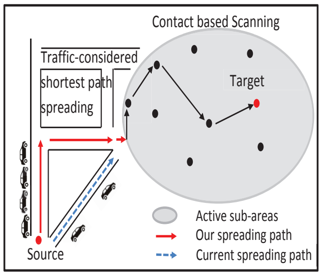


Fig. 8: An example of the routing process.

A. *Traffic-considered Shortest Path Spreading*

To spread the copies of a packet to different active sub-areas of the target vehicle, as we indicated previously, if we directly calculate the shortest path only based on the distance, the identified path may have few vehicles to function as relays, which leads to low routing efficiency. Therefore, our traffic-considered shortest path spreading algorithm jointly considers distance and traffic in selecting a spreading path. To spread the multiple packet copies, the source vehicle can send a copy to each sub-area individually, which however generates high overhead. To handle this problem, our spreading algorithm builds the spreading path tree that combines common paths of different copies in copy spreading. For example, Figure 10 shows an example of such a spreading path tree to spread packet copies to active sub-areas as shown in Figure 10, where letters $a - i$ represents the road units. Then, the copies of a packet are relayed to their responsible active sub-areas one road unit by one road unit through vehicles. Each source vehicle needs the traffic information in determining the spreading path. Below, we first introduce how the traffic is calculated and dynamically updated in each vehicle in Section V-A1. We then introduce the details of our spreading algorithm in Section V-B.

1) Road Traffic Measurement: Road traffic varies from time to time. Therefore, it is necessary to measure the road traffic dynamically. Each the road unit in each intersection measures the traffic of each road section as follows:

(1) When a vehicle v passes intersection a, vehicle v sends ID of the previous road unit it passed to road unit in intersection a (denoted by u_a).

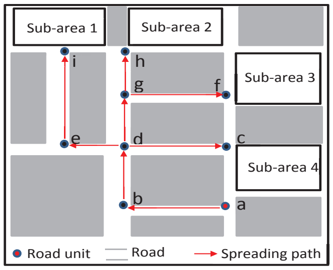


Fig. 9: The shortest path to different sub-areas.

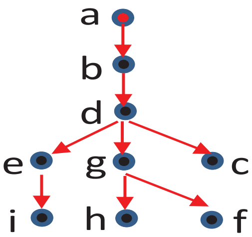


Fig. 10: An example of spreading path tree.

(2) Road unit u_a periodically updates the traffic in road section ab by:

$$T_{ba}^{t} = \alpha T_{ba}^{t-1} + (1-\alpha) N_{ba}^{t} \tag{5}$$

where T_{ba}^{t} is the traffic from intersection b to a at time t and N_{ba}^{t} is the number of vehicles that have pass through intersection b to a during the time period.

Also, each vehicle updates its road traffic information via two ways as follows:

1) When a vehicle v_a passes road unit u_a, road unit u_a sends its stored traffic information to vehicle v.
2) When vehicles v_a and v_b encounter each other, they exchange their stored traffic information. Then, the vehicles compare and update the traffic information of each road section with updated information.

2) Building Traffic-considered Shortest Path Tree: Based on the traffic information, we introduce our algorithm for each vehicle to build the traffic-considered shortest path tree to spread packet copies to different active sub-areas.

(1) First, we introduce a metric called *traffic-considered distance* that jointly considers the distance and traffic of a road section:

$$D_{ba} = \frac{d_{ba}}{T_{ba}} \tag{6}$$

where T_{ba} is the updated traffic from intersection b to a, d_{ba} is the physical distance length between b and a and D_{ba} is traffic-considered distance from b to a. It is not necessary that $D_{ba} = D_{ab}$. Using this metric in selecting spreading path, we can find path with shorter distance and higher traffic (i.e., more relay candidates), which can improve the routing efficiency.

(2) Recall that an active sub-area of a target vehicle consists of several connected road sections. In order to successfully sends a packet copy to a sub-area, a source vehicle must send the copy to an intersection of one road section in the sub-area. To find the path with minimum traffic-considered distance for a sub-area, the source vehicle first builds a graph, in which the nodes are the intersection it will pass and all the intersections of the road sections in the sub-area, two nodes are connected if their corresponding intersections are connected by a road section, and the weight of each edge equals the

Road section	Time stamp	Road section	Time stamp
ac	1:01	fg	1:14
bd	1:12	fh	1:03
cd	0:00	gi	1:09
cf	1:02	hi	1:06
dg	1:13	hj	1:05
ef	1:16	ik	1:08

Fig. 11: An example of the scanning history table.

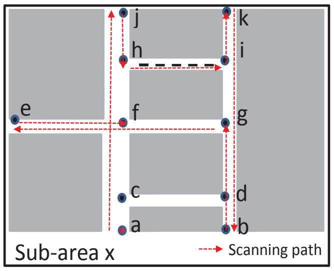


Fig. 12: An example of the scanning road section selection.

traffic-considered distance. It then finds the shortest path to each road intersection using the Dijkstra algorithm. Among these paths, it further picks up the shortest path as the shortest path to the sub-area.

(3) After the source vehicle calculates the shortest paths to all the active sub-areas of the target vehicle, it combines the common paths in these shortest paths to build the spreading path tree. For example, paths $a \rightarrow b \rightarrow d \rightarrow c$, $a \rightarrow b \rightarrow d \rightarrow e$, $a \rightarrow b \rightarrow d \rightarrow g \rightarrow h$, $a \rightarrow b \rightarrow d \rightarrow g \rightarrow f$ and $a \rightarrow b \rightarrow d \rightarrow e \rightarrow i$ are combined to a tree by merging the same road intersections on different paths (as shown in Figure 9 and Figure 10), so that copies can be spread efficiently.

When the source vehicle arrives at the next road unit, i.e., u_a, it drops the packet to u_a. When a vehicle travelling to road unit u_b passes u_a, u_a sends a packet copy to the vehicle, which will drop the packet to u_b. u_b will send a copy to u_d through a vehicle travelling to u_d. Then, u_d sends packet copies to three vehicles travelling to u_e, u_g and u_c respectively. This process repeats until all road units in the spreading path tree receive a packet copy.

B. Contact-based Scanning in Each Active Sub-area

After a packet copy arrives at an active sub-area of the target vehicle, the packet carriers (i.e., road units and vehicles) use the contact-based scanning algorithm in the sub-area to forward the packet to the target vehicle. As in current contact based routing algorithms, each vehicle records its contact frequency to others and exchange such information upon entering. Therefore, a packet carrier can judge if its encountered vehicle is a better packet carrier, i.e., has a higher encounter frequency with the target vehicle. In our contact-based scanning algorithm, the packet is being forwarded to vehicles travelling in different road sections in order to evenly scan the sub-area to meet the target vehicle. During the scanning process, if the packet carrier meets a vehicle which is a better packet carrier or can lead to more even scanning, it forwards the packet to this entered vehicle. Once a packet carrier is about to leave the active sub-area, it drops the copy to the boundary road unit of the sub-area, which will forward the packet to the vehicle whose traveling direction is the road section that should be scanned.

1) Maintaining Scanning History Table: In order to ensure that the entire active sub-area can be scanned by a packet, each packet maintains a scanning history table. The scanning history table records the scanning history of the packet. For example, as shown in Figure 11, each road section in the sub-area has a time stamp which is its last scanning time. A road unit chooses the road section that has the oldest time stamp among the reachable road sections as the next scanning road section. For example, as shown in Figure 12, from intersection c, the road sections that can be scanned are road sections ac, cd and cf, while road section ef cannot be scanned. Since road section cd has the oldest time stamp, it is the next scanning road section. Once a packet finishes scanning a road section, the time stamp of this road section in its scanning history table is updated with the current time.

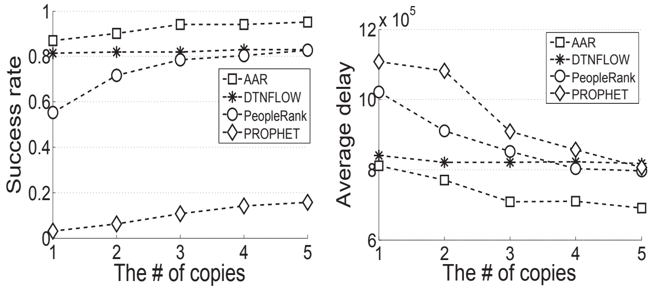
2) Routing Algorithm in a Sub-area: We adopt the method in [1] to measure the encounter frequency of each pair of vehicles. Specifically, the contact utility is calculated every time when once two vehicles encounter by:

$$C(v_i, v_j) = C_{old}(v_i, v_j) + (1 - C_{old}(v_i, v_j)) \times C_{init}(v_i, v_j) \tag{7}$$

where $C(v_i, v_j)$ is the updated encounter frequency utility; $C_{old}(v_i, v_j)$ is the old encounter frequency utility and $C_{init}(v_i, v_j)$ is the initial value of contact utility of all the pairs of vehicles, which is set to a value selected from $(0, 1)$. This definition ensures that the two vehicles with a high encounter frequency have a larger encounter frequency utility.

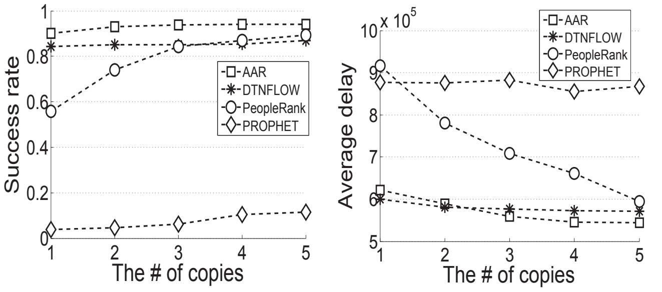
Below, we explain the contact-based scanning algorithm. Recall that the traffic-considered shortest path algorithm sends a packet copy to a road unit in an active sub-area of the target vehicle. Then, the contact-based scanning algorithm is executed. First, the road unit determines the road section that the packet should scan, which is the road section that has the oldest scan time stamp among the reachable road sections, as explained previously. Then, the road unit will forward the packet to the passing vehicle, say v_i, with the direction to the selected road section. When v_i travels along the road section, for each of its encountered vehicle v_j, if v_j has a higher contact utility to the target vehicle or v_j's direction has a smaller time stamp then v_i's direction in the scanning history table of the packet among all the reached road sections, v_i forwards the packet to v_j. After a vehicle finishes scanning a road section, it drops the packet to the road unit on the intersection in the end of this road section. If a vehicle is leaving the active sub-area, it also drops the packet to the boundary road unit. Then, the road unit decides the next scanning road section and the process repeats until the packet meets the target vehicle.

As shown in Section IV, the target vehicle spends most of travelling time in its active sub-areas and also it meets its frequently encountered vehicles its active sub-areas. Therefore, by scanning the target vehicle's active sub-areas and relying on vehicles with high contact utilities with the target vehicle in routing can greatly improve routing efficiency and success rate.



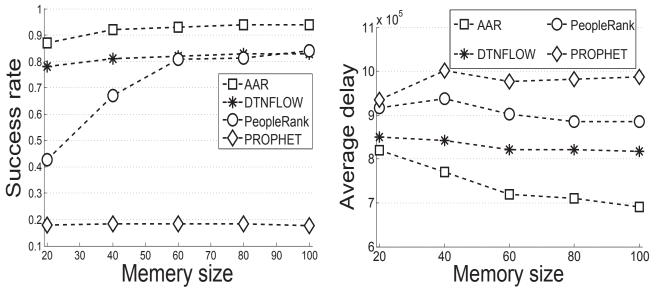
(a) Success rate (b) Average delay

Fig. 13: The performance with different number of copies on *Roma* trace.



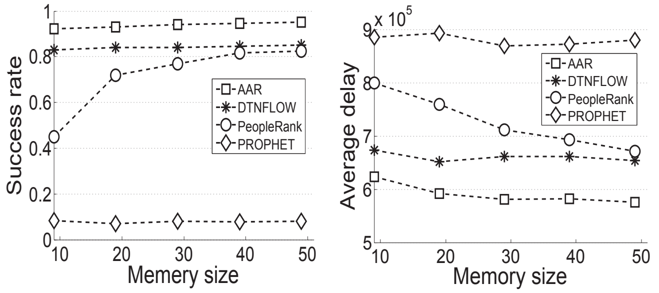
(a) Success rate (b) Average delay

Fig. 14: The performance with different number of copies on *SanF* trace.



(a) Success rate (b) Average delay

Fig. 15: The performance with different memory sizes on *Roma* trace.



(a) Success rate (b) Average delay

Fig. 16: The performance with different memory sizes on *SanF* trace.

VI. PERFORMANCE EVALUATION

In order to evaluate the performance of AAR, we conduct the trace-driven experiments on both the *Roma* and *SanF* traces in comparison with DTN-FLOW [11], PeopleRank [5] and PROPHET [1] algorithms. DTN-FLOW represents location based routing algorithms, PeopleRank represents centrality based routing algorithms, and PROPHET represents contact based routing algorithms. The details of the algorithms are introduced in Section II. We measured the success rate that packets arrive at their target vehicles and the average delay for successfully delivered packets to their target vehicles. In our experiments, the number of active sub-areas of the target vehicle depends on the number of multiple copies of the packet. To be more specific, we spread one copy of a packet to each active sub-area and therefore, the number of active sub-areas equals the number of multiple copies.

A. Performance with Different Number of Copies

Since our algorithm is designed for multi-copy routing, we compare AAR with the other three algorithms with multiple copies of each packet replicated by the spray and wait multi-copy routing algorithm [15] for fair comparisons. Figure 13(a) and Figure 14(a) show the success rates with different numbers of copies per packet in the *Roma* and *SanF* traces , respectively. Generally, the success rate follows AAR>DTN-FLOW>PeopleRank>PROPHET. The performance of DTN-FLOW is better than PeopleRank since DTN-FLOW divides the very large area to sub-areas and avoid to search the target vehicles on a very large area. AAR performs better than DTN-FLOW since AAR considers the encounter history. PROPHET performs the worst, since it is difficult to encounter a vehicle that encounters the target vehicle frequently in the very large area.

Figure 13(b) and Figure 14(b) show the average delays with different numbers of copies per packet. Generally, the average delays follow PROPHET>PeopleRank>DTNFLOW>AAR. The delay of PROPHET is the largest, since the copies of a packet waste most time outside of active sub-areas where target vehicle barely visits brought by relay vehicles, as shown in the left part of Figure 1. The delay of DTN-FLOW is smaller than PROPHET since DTN-FLOW limits the routing paths in certain sub-areas. The delay of AAR is the smallest, since AAR not only spreads each copy to its responsible active sub-area efficiently by traffic-considered paths, but also scans different active sub-areas with the help of vehicles that encounter target vehicles frequently simultaneously.

B. Performance with Different Memory Sizes

Besides the number of copies per packet, the memory size of each vehicle also influences the performance. Therefore, we analyze the influence of memory size to different algorithms. Figure 15 and Figure 16 shows the success rates and average delays with different memory sizes, where we suppose that 1 unit memory (horizontal axis) can save 1 packet. Generally, the sensitivities of different algorithms to the memory sizes follow PeopleRank>AAR>DTNFLOW>PROPHET. The performance of PeopleRank is very sensitive to the memory size, since all the packets tend to be forwarded to few vehicles with very high PeopleRank values and the limited memory size can significantly influence the routing process negatively. PROPHET is insensitive to the memory size, since the packets only tend to find those specific vehicles with high probability to encounter the target vehicles, which guarantees load balance. However, PROPHET generates low success rate and long delay due to the reasons we mentioned in Section VI-A. DTNFLOW is also not sensitive to the memory size since each packet is relayed in limited times from one landmark to another landmark. The performance success rate and average delay of AAR is slightly improved with the increasing memory size since a larger memory size allows packets to scan sub-areas more frequently.

To sum up, AAR has the highest success rate and the lowest average delay. However, AAR is a little sensitive to the number of copies and the memory size. DTNFLOW and PeopleRank have the medium success rate and average delay. However, PeopleRank is very sensitive to the number of copies and the memory size. DTNFLOW and PROPHET is not sensitive to the number of copies and the memory size. However, PROPHET has very low success rate and high average delay. To sum up, considering memory size and limited number of copies are not a main concern in VDTN routing, AAR performs best in the four routing algorithms.

VII. CONCLUSION

In this paper, we first measured the pattern of vehicles mobility and the relationship between contact and location for each pair of vehicles. Then, by taking advantage of the observations, we proposed Active Area based Routing method (AAR). Instead of pursuing the target vehicle on the entire VDTN area, AAR spreads copies of a packet to the active sub-areas of the target vehicle where it visits frequently and restricts each copy in its responsible sub-area to search the target vehicle based on contact frequency. The trace-driven simulation demonstrates that AAR has a highest success rate and lowest average delay in comparison with other algorithms. In our future work, we will discuss the possibility of routing in VDTNs without the help of road units.

ACKNOWLEDGEMENTS

This research was supported in part by U.S. NSF grants NSF-1404981, IIS-1354123, CNS-1254006, CNS-1249603, Microsoft Research Faculty Fellowship 8300751.

REFERENCES

[1] A. Lindgren, A. Doria, and O. Scheln, “Probabilistic routing in intermittently connected networks.,” *Mobile Computing and Communications Review*, vol. 7, pp. 19–20, 2003.

[2] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, “Maxprop: Routing for vehicle-based disruption-tolerant networks.,” in *Proc. of INFOCOM*, IEEE, 2006.

[3] A. Symington and N. Trigoni, “Encounter based sensor tracking.,” in *Proc. of MobiHoc*, pp. 15–24, ACM, 2012.

[4] H. Zhu, S. Chang, M. Li, K. Naik, and S. X. Shen, “Exploiting temporal dependency for opportunistic forwarding in urban vehicular networks.,” in *Proc. of INFOCOM*, IEEE, 2011.

[5] A. Mtibaa, M. May, C. Diot, and M. H. Ammar, “Peoplerank: Social opportunistic forwarding.,” in *Proc. of INFOCOM*, pp. 111–115, IEEE, 2010.

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

[7] P. Hui, J. Crowcroft, and E. Yoneki, “Bubble rap: Social-based forwarding in delay-tolerant networks.,” *IEEE Trans. Mob. Comput.*, vol. 10, no. 11, pp. 1576–1589, 2011.

[8] I. Leontiadis and C. Mascolo, “Geopps: Geographical opportunistic routing for vehicular networks.,” in *Proc. of WOWMOM*, pp. 1–6, IEEE, 2007.

[9] J. g. B. Link, D. Schmitz, and K. Wehrle, “Geodtn: Geographic routing in disruption tolerant networks.,” in *Proc. of GLOBECOM*, pp. 1–5, IEEE, 2011.

[10] Y. Wu, Y. Zhu, and B. L. 0001, “Trajectory improves data delivery in vehicular networks.,” in *Proc. of Infocom*, pp. 2183–2191, IEEE, 2011.

[11] K. Chen and H. Shen, “Dtn-flow: Inter-landmark data flow for high-throughput routing in dtns.,” in *Proc. of IPDPS*, pp. 726–737, IEEE, 2013.

[12] B. Lorenzo, B. Marco, L. Pierpaolo, B. Giuseppe, A. Raul, and R. Antonello, “CRAWDAD data set roma/taxi (v. 2014-07-17).” Downloaded from http://crawdad.org/roma/taxi/, July 2014.

[13] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, “CRAWDAD data set epfl/mobility (v. 2009-02-24).” Downloaded from http://crawdad.org/epfl/mobility/, Feb. 2009.

[14] T. Hossmann, T. Spyropoulos, and F. Legendre, “Know Thy Neighbor: Towards Optimal Mapping of Contacts to Social Graphs for DTN Routing.,” in *Proc. of INFOCOM*, pp. 866–874, IEEE, 2010.

[15] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Efficient routing in intermittently connected mobile networks: the multiple-copy case.,” *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 77–90, 2008.

[16] B. Hong and Y. Haizheng, “An Efficient Control Method of Multi-copy Routing in DTN.,” in *Proc. of NSWCTC*, pp. 153–156, IEEE, 2010.

[17] M. Y. S. Uddin, H. Ahmadi, T. F. Abdelzaher, and R. Kravets, “A low-energy, multi-copy inter-contact routing protocol for disaster response networks.,” in *Proc. of SECON*, pp. 1–9, IEEE, 2009.

[18] C. Song, M. Liu, Y. Wen, J. Cao, and G. Chen, “Buffer and switch: An efficient road-to-road routing scheme for vanets.,” in *Proc. of MSN*, pp. 310–317, IEEE, 2011.