Hybrid Position-based and DTN Forwarding in Vehicular Ad Hoc Networks

Lei Zhao* Fan Li* Yu Wang†
* School of Computer Science, Beijing Institute of Technology, Beijing, 100081, China.
† Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC 28223, USA.

***Abstract*—Efficient data delivery in *vehicular ad hoc networks* (VANETs) is still a challenging research issue. Position-based routing protocols have been proven to be more suitable for dynamic VANETs than traditional ad hoc routing protocols. However, position-based routing assumes that intermediate nodes can always be found to setup an end-to-end connection between the source and the destination, otherwise, it suffers from network partitions which are very common in VANETs and leads to poor performances. This paper addresses data delivery challenge in the possible disconnected VANETs by combining position-based forwarding strategy with store-carry-forward routing scheme from delay tolerant networks. The proposed routing method makes use of vehicle driving direction to determine whether holding or forwarding the packet. Experimental results show that the proposed mechanism outperforms existing position-based solutions in terms of packet delivery ratio.**

***Index Terms*—Hybrid routing, position-based routing, DTN routing, vehicular ad hoc networks.**

I. Introduction

Vehicular ad hoc network (VANET) is a high-speed mobile wireless network containing a set of smart vehicles which could communicate with each other via wireless medium. VANETs aim to provide ubiquitous and efficient networking connectivity for mobile users on the road and support a variety of safety applications such as co-operative traffic monitoring and prevention of collisions. To realize this objective, the Federal Communications Commission (FCC) has allocated 75 MHz of spectrum for short range Vehicle-to-Vehicle (V2V) or vehicle-to-roadside communications. IEEE also formed the new IEEE 802.11p task group which adds wireless access in vehicular environments. As the increasing popularity of mobile wireless devices and smart vehicles, in the near future, VANETs service will become a necessary part of the general public life.

Different from traditional mobile ad hoc network, VANET has its unique characteristics which pose new challenges in the design of networking protocols, especially for routing protocols. For example, as vehicles move at relevant high speeds, the network topology is highly dynamic and may be even frequently disconnected. Vehicles rely on wireless links to relay data, and communication bandwidth is consequently limited and unstable [1]. Even some of the existing ad hoc routing protocols can still be applied to VANETs, simulation results [2]–[5] have showed that they suffer from poor performances because of the fast movement of vehicles and limited chances for information exchange. Therefore, finding and maintaining stable routes is a challenging task in VANETs.

Considering the highly dynamic nature of node mobility in VANETs, many different VANET routing protocols (traditional topology-based ad hoc routing, cluster-based routing, or position-based routing) have been proposed in recent years [6]. Several studies [4], [5] have shown that position-based routing (such as GFG [7] and GPSR [8]) is more suitable and promising than topology-based routing (such as AODV and DSR) for VANETs. However, position-based method works best in a free open space scenario with evenly distributed nodes but suffers from various obstacles in city conditions. More importantly, position-based routing assumes that intermediate nodes can always be found to setup an end-to-end connection between the source and the destination, which is not true in many VANET environments. On the other hand, in some applications, for example, those to get information about remaining stocks at a department store or the available parking lots at a parking place, the mobile users can tolerate up to seconds or minutes of delay as long as the reply eventually returns. Such applications can be supported by network protocols designed for delay tolerant networks (DTNs) and enable the new research direction in the routing design for VANETs.

In this paper, by leveraging the DTN technology, we study hybrid forwarding schemes for efficient data delivery in VANETs. Specifically, when a vehicle issues a delay tolerant data query to a fixed site, we propose hybrid forwarding techniques to efficiently route the packet by combining position-based forwarding (such as *Greedy Perimeter Stateless Routing*, GPSR [8]) and the idea of *store-carry-forward* from DTNs [9], [10] where the data can be incrementally moved and stored throughout the network. In our proposed method, when the packet enters perimeter mode, according to the driving direction of the vehicle and the delivery direction of the packet, the vehicle will determine to either hold or deliver this packet. Our simulation results show that this proposed method outperforms existing position-based solutions.

Notice that there are a few existing studies which also exploit the combination of position-based routing and DTN routing [11]–[13]. LeBrun *et al.* [13] described several opportunistic forwarding schemes for VANETs where location

The work of F. Li is partially supported by the National Natural Science Foundation of China under Grant 60903151, the Beijing Natural Science Foundation under Grant 4122070 and SRF for ROCS, SEM. The work of Y. Wang is supported in part by the US National Science Foundation under Grant CNS-0915331 and CNS-1050398. F. Li (fli@bit.edu.cn) is the corresponding author.

978-1-4673-1881-5/12/$31.00 ©2012 IEEE

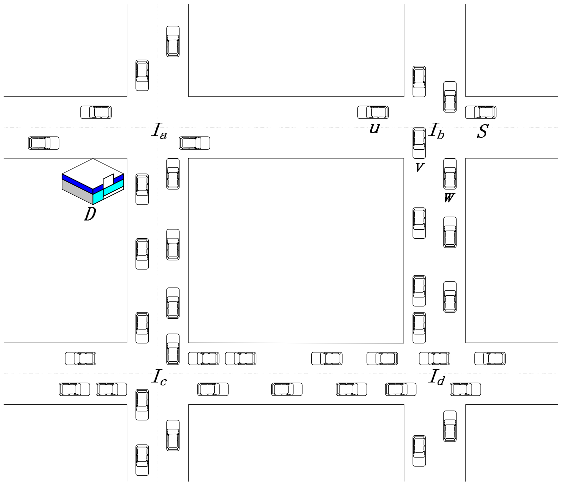


Fig. 1: An example of routing problem in VANETs: vehicle S wants to send a packet to a fixed site D at the parking deck.

or relative velocity of vehicles is used for relay selection. Leontiadis and Mascolo [12] proposed a DTN routing algorithm that exploits the availability of suggested route information from the navigation system in order to opportunistically route a packet to certain geographical location. Cheng *et al.* [11] also proposed a hybrid geographic and DTN routing, GeoDTN+Nav, which combines GPSR with an additional DTN mode. In the perimeter mode, if the network is disconnected, GeoDTN+Nav switches to DTN mode where the packet is stored and carried. However, we take a different approach to consider the traffic, geographical location, and driving direction information to make carrying or forwarding decisions.

The rest of this paper is organized as follows. Section II describes the principle idea of our proposed hybrid forwarding method. Section III presents our simulation results and Section IV concludes the paper.

II. Hybrid Position-based and DTN Forwarding

In this section, we will present the detail of our proposed hybrid position-based and DTN forwarding strategy.

A. Assumptions

Position-based routing protocol is usually used in well-connected networks, but VANETs sometimes are sparse and disconnected networks. In this paper, we assume that vehicles are equipped with an on-broad navigation system loaded with digital maps and a GPS receiver, which provides the location service for the whole region. Mobile vehicles can obtain their location, velocity and direction through the GPS and the unique location information of any fixed site via the navigation system as well. In addition, vehicles can communicate with nearby vehicles through short range wireless channel and learn their location information through periodic beacon messages. We assume that the data packets are generated at mobile vehicles and the destinations are usually fixed sites.

B. Motivation

Let us first consider an example shown in Fig. 1, in which a vehicle S sends a data packet to the fixed destination D at the corner of intersection I_a. According to the GPSR protocol, vehicle u will receive the packet (because u is S's closest neighbor to D) but u cannot find a closer neighbor to D thus the packet enters *perimeter mode* at u. Based on the right-hand rule, u will deliver the packet to v and the packet will be relayed through a path $I_b \rightarrow I_d \rightarrow I_c \rightarrow I_a$ to the destination. In this situation, there are enough vehicles along these three segments so that packet can be delivered to the destination. However, when the segment between intersections I_c and I_d is blocked by either traffic lights or a sudden accident, the route may become disconnected and the packets need to be detoured to a longer route or even be dropped. On the other hand, vehicle u is driving towards the destination, even it currently does not have a good relay node but it can carry the packet and may deliver it to destination by itself within much shorter time than the route of $I_b \rightarrow I_d \rightarrow I_c \rightarrow I_a$. Therefore, it will be interesting to exploit such enhancement to GPSR by considering possible store-carry-forward options.

C. Detailed Hybrid Forwarding Mechanism

Our proposed mechanism is based on a position-based routing, GPSR, and the idea of store-carry-forward from DTN routing. In sparsely connected vehicular networks, when a vehicle finds no better neighbors to be the next hop to relay during geographic forwarding, it can store and carry the packet. The key issues are when and how to store and carry the packet and which next hop node to choose for relaying the packet. The basic idea of our approach is smartly switching between position-based forwarding and store-carry forwarding based on the current traffic situation and locations of neighboring vehicles.

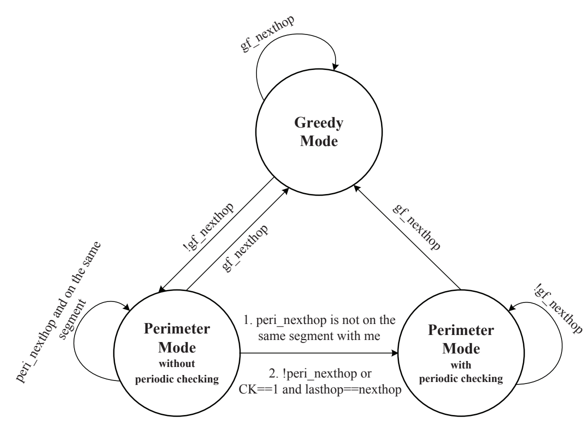


Fig. 2: Transitions among the three possible statuses.

In our design, there are three possible statuses: *greedy mode*, *perimeter mode* without periodic checking, and *perimeter mode* with periodic checking, as shown in Fig. 2. Initially, all packets are in *greedy mode* at the source node.

When a vehicle receives a packet in *greedy mode*, it tries to forward the packet based on greedy forwarding. If it can find

Algorithm 1 Hybrid Forwarding Scheme

```
1: Initialization: mode = greedy_mode; CK = 0
   when receive a packet in greedy_mode:
2: if gf_nexthop is found then
3:    CK = 0
4:    lasthop = my_id; nexthop = gf_nexthop
5:    send the packet p to nexthop
6: else
7:    nexthop = peri_nexthop; mode = perimeter_mode
8:    if CK == 1 and lasthop == nexthop or peri_nexthop is not found then
9:       continue to hold the packet p
10:      enter the Periodic Checking
11:   else
12:      CK = 0; lasthop = my_id
13:      if nexthop is not on the same segment with me then
14:         store a copy of the packet p
15:         enter the Periodic Checking
16:      end if
17:      send the packet p to nexthop
18:   end if
19: end if
   when receive a packet in perimeter_mode:
20: if gf_nexthop is found then
21:    lasthop = my_id; nexthop = gf_nexthop
22:    mode = greedy_mode
23:    send the packet p to nexthop
24: else
25:    nexthop = peri_nexthop
26:    if peri_nexthop is not found then
27:       continue to hold the packet p
28:       enter the Periodic Checking
29:    else
30:       lasthop = my_id
31:       if nexthop is not on the same segment with me then
32:          store a copy of the packet p
33:          enter the Periodic Checking
34:       end if
35:       send the packet p to nexthop
36:    end if
37: end if
```

Algorithm 2 Periodic Checking

```
Initialization: CTTL = MAXCTTL
CK = 1
if CTTL > 0 then
   CTTL − −
   if gf_nexthop is found then
      nexthop = gf_nexthop
      if nexthop's direction is different with mine and nexthop == lasthop and my_angle < nexthop_angle then
         continue to hold the packet
      else
         mode = greedy_mode
         send the packet to nexthop
      end if
   else
      continue to hold the packet
   end if
else
   delete the packet form the buffer
end if
Repeat Periodic Checking at the next checking interval
```

a better next hop ($gf\_nexthop$) based on location, it forwards the packet to $gf\_nexthop$. Otherwise, it enters *perimeter mode* and needs to further determine whether to store a copy and send the packet or just hold the current packet.

While a vehicle receives a packet in *perimeter mode*, if it has a closer next hop ($gf\_nexthop$), the packet switches back to *greedy mode* and is forwarded to $gf\_nexthop$ based on greedy forwarding. Otherwise, the vehicle carries a copy of the packet and sends the packet to the next hop ($peri\_nexthop$) selected by right-hand rule, if it exists. However, not all of the packets in the *perimeter mode* must be carried. For example, in Fig. 1, vehicle v finds its $peri\_nexthop$ (w) on the $I_b \rightarrow I_d$ segment. No matter what direction of $peri\_nexthop$ (w) is, v can just send the packet to w and do not need to carry the packet. Here, we use the segment of $peri\_nexthop$ to determine whether to carry the packet or not. If the current vehicle and its next hop are on the same segment, there is no need to carry the packet.

When the packet is carried by a vehicle (happened in *perimeter mode*), the vehicle needs to check its neighbor list periodically to see whether there is a possible next hop towards the destination. If there is a closer neighbor, it can switch back to greedy forwarding immediately. To save the store space, each packet is held for at most $MAXCTTL$ time units. When the timer $CTTL$ expires, the packet will be discarded.

The detailed forwarding algorithm and periodic checking mechanism are given as Algorithm 1 and Algorithm 2, respectively. Fig. 2 illustrates the possible transitions among three statuses. Here, CK is used to remember whether a packet has been switched from *perimeter mode* with periodic checking to *greedy mode*.

D. Examples

Next, we use examples shown in Fig. 3 (a close look at intersection I_b from Fig. 1) to explain how our algorithm works on a sparse segment when current vehicle v and the next hop vehicle u are on the same or reverse direction.

Same direction case: Fig. 3(a) illustrates the example when vehicles u and v are driving in the same direction towards the destination. When vehicle u receives a packet from v, the packet enters *perimeter mode* and u will send it back to v. When v receives the packet again (but in *perimeter mode*), it finds vehicle w is the next hop and they are not on the same segment. According to Algorithm 1 (Line 31), v stores a copy

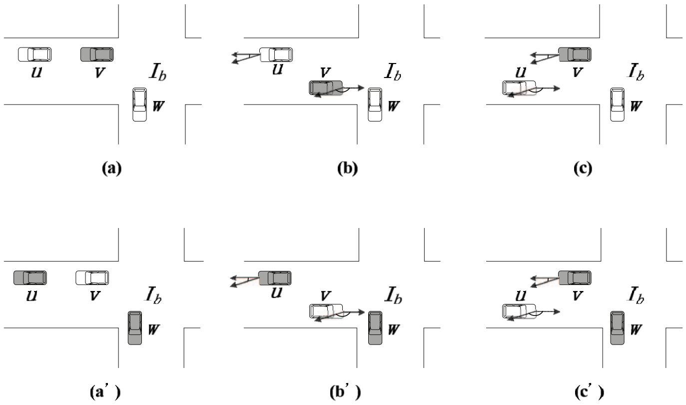


Fig. 3: Examples of three different cases: (a) and (a') u, v on the same direction; (b), (b'), (c) and (c') u, v on reverse directions. Here the shaded vehicles hold a copy of the packet.

of the packet and forwards the packet to w. Then the copy of the packet at v enters the periodic checking and marks the CK bit to 1. This means that the packet has experienced the periodic checking. v finds u is its best neighbor (Algorithm 2: Line 5) and they are driving in the same direction (Algorithm 2: Line 10), so v sends the copy to u again. When u receives the packet, it should hold this packet instead of sending it back (otherwise causing a routing loop). Since the CK bit is 1 and $nexthop = lasthop$ (Algorithm 1: Line 8), u will only hold the packet. Therefore, there will be two copies of the original packet (w and u) as shown in Fig. 3(a') which can explore both possible routes (DTN one and perimeter one).

Reverse direction case: In Fig. 3(b) and (c), vehicles u and v are driving in opposite direction. Our proposed scheme will make use of the angle (i.e. my_angle), which is constituted by the vehicular driving direction and the destination's direction (as shown in Fig. 3(b) and (c)), to determine whether holding or forwarding the packet. Recall that in these examples the destination (i.e., the parking deck) is in the left-bottom corner of these figures. Like the previous example, when the packet reaches v again, v stores a copy and sends packet to w, then enters periodic checking. v will find u is its next hop which is closer to destination. As shown in Fig. 3(b), the relationship between u and v ($my\_angle > nexthop\_angle$) does not satisfy Line 7 of Algorithm 2. Thus, v will send the packet to u again. The final result is the same with the last example, as shown in Fig.3(b'). On the other hand, if u and v are driving as Fig. 3(c) shows, since $nexthop = lasthop = u$ and $my\_angle < nexthop\_angle$, which satisfies Line 7 of Algorithm 2, v will continue holding the packet and driving towards the destination. Result is shown in Fig.3(c'). In both cases, the routing loop is prevented.

III. Performance Evaluations

We evaluate the performance of our proposed protocol (GPSR-DTN) and compare it with GPSR and its variant via simulations conducted in NS-2 [14]. Since GPSR is not proposed for sparsely connected networks, to be fair, we extend GPSR by allowing that each node has buffers to hold packet. In this way, GPSR-B (GPSR with buffer) can be considered as combining a simple carry and forward protocol with GPSR but without loop-free prevention.

A. Simulation Environment

We use MOVE (MObility model generator for VEhicular networks) [15] to generate realistic mobility model for VANETs. The street is designed in both directions and traffic lights are deployed at each intersection. The distance between two adjacent traffic lights can have a significant effect on the network connectivity. Specifically, the network can be "fragmented" by the traffic lights when the radio transmission range is smaller than the distance between two adjacent clusters. In order to evaluate the proposed methods, we test it in two types of network settings: *almost connected* and *intermittently connected*. In the intermittently connected one, the network is interrupted periodically because of the traffic lights. All networks are deployed in a $2052m \times 2052m$ square map. 100 vehicles are deployed to the street layout. Vehicles move between $0-20m/s$ along the street. The communication range is set to $250m$, and the period of beacon message is 1 second. Ten vehicles are selected as data sources and keep sending data with different intervals from 0.5 to 5 seconds. The destination of all data packets is a static node.

In all experiments, we compare GPSR, GPSR-B (with buffer) and GPSR-DTN with the following routing metrics.

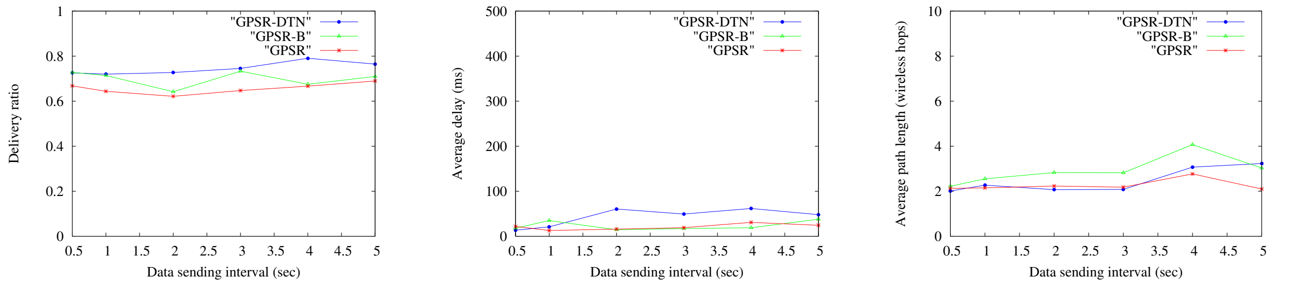
- **Delivery ratio:** the average percentage of successfully delivered packets from the sources to the destination.
- **Average delay:** the average time duration of successfully delivered packets from the sources to the destination.
- **Average path length:** the average number of intermediate vehicles of successfully delivered packets passing through from the sources to the destination.

B. Simulation Results

Fig. 4 and Fig. 5 plot simulation results of our experiments for *almost connected* and *intermittently connected* scenarios, respectively.

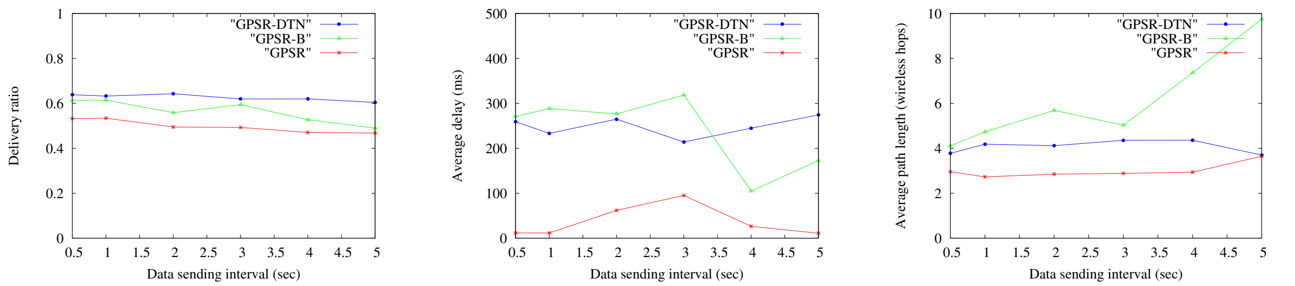
As shown in Fig. 4(a) and 5(a), GPSR has the lowest data delivery ratio, GPSR-DTN and GPSR-B have higher delivery ratio than GPSR does in both scenarios. This confirms that combining DTN routing strategies with position-based routing improves the chances of final delivery. Notice that GPSR-B may lead to routing loops due to the lack of knowledge of moving directions, thus it has lower delivery ratio than GPSR-DTN. In GPSR-DTN, we carefully design the loop prevention mechanism based on the moving directions and whether two vehicles are on the same segment, so that the best vehicle is selected to hold or forward the packets. In addition, it is clear that all protocols have better performance under the almost connected scenario than under the intermittently connected scenario.

Fig. 4(b) and 5(b) show the average delay of different protocols. It is obvious that all of three protocols have lower



(a) Delivery ratio (b) Average delay (c) Average path length

Fig. 4: Simulation results for almost connected scenario.



(a) Delivery ratio (b) Average delay (c) Average path length

Fig. 5: Simulation results for intermittently connected scenario.

average delay for the almost connected scenario than for the intermittently connected scenario. Better connectivity provides better relay selection and thus leads to quicker transmissions. Both GPSR-DTN and GPSR-B usually have longer delay than GPSR has, since they both apply store-carry-forward strategy which causes longer delay. For the same reason, the average path lengths of GPSR-DTN and GPSR-B are usually longer than those of GPSR, as shown in Fig. 4(c) and 5(c). Note that GPSR-B has longer average path length than GPSR-DTN due to possible routing loops.

In summary, even though with longer delay and path length, the proposed GPSR-DTN indeed improves the delivery ratio, especially in intermittently connected networks.

IV. Conclusion

Traffic lights, accidents, or low density may lead intermittent connectivity very common in VANETs. While traditional position-based VANET routing protocols are not suitable for sparsely connected vehicular networks, in this paper, we propose a new hybrid forwarding protocol which combines position-based forwarding with the idea of store-carry-forward from DTNs. In the proposed method, driving directions of vehicles are used to make forwarding or carrying decisions and a carefully designed loop prevention mechanism is also introduced. Experimental results show that the proposed hybrid method outperforms existing position-based solutions.

References

[1] F. Bai, H. Krishnan, V. Sadekar, G. Holland, and T. Elbatt, "Towards characterizing and classifying communication-based automotive applications from a wireless networking perspective," in *Proc. of IEEE Workshop on Automotive Networking*, 2006.

[2] R. A. Santos, A. Edwards, R. Edwards, and L. Seed, "Performance evaluation of routing protocols in vehicular ad hoc networks," *Int'l J. of Ad Hoc and Ubiquitous Computing*, 1(1/2):80–91, 2005.

[3] V. Namboodiri, M. Agarwal, and L. Gao, "A study on the feasibility of mobile gateways for vehicular ad-hoc networks," in *Proc. of 1st Int'l Workshop on Vehicular Ad Hoc Networks (VANET)*, 2004.

[4] G. Liu, B.-S. Lee, B.-C. Seet, C. H. Foh, K. J. Wong, and K.-K. Lee, "A routing strategy for metropolis vehicular communications," in *Proc. of Int'l Conf. on Information Networking (ICOIN)*, 2004.

[5] H. Füßler, M. Mauve, H. Hartenstein, M. Kasemann, and D. Vollmer, "Location-based routing for vehicular ad-hoc networks," *ACM Mobile Computing and Communications Review (MC2R)*, 7(1):47–49, 2003.

[6] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *IEEE Vehicular Technology Magazine*, 2(2):12–22, 2007.

[7] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *ACM/Kluwer Wireless Networks*, 7(6):609–616, 2001.

[8] B. Karp and H.T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. of ACM MobiCom*, 2000.

[9] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Tech. Rep. CS-200006, Duke University, 2000.

[10] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet," *SIGPLAN Not.*, 37:96–107, 2002.

[11] P.-C. Cheng, K.-C. Lee, M. Gerla, and J. Harri, "GeoDTN+Nav: Geographic DTN routing with navigator prediction for urban vehicular environments," *Mobile Networks and Applications*, 15(1):61–82, 2010.

[12] I. Leontiadis and C. Mascolo, "GeOpps: Geographical opportunistic routing for vehicular networks," in *Proc. of IEEE Int'l Symp. on World of Wireless, Mobile and Multimedia Networks*, 2007.

[13] J. LeBrun, C.N. Chuah, and D. Ghosal, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," in *IEEE VTC*, 2005.

[14] The Network Simulator (NS-2), http://www.isi.edu/nsam/ns.

[15] The MObility model generator for VEhicular networks (MOVE), http://lens1.csie.ncku.edu.tw/MOVE/.