GeOpps: Geographical Opportunistic Routing for Vehicular Networks

Ilias Leontiadis and Cecilia Mascolo
Department of Computer Science, University College London
Gower Street, London, WC1E 6BT, United Kingdom
{i.leontiadis|c.mascolo}@cs.ucl.ac.uk

Abstract

Vehicular networks can be seen as an example of hybrid delay tolerant network where a mixture of infostations and vehicles can be used to geographically route the information messages to the right location. In this paper we present a forwarding protocol which exploits both the opportunistic nature and the inherent characteristics of the vehicular network in terms of mobility patterns and encounters, and the geographical information present in navigator systems of vehicles. We also report about our evaluation of the protocol over a simulator using realistic vehicular traces and in comparison with other geographical routing protocols.

1 Introduction

It is anticipated that in the near future many vehicles will be equipped with wireless interfaces, enabling them to form mobile ad-hoc networks on the fly and connect with fixed infostations while passing by [7]. *Vehicular networks* are hybrid mobile ad-hoc networks where infostations and vehicles are present. Infostations are fixed access points that are potentially connected to the Internet. They may act as dissemination points, from where information from the backbone network flows towards the vehicles. Vehicles internetwork with each others, disseminating messages further.

Potential applications for vehicular networks may require to disseminate information to specific geographical areas. Such information can include, traffic condition updates, accident warnings, free parking spots, advertisements, or even vehicle synchronisation (e.g., in order to allow platoons formation) etc. Centralised solutions are of course possible but they suffer from scalability issues and might result in being quite expensive especially if the area to be covered is large. Therefore, the use of an ad-hoc opportunistic routing, possibly exploiting geographical information might be preferable.

A number of existing geographic routing protocols are available [13, 12]. However, these protocols have not been specifically designed for vehicular networks and are not suitable for a number of reasons [6]; in these networks, the topology is constantly changing but in a somewhat predictable way (e.g., cars move on roads). Furthermore, vehicles tend to move in clusters towards a specific direction, creating networks that might be not always connected (i.e., there is no end-to-end connectivity). Therefore, a *geographical* but also *delay tolerant* approach is needed.

Delay Tolerant Network (DTN) protocols [9] provide communication in performance-challenged environments where continuous end-to-end connectivity cannot be assumed. These protocols often employ a store-and-forward message switching: fragments of a message (or the whole message) are forwarded and stored from host to host along a path until the message reaches the destination. This data exchange occurs during *opportunistic contacts* of the hosts. The mobility patterns of the hosts and the selection of the next message carrier are responsible for the successful (or not) delivery of the message.

There are a number of existing delay tolerant routing protocols ([18, 20, 17], etc) These protocols exploit different mechanisms to route a message to the destination such as statistics of previous encounters, or precise mobility schedules to find the best routes. However, none of these approaches was specifically designed for vehicular networks. Moreover, projects that employ vehicular DTN routing protocols like CarTel [2], DieselNet [1], Drive Through Internet [16], FleetNet [11] are available. These systems however do not consider geographical routing.

In this paper, we will present GeOpps: a novel delay tolerant routing algorithm that exploits the availability of information from the navigation system (NS) in order to opportunistically route a data packet to a certain geographical location. We take advantage of the vehicles' NS suggested routes to select vehicles that are likely to carry the information closer to the final destination of the packet. In our work, we assume that vehicles are equipped with a satellite navigation system. Navigation systems typically consist of a Global Positioning System (GPS) device, maps, and the appropriate hardware and software. Their main function is to calculate a suggested route from the current position of the vehicle to the destination of the driver. To the best of our knowledge, the closest works to ours are Move [10] and Greedy [11, 6]. However, with respect to these works, GeOpps exploits information from the navigation system to efficiently route packets. We evaluate our work by using realistic vehicular traces plugged into a simulator and we will compare with these protocols in terms of delivery ratio, delivery delay and transmission overhead.

2 Scenario

In this section we present a scenario that helps motivating our work. We assume that vehicles are equipped with navigation systems that contain information regarding the geographical location of local infostations (access points to the Backbone). We can consider vehicles as mobile sensors that gather information about traffic and road conditions (e.g., potholes), etc. This is quite a realistic assumption and other systems build on it (e.g., CarTel [2]). A navigation system may employ various available sensors (e.g., speed sensor, acceleration sensor) and its map to evaluate the current traffic conditions of a specific road segment. Afterwards, it reports this information to the closest infostation. And since we cannot assume constant connectivity between vehicles and infostations (especially in remote areas), other vehicles need to act as data mules from the sensing vehicle to the geographical location of the nearest known infostation.

A centralised system can combine the information gathered from various sources and produce estimates of the current traffic conditions. Afterwards, it can generate traffic warnings concerning specific road segments and suggest alternative routes to vehicles that are approaching them. To warn the drivers, the traffic management centre has to dispatch this information to the vehicles in these areas. Initially, warnings are sent to the nearest infostation. From here, they need to be routed to the affected road segments using the vehicular network. Upon reaching the area, local message dissemination techniques (like constrained flooding or localised epidemic) can be employed to spread the information to nearby vehicles. The navigation system of vehicles that receive such a warning can evaluate the information provided and automatically recalculate a route avoiding road segments that are currently congested.

Geographic delay tolerant routing protocols should be utilised to route the information *from* and *back* the vehicles from an infostation. We illustrate our suggested approach in the next section.

3 GeOpps

In this section, we present GeOpps: a geographical delay tolerant routing algorithm that exploits information from the vehicles' navigation system to route messages to a specific location. Briefly, to select the next packet carrier:

- Neighbour vehicles that follow *suggested routes* to their driver's destination calculate the *nearest point* that they will get to the destination of the packet.
- Afterwards, they use the nearest point and their map in a *utility function* that expresses the *minimum estimated time* that this packet would need in order to reach its destination.
- The vehicle that can deliver the packet quicker/closer to its destination becomes the next packet carrier.

We will now give a detailed description of GeOpps.

3.1 Navigation System

There is a large number of available navigation systems (NS) like TomTom, Destinator, Microsoft AutoRoute, Route 66 and many others. These systems provide turn-to-turn navigation assistance to the driver until the vehicle reaches the destination. The driver may select his/her destination and preferences (e.g., calculate fastest route, shortest route, avoid tolls, etc), and the navigation system calculates a *suggested route* from the current position of the vehicle to the final destination. To calculate the suggested route, the map of the navigation system contains information about speed limits and average speed statistics and it employs a shortest path algorithm for weighted graphs. Furthermore, NS provides information about the Estimated Time of Arrival (ETA) of the vehicle to the destination.

3.2 Calculation of the Nearest Point

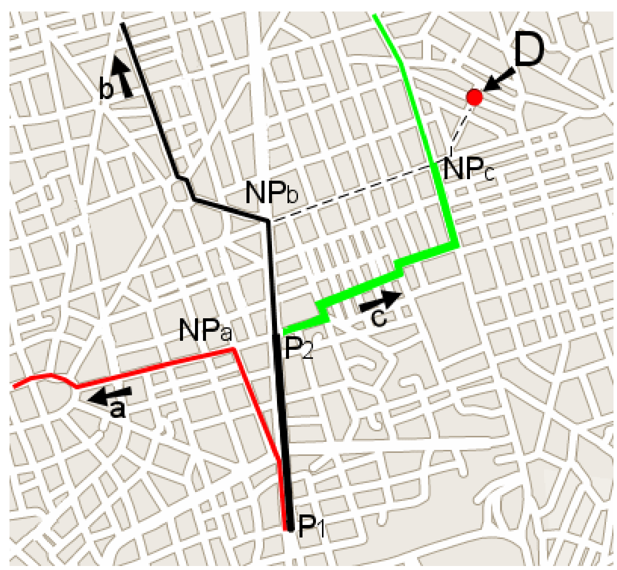

Figure 1: Example of calculation of the Nearest Point (NP) from packet's Destination (D).

Let us assume that a vehicle has a calculated route to its destination. When this vehicle is given a data packet for a specific geographical location D, it is able to calculate a point NP on its suggested route that is *its nearest point* to D. In other words, it calculates the closest point to the destination of the packet that this vehicle is going to reach. Figure 1 demonstrates an example of this calculation for vehicles a,b and c.

To find NP, the actual road-distance (dashed line in our example) or the straight-line distance from NP to D may be used. The first technique is more CPU intensive because it requires running a weighted shortest path algorithm (already implemented in the NS) from every intersection along the suggested route of the vehicle. The second method is less precise (because it assumes that smaller line-distances result in smaller road distances, which, however, is usually the case), but it is much faster.

3.3 Utility Function

When a vehicle encounters one or more vehicles (contacts), the NS has to evaluate if it should keep the packet or forward it to a selected neighbour. To make this assessment a *utility function* is computed. The utility function provides an estimate of the *minimum time* that a packet would need to reach its destination D.

After calculating the nearest point, the NS can use map information to calculate the ETA of the vehicle to NP. Similarly, it can also calculate the *estimated time that a vehicle would need from NP to the final destination D of the packet*. The sum of these two values is an indication of how much time is needed for this packet to be delivered if this vehicle carries it until NP. This assumes that when the current vehicle arrives to NP there will be another vehicle around that will carry the packet to its final destination D. This is why this measure is called *Minimum Estimated Time of Delivery for the packet* (METD). Therefore:

$$\text{METD} = \text{ETA to NP } + \text{ETA from NP to D}$$

It is obvious that this value is mainly affected by how close this vehicle is going to travel to the destination of the packet. For example in Figure 1, the METD value for vehicle b will be lower from the value of a because the time required to go from P_1 to NP_a and then to D is higher than the time required to go from P_1 to NP_b and then to D.

If the straight line calculation of distance is used instead of route calculation from NP to D (due to higher processing time), we can further simplify this utility function by using the straight-line distance between NP and D:

$$\text{METD} = \text{ETA to NP } + \frac{\text{Distance Bet. NP and D}}{\text{Average Speed}}$$

Although this is less accurate than the previous, it allows us to calculate a close approximation of the actual $METD$. Furthermore, the *Average Speed* can be considered as a weight that puts emphasis on distance (i.e., select a vehicle that is going closest to the destination no matter how much time it takes) or delay (i.e., select a vehicle that might not be going that close, but it is getting to NP faster). It is expected that high values of the estimated average speed of a possible vehicle between NP and D will decrease the delivery ratio of the algorithm.

3.4 Carrier choice

The main step of the algorithm is to keep looking for vehicles that can potentially deliver the message earlier (i.e., vehicles that minimise the $METD$). This means that either these vehicles plan to go *closer to the destination* (i.e., estimated time from NP to destination is low due to lower distance) or that they will travel faster to an area where we estimate that there is a fast route to the destination of the packet (e.g., use a hi-speed highway that leads to the packet destination). Somehow then the information about the routes of the various cars needs to be exchanged among the vehicles. The algorithm follows these steps:

- Vehicles periodically broadcast the destinations of the packets that they have stored.
- One-hop neighbours, calculate the Minimum Estimated Time Of Delivery ($METD$) that they require to deliver the packet and send this value to the enquiring vehicle.
- The current carrier either keeps the packet (if it has the lowest $METD$) or forwards the packet to the neighbour with the lowest value.
- This process is repeated until the packet arrives to its destination or the packet expires.

For example, in Figure 1, at point P_1 a vehicle polls vehicles a and b for their $METD$ values. These vehicles calculate the nearest point that the will get to D (NP_a and NP_b). Vehicle b becomes the next packet carrier. As b travels to its destination, it keeps looking for other vehicles that have even lower $METD$ values. At point P_2, it encounters vehicle c that is going even closer to the destination and, therefore, it forwards the packet to c. Notice that the packet never reached NP_b.

An interesting side-effect that we noticed is that when a large group of vehicles have the same NP (e.g. a part of their route that contains NP is the same) then the packet is forwarded to the leading neighbour because it reports smaller $METD$. Therefore, when the density is high, the packets travel faster than the flow of vehicles.

3.5 Special Cases

This protocol exploits navigation information (e.g., suggested route, ETA) to opportunistically select a neighbour that is estimated to get closer and faster to the destination of the packet. However, there are some assumptions concerning the accuracy of this information.

We have to consider what happens in cases where the drivers do not follow the suggested route. When a driver deviates from the route, its navigation system automatically recalculates an alternative route and ETA. In every contact, the NS always uses the latest METD value and thus, this includes any deviation. There is also the case where a vehicle is ignoring the calculated route. Most of the existing navigation systems will automatically cancel a route if the driver misses a number of turns or deviates completely from the suggested. This behaviour can also be detected by observing a sequence of missed turns: the solution used here is to just forward the message to any neighbour. Furthermore, the NS can constantly evaluate the driver's behaviour in order to predict how likely he/she is to follow the suggested route.

Additionally, we should also consider vehicles that stop/pause their trip. If the driver switches off the engine, the system will forward all the messages to any neighbouring vehicle. In case the vehicle stops for a long time without switching off the engine (in our simulation more than five minutes) the NS forwards all the messages to any neighbour.

Finally, notice that GeOpps does not require all the vehicles to have calculated routes (e.g. id does not require all the drivers to indicate their destination). Source vehicles may begin routing the packets using a greedy algorithm until the packet contacts a vehicle that has a calculated route that leads closer than the current position. Furthermore, GeOpps delivery ratio doesn't directly depend on the high density of such vehicles but only on the road topology and the mobility patterns of the vehicles (e.g., the probability to find a vehicle that is going closer to the destination of the packet).

4 Evaluation

To evaluate GeOpps, we used *OMNet++*[15, 19], a discrete event simulation environment and the *mobility framework plug-in* [5], which supports node mobility, dynamic connection management and a wireless channel model. We also used realistic vehicular traces in order to make this simulation as realistic as possible.

We compare our protocol with two other approaches: *Location-Based Greedy* routing and *MoVe* routing algorithm. The former algorithm is a DTN variation of existing location-based greedy algorithms [11, 12] where the packet is forwarded to the neighbour that is closest to the destination (if closer than the position of the current carrier). This process is repeated until the message reaches its destination. MoVe algorithm[10] uses information about relative velocities of the current vehicle and the neighbours to predict the closest distance that they vehicles are predicted to get to the destination following their current trajectories (straight-line paths).

To accurately evaluate our protocol in the context of vehicular networking, it would not make much sense to use any random mobility models [3]. Because no large scale vehicular traces exist, we have evaluated our approach by using traffic traces generated by a multi-agent microscopic traffic simulator (MMTS) developed by K. Nagel at ETHZ [14, 4]. These traces contain mobility patterns of

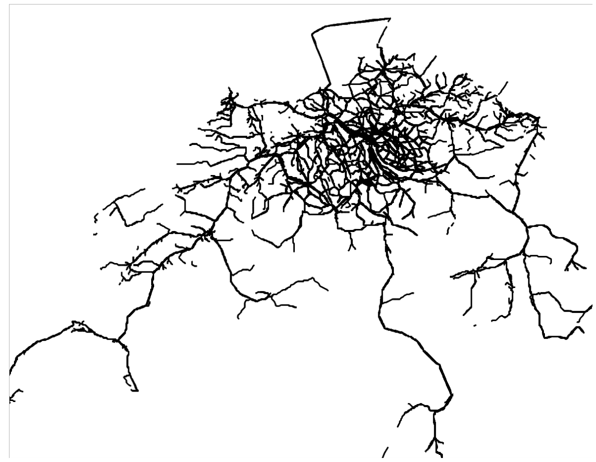

Figure 2: Map of the Vehicular Traces

260,000 vehicles over real road maps in the canton of Zurich within a period of 24 hours. (Figure 2).

4.1 Simulator settings

For our evaluation we extracted smaller areas from the 250km x 260km area of the traces. The reason is that simulation of 260,000 vehicles makes the simulation extremely slow. The area that we selected is 15km x 15km large and contains the city centre. More than 21,500 cars enter the simulation area during the peak-time of our simulation, with an average of more than 2,000 cars in the area at the same time.

The vehicles are equipped with 802.11b [8] wireless radio interfaces. The maximum possible communication range is 250m. All the broadcasts occur at the same channel frequency.

During the simulation, 1,000 random vehicles are selected and from each, a packet is sent through each of the three protocols to the same destination D. These packets are then routed using the three algorithms that we compare. We measure the delivery ratio, hop count and delay. To calculate the nearest point and evaluate the ETA we use the simplified version described in Section 3. Vehicles always follow their suggested routes and poll their neighbours every 5 seconds. The results that we present are averages of 20 runs.

4.2 Results

Figure 3 illustrates the cumulative number of packets delivered within a certain time after sending. We observe that GeOpps is able to deliver nearly 98% of the packets within twenty minutes in the 15km x 15km scenario. At the same time, Greedy delivered 72% of the packets whereas MoVe 53%. These results indicate that GeOpps can deliver the vast majority of the packets to the final destination. The MoVe algorithm shows poor performance due to the fact that the current trajectories of the vehicles do not actually

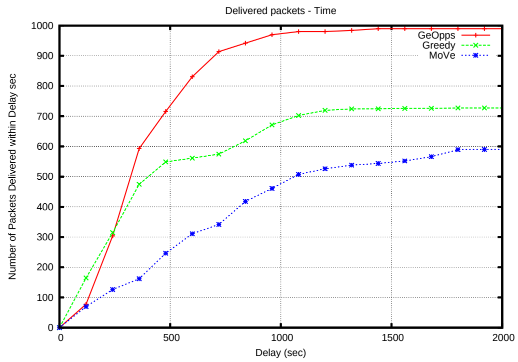

Figure 3: Delivery ratio through time.

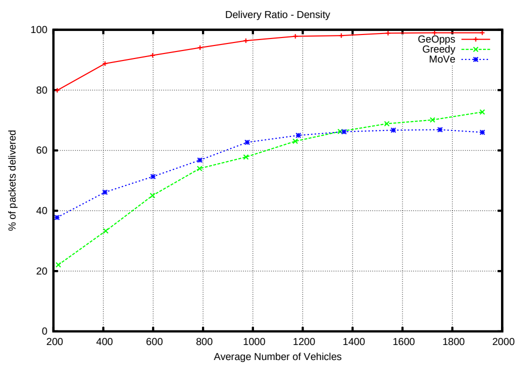

Figure 4: Delivery ratio for different densities.

indicate their final destination because vehicles have to follow the road topology. Greedy delivers some of the packets quickly (mainly packets generated near the destination) but total delivery ratio is only 73% because of the higly partitioned (and mobile) vehicular network (messages sent from remote -not directly connected- areas were not delivered). In fact, GeOpps delivers 73% percentage of messages earlier than greedy.

In Figure 4, we have plotted the delivery ratio of the algorithms for varying densities (TTL is 1800sec). Greedy shows acceptable performance only in dense networks (peak-time) due to the fact that it requires the presence of neighbours that are closer and closer to the final destination. In fact, MoVe algorithm outperforms Greedy in sparse road traffic conditions where trajectory information is more important than the position of the neighbours. However, GeOpps is able to outperform both algorithms in any network conditions. It is adequate to find only one vehicle that will carry the message to its destination and thus, it is not required to have very frequent encounters like greedy and MoVe. More encounters just increase the probability to find an ideal carrier.

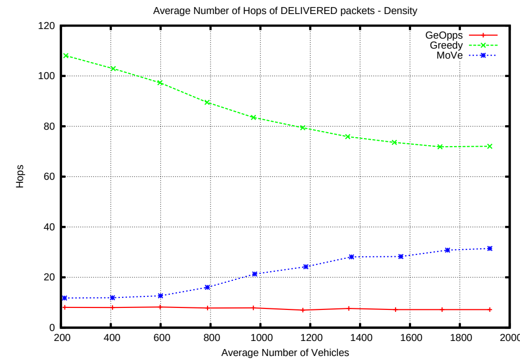

Figure 5: Average number of hops for different network densities. Smaller is better.

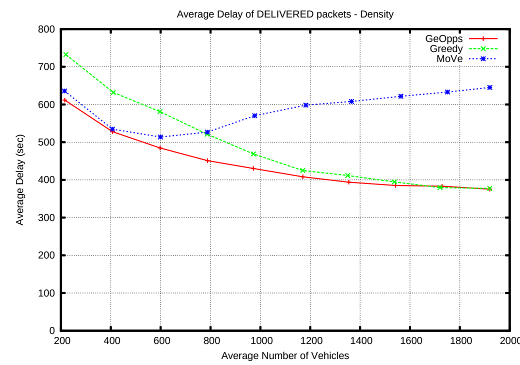

Figure 6: Average packet delay for different network densities. Smaller is better.

We can further support this observation by evaluating the number of hops required to deliver a message, shown in Figure 5. We notice that the number of hops required for Greedy is much higher than for the other two algorithms, because it constantly attempts to forward the message to neighbours that are closer to the destination. However, GeOpps requires only a few encounters before finding a vehicle that drives near the destination of the packets. Furthermore, this number does not depend on the density of the network but only on the road topology (e.g., the probability to find in this road segment a vehicle that is going close to the destination of the packet).

Figure 4 also indicates the delivery ratio for different penetration of navigation systems. If we compare the delivery of ratio of Greedy and MoVe when we utilise 2000 vehicles to GeOpps when we utilise only 200 vehicles (10% of the drivers use their navigation system), we notice that GeOpps still delivers more packets (about 80% compared to 70% of Greedy).

Additionally, Figure 6 depicts the average delay of *delivered packets*. As we can see the delay of our algorithm

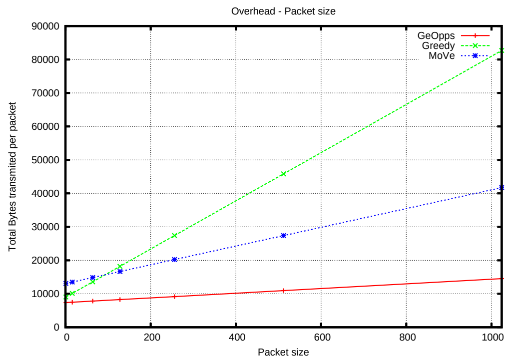

Figure 7: Overhead for different packet sizes.

is lower than that of the other two algorithms which is another indication that the minimisation of $METD$ is effective. Furthermore, the delay drops as the density increases because the probability to find a better carrier is higher and because the packets hop to leading vehicles as we discussed in section 3.4.

Finally, Figure 7 demonstrates the transmission overhead of the delivered messages for various packets sizes. As we can see the message overhead of Greedy is high due to the fact that packets require a high number of hops before delivered. The results indicate that our algorithm is able to deliver almost 99% of large packets with less than one fifth the overhead of Greedy.

5 Conclusions

In this paper we have illustrated GeOpps an opportunistic geographical routing algorithm. The main contribution of this protocol is the exploitation of available information in modern vehicles to efficiently select the next packet carrier. We have evaluated our approach by using realistic traffic traces generated by a traffic simulator. The results show good performance in various settings in terms of delivery ratio, delay and overhead with respect to existing algorithms.

This paper has not considered various optimisations of routing algorithm (e.g., polling when the vehicle is on a crossroad) . Furthermore, privacy and security aspects of the vehicular forwarding ought to be addressed.

Acknowledgements: We would like to thank Anders Lindgren and Mirco Musolesi for comments on an earlier draft. We also acknowledge the support of the EPSRC through Project CREAM.

References

[1] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. In *Proc. IEEE INFOCOM*, April 2006.

[2] V. Bychkovsky, K. Chen, M. Goraczko, H. Hu, B. Hull, A. Miu, E. Shih, Y. Zhang, H. Balakrishnan, and S. Madden. The cartel mobile sensor computing system. In *SenSys '06*, pages 383–384, New York, NY, USA, 2006. ACM Press.

[3] T. Camp, J. Boleng, and V. Davies. A Survey of Mobility Models for Ad Hoc Network Research. *WCMC*, 2(5):483–502, 2002.

[4] ETH Zurich Vehicular Network Traces. http://lst.inf.ethz.ch/ad-hoc/car-traces/.

[5] M. F. for OMNeT++. http://mobility-fw.sourceforge.net/.

[6] H. Fubler, H. Hartenstein, D. Vollmer, M. Mauve, and M. Kasemann. MobiCom poster: Location-Based Routing for Vehicular Ad-Hoc Networks. *SIGMOBILE*, 7(1):47–49, 2003.

[7] R. G. Herrtwich. Communicating Vehicles - Communicating Roadways: New Approaches to Driver Information and Road Safety. In *Invited Talk at MobiCom*, 2005.

[8] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-speed Physical Layer Extension in the 2.4 GHz Band. Technical Report IEEE Std 802.11b-1999 (R2003), IEEE, 2003.

[9] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *SIGCOMM '04*, pages 145–158, New York, NY, USA, 2004. ACM Press.

[10] J. Lebrun, C.-N. Chuah, D. Ghosal, and M. Zhang. Knowledge-Based Opportunistic Forwarding in Vehicular Wireless Ad Hoc Networks. *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, 4:2289–2293, 2005.

[11] C. Lochert, H. Hartenstein, J. Tian, H. Fussler, D. Hermann, and M. Mauve. A routing strategy for vehicular ad hoc networks in city environments. *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 156–161, 2003.

[12] M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *IEEE Network*, 15(6):30–39, 2001.

[13] P. Mohapatra, C. Gui, and J. Li. Group communications in mobile ad hoc networks. *Computer*, 37(2):52–59, 2004.

[14] V. Naumov, R. Baumann, and T. Gross. An Evaluation of Inter-Vehicle Ad-Hoc Networks Based on Realistic Vehicular Traces. In *MobiHoc '06*, pages 108–119, New York, NY, USA, 2006. ACM Press.

[15] OMNET++ Community Website. http://www.omnetpp.org/.

[16] J. Ott and D. Kutscher. A disconnection-tolerant transport for drive-thru internet environments. *INFOCOM 2005*, 3:1849–1862 vol. 3, 2005.

[17] A. Pentland, R. Fletcher, and A. Hasson. Daknet: rethinking connectivity in developing nations. *Computer*, 37(1):78–83, 2004.

[18] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: modeling a three-tier architecture for sparse sensor networks. In *Sensor Network Protocols and Applications*, pages 30–41, 2003.

[19] A. Varga. The OMNET++ Discrete Event Simulation System. *Proceedings of the European Simulation Multiconference (ESM'2001)*, 2001.

[20] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *MobiHoc '04*, pages 187–198, New York, NY, USA, 2004. ACM Press.