REACT: routing protocol for emergency applications in car-to-car networks using trajectories

REACT: Routing Protocol for Emergency Applications in Car-to-Car Networks using Trajectories

Erwin Van de Velde[1], Chris Blondia
PATS Research Group
Dept. Mathematics and Computer Science
University of Antwerp - IBBT VZW
Middelheimlaan 1,
B-2020 Antwerpen, Belgium
{firstname.lastname}@ua.ac.be

Luca Campelli

Dipartimento di Elettronica e Informazione
Politecnico di Milano
Piazza L. da Vinci 32
20133 Milan, Italy
{lastname}@elet.polimi.it

***Abstract*— Traffic safety is a very important issue in modern society and the last couple of years researchers have started looking at telecommunication solutions to enhance traffic safety. More specifically researchers are looking at emergency alert services, enabling cars to warn eachother for incidents (slippery road, accident, . . .). Existing forwarding protocols for vehicular networks often use an approach based on geographical information as a topology based forwarding protocol fails to adapt to the quick movement of the vehicles. A geographical based approach however uses only a part of the information available to the system. In this paper a new routing algorithm, REACT, will be proposed, using not only geographical information, for both position and speed, but also road map information. Using the knowledge of both neighbors and road maps, a node that wants to send a packet can choose the best forwarder along a given trajectory.**

I. Introduction

Although the concept of position-based routing exists already for several years, it was not used as position information was often not available and the more traditional approach of topology-based routing performed well in MANETs [11] with low mobility. In a vehicular network however, more and more nodes are equipped with a GPS system, providing both positional and road map information, and the typical speeds are too high for topology based routing to keep a correct view of the network. Some research has already been conducted in projects and consortia (e.g. Fleetnet[6], DSRC [5], Car 2 Car Communication Consortium[3]), but this research area is still fairly new.

A vehicular network is a mobile ad hoc network, but differs from other ad hoc networks in the following aspects:

- Nodes move fast.
- Nodes also follow certain patterns, directly linked to the road topology as vehicles stay on the road. Some routes are more likely than others: e.g. on a highway most vehicles follow the highway instead of taking the exit.
- Energy constraints are not that high as every vehicle has a large enough battery capacity.

The challenge of forwarding in vehicular networks is to take these specific properties into account and to use them.

Due to the fast movement of the nodes, the topology changes quickly, i.e. links come up and are broken very rapidly. Positional information however is much more predictive: if a node is on a certain position, the next second it will still be there in the vicinity. With two position updates of a node, it is even possible to estimate both speed and direction. Topology based routing protocols keep a view of the entire network (proactive routing protocols, e.g. OLSR [4]) or of the routes they use (reactive protocols, e.g. AODV [13], DSR [9]), where position-based routing protocols often use only information of the neighbors which reduces the chance of having outdated information.

Other routing protocols like GPSR [10] and CBF [7] already use geographical information. However, these protocols always try to get closer to the destination

[1]Funded by grant of the Fund for Scientific Research Flanders (Aspirant)
[2]Partially funded by EuroNGI (VNET)

considering only geographical information. In vehicular networks one can take more intelligent forwarding decisions with knowledge of the road maps. Due to the fact that in emergency alert services one wants to only notify drivers on the road driving towards the place of the incident, it is better to make the alert message 'stay on the road' rather than to try to find the shortest path in a geographical sense. This can be done using prediction of node movement and will reduce protocol overhead as you restrict the network traffic only to some trajectories along the road.

Due to the lacking of any severe energy constraints, beaconing is not an issue, nor is sending only broadcast packets and having some more complicated calculations.

As this paper tries to solve the routing and forwarding problem in vehicular networks for emergency applications, the following properties should be fulfilled too:

- Packets must reach their destination, information loss should be avoided even when network conditions are bad (e.g. sparse topology, i.e. too few cars for full connectivity all of the time)
- Delay must not be too high

"Vehicle-to-Vehicle Wireless Communication Protocols for Enhancing Highway Traffic Safety" [2] presents an intelligent broadcasting protocol using geographical information which somehow resembles the REACT protocol: it uses broadcast messages and neighbors forward the message with a forwarding decision based on geographical information. However there are also many differences: in the REACT protocol the previous forwarder (or source) appoints the next forwarder while in the intelligent broadcasting protocol the receiver decides on forwarding using a time out. The REACT protocol also has support for temporarily forwarding failure due to topology gaps (temporarily because of the mobility in the vehicular network), which lacks in the intelligent broadcasting protocol of the cited paper but is required by the emergency application as the information can be critical.

REACT uses the special properties of vehicular networks and tries to fulfill the requirements for emergency alerts. The protocol uses trajectory-based forwarding where each node elects the forwarder in the list of neighbors, instead of the forwarder electing itself by timer time out as happens in some other ad hoc routing protocols (e.g. BLR [8]). It uses beaconing to keep an up to date view of the neighborhood. The beacons contain information about the current position of the nodes and all forwarding decisions are made without using information of other nodes but the neighbors. The protocol's decision is also based on knowledge of the road maps as it tries to elect forwarders only among those neighbors that are driving on the roads on which the forwarder has based the alert's trajectory. This should prevent the packet being handed over to a forwarder that leaves the trajectory and that could be unable to forward the packet any further along the roads it should follow.

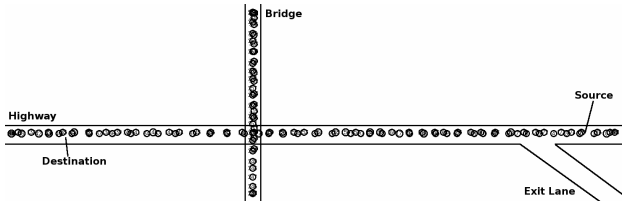

Fig. 1. Scenario Description

A. Scenario Description

The scenario that is being demonstrated in this paper is a highway scenario with a crossing bridge and an exit lane. In figure 1, you see the highway on the horizontal axis and the bridge on the vertical axis. The highway has an exit lane that should be chosen by cars driving from left to right on the highway when they have received the alert message. An alert packet is being transmitted at the Source and should reach the destination area as soon as possible. Some scenario parameters can be found in table I.

Description	Value
Highway length	10 km
Bridge position	4 km on highway
Bridge length	3 km
Sender	9.5 km on highway
Destination	1.5 km on highway
Exit position	8.5 km on highway

TABLE I
SCENARIO PARAMETERS

The following assumptions were made in this paper:

- All links are bidirectional
- All vehicles are equipped with GPS receivers

II. Protocol Description

A. Description of the protocol basics

Every node must always send beacons every β seconds containing the following information:

- Timestamp
- Position of the node

While the node requires GPS anyway to get its position information, GPS can also be used to synchronize its clock [12].

Each node receiving the beacon will add the neighbor if it did not know that neighbor yet or will update its position, keeping the previous measurement in order to estimate the speed and direction of that neighbor. If no beacons have been received in $\gamma * \beta$ seconds (γ is the number of beacons a node is allowed to miss), the neighbor will be removed from the neighbor list.

Whenever a node generates an alert message, it will use the following algorithm to find the best forwarder:

1) Calculate distance between the current node and the destination along the trajectory, we call it δ_0 and we take $\delta_{cur} = \delta_0$.
2) If $\delta_0 < \Delta$, with $2 * \Delta$ diameter of the destination range, the algorithm stops.
3) For each neighbor in the neighbor list:
   a) Calculate the distance between the neighbor's last known position and the destination and call it δ_i. If $\delta_i < \delta_{cur}$, proceed, else go to the next neighbor.
   b) Calculate the cosine of the angle of the trajectory and the neighbor's direction (if two positions are available for that neighbor). If the cosine of the angle is smaller than α, proceed, else go to the next neighbor. The value of α depends on the maximum angle between the trajectory and the neighbor's direction you want to allow for the next forwarder and is defined as the cosine of this maximally allowed angle.
   c) The neighbor is accepted as best option until now, $\delta_{cur} = \delta i$
4) If a neighbor that is in a good position to forward the packet has been found, elect him as next forwarder.

The packet sent into the network contains the data fields described in table II.

Packet class (beacon or alert message)
Source ID and location
Trajectory and final destination of the packet
Next Forwarder ID
Packet Life Time (until when is this packet valid in the network, equivalent to the Time To Live in the IP protocol)
Message ID
Message

TABLE II
Packet information

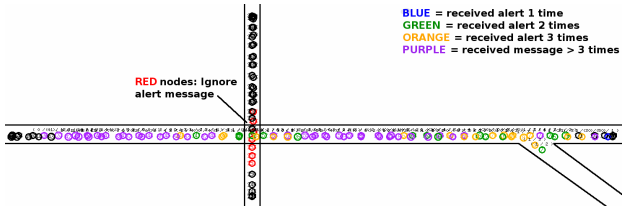

Fig. 2. Simulation in action

The source of a packet calculates the trajectory location using information from its GPS system. The trajectory should cover all cars that could be affected by the alert (e.g. accident warning). Packet Life Time is given by the source as $current\ time + validity\ time$ and gives the period of time in which the information in the packet is valid. When it is exceeded, the packet should be dropped. The next forwarder ID is filled in with a neighbor's ID if a next forwarder is found or a special flag in case the destination has been reached. If the destination has not been reached, but there is no better next forwarder (i.e. there are no vehicles on the trajectory closer to the destination), the packet will be broadcasted once more with an invalid neighbor address (special flag address). The message ID can be used to identify a unique message in the network (together with the Source ID) and the message itself should be delivered to the application. The exact information in the message part of the packet has been abstracted in this paper.

On reception of an alert message, the node will check if the packet is valid along its own driving trajectory (i.e. if the trajectory of the alert message and the node's trajectory coincide). If so, the packet will be delivered to the emergency application where e.g. the driver will be warned. If the ID of the next forwarder in the packet is the node's own ID, it will recalculate the next forwarder using the same algorithm as described above.

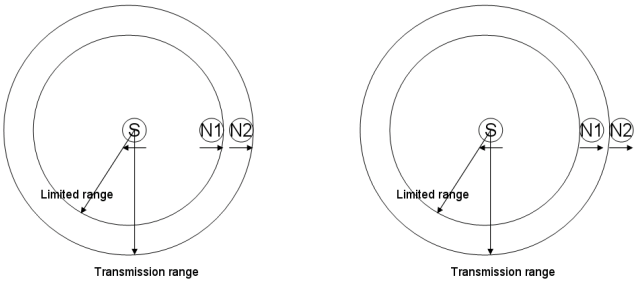

Fig. 3. Range Limiting

B. Protocol enhancements

1) Acknowledgements: While the wireless medium can sometimes be error-prone and popular MAC protocols like IEEE 802.11 do not have acknowledged delivery of broadcast packets, it is necessary for the routing protocol to acknowledge delivery as the messages deal with safety issues in vehicular networks. This is easily done by keeping the packet a node has forwarded and listening to further broadcasts. If a node does not overhear that the message it sent is being forwarded (and forwarding was expected), it will retransmit the message (recalculating the next forwarder to avoid choosing a neighbor that has gone).
It is true that it is still possible that a node (not being a next forwarder) does not hear the alert message due to collisions etc, but most of the nodes will overhear the broadcasted message twice (once from a car driving closer to the incident and once from a car driving further away from the incident, the next hop of the first one). This already gives more certainty of packet reception.

2) Overcoming topology gaps: If no next forwarder can be found, it should not mean that the packet is dropped and thus the information is lost in the network. As long as the information in the packet is valid, it should be kept in the network. Therefore, an adaptation to the protocol is made: the packet is sent with the "no forwarder found" flag, in order to acknowledge reception of the packet to the previous forwarder. The packet is queued and after τ time, the node will try again to find a next forwarder. This continues till a next forwarder has been found or till the Packet Life Time has been reached and the packet must be discarded.

3) Range Limiting: One could select a next forwarder from neighbors in the full transmission range, but this can lead to undesirable side effects. A neighbor that was in the transmission range at the moment of its last beacon, but at the edge, can already have left this range and choosing this neighbor as next forwarder will lead to an acknowledgement timeout. As alerts should travel as fast as possible through the network due to time constraints, this situation should be avoided. Using a limited range in the algorithm can help preventing this situation. If you use a range smaller than the transmission range, the probability of the next forwarder leaving the transmission range before receiving the alert can be dramatically reduced. E.g.: Figure 3: if node S wants to transmit a packet to D, it can choose between nodes N1 and N2. N2 is closer to the destination than N1 (and suppose both are on the right trajectory), so naively N2 would be chosen as next forwarder. However, if after receiving a beacon from node N1 and N2, node S has moved in the opposite direction as those nodes, it can happen that N2 is out of the transmission range of S. Choosing N2 as next forwarder would introduce delay as node S has to wait for a transmission acknowledgement that will not come. If you limit the range used in the next forwarder selection algorithm, it is more probable that the node will still be in range. Thus if N1 has been chosen, the algorithm ensures a higher probability of the next forwarder still being in range (unforeseen events like acceleration, N1 being switched off etc. can still result in an acknowledgement timeout).
On the other hand, if traffic is very sparse, it is more likely that the current forwarder can find a next forwarder in the full transmission range than in the limited range. Even when there is no next forwarder in range, it will definitely arrive in the transmission range faster than in the limited range.

C. Protocol Beaconing Overhead

The beaconing overhead depends on three parameters: the beaconing interval, the number of neighbors that a vehicle has and the size of a beacon. Since the number of vehicles depends on the vehicle density on the road, this can vary strongly. In the scenario as described above, we get the following:

Suppose that the distance between two cars is uniformly distributed over $[30m, 60m]$ (this is the calculation for the most dense scenario in the simulations in the next section), which gives on average a distance of 45m. On a part of the highway further than R = 250m (transmission range) away from the bridge, a car can see $c = \frac{4*R}{45}$ other cars on average. $4 * R$ is 4 times the transmission range: one lane in each direction and cars traveling in front of and behind the car. For the simulation below this results in the most dense scenario in $c = \frac{1000}{45} = 22.22....$

The beacon consists of a position of 32 bit (which is

enough to define the position of a vehicle in an area of more than 4200 square km with a granularity of 1m) and a timestamp and source identifier which are also 32 bits long. An 8 bit long field identifies the packet as a beacon (1 bit used for beacon/alert identification, other bits for future use). This results in a size of 13 bytes per beacon.

If we take the fastest beaconing interval of the simulations (0.25s), the total bandwidth on average used by beaconing for this car density is $\frac{1000}{45} * 13 * 4 = 1155.55...B/s$ (overhead on the network layer). On a 2 Mbps link, as used in the simulations below, this is negligible.

III. Simulation Study

The simulations were carried out using the NS-2 network simulator [1]

A. Simulation Description

Description	Value
Transmission range	250m
Bandwidth	2 Mbps
Highway speed	30m/s (108km/h)
Bridge speed	20m/s (72km/h)
Distance between cars	between $x * v$ and $2 * x * v$, uniformly distributed
x	between 1 and 15
Destination range	50m
Beaconing interval (=τ, =β in the algorithm description)	0.25s, 0.5s or 1.0s
Limited range	220m, 235m or 250m
Distance source - destination	8000m
γ in algorithm description	3
α in algorithm description	0.5

TABLE III
Range Limiting

For the first simulation results a limited range of 250m (no range limiting, full transmission range) was used and the beaconing interval was changed (see table III) and for the second simulation the limited range was varied for a fixed beaconing interval of 0.5s. For both simulations there were ten runs for each value of x between 1 and 15.

Concerning the parameter x, if $1 <= x <= 3$, the distance between two consecutive cars will be smaller than 220m (between 30 and 60m for 1 and between 90m and 180m for 3) and there is full connectivity for all limited ranges. If $x = 4$ (distance between two consecutive cars is between 120m and 240m), there is only full connectivity guaranteed when using the full transmission range. For smaller limited ranges and for $x >= 5$ (distance between two consecutive cars is between 150m and 300m) full connectivity cannot be guaranteed as the distance between two consecutive cars can be larger than 250m. However, cars driving in the other direction and the network mobility can still provide connectivity over time (i.e. sometimes a node has to wait for a next forwarder, but connectivity is still possible). From $x = 8$ (distance between two consecutive cars is between 240m and 480m) or $x = 9$ (distance between two consecutive cars is between 270m and 520m) on, depending on the limited range used, the distance between two consecutive cars is definitely larger than the limited range and so two consecutive cars will never be able to talk directly to eachother. From than on communication is almost impossible as you have to depend on cars driving in the opposite direction, which also drive with 240m to 480m in between them (for $x = 8$). In fact in the latter case, the packet will be carried most of the time by a car C driving in the same direction the message has to travel (opposite lane of the highway) and C will pass it to cars driving towards the source when possible. While these cars are not able to forward the packet (distances too big), they will hand the packet back over to C when they cross it. This means that the packet travels only as fast as car C in these cases. Notice that all cars driving towards the incident get warned as soon as possible and that the information does not get lost even though it probably does not reach the initially intended final destination area.

B. Simulation Results

In all graphs discussed below, there will be ten different simulation runs for every tuple of parameters, which are shown separately on the graphs. This shows that you can be more or less lucky, but that a certain tendency holds in any case. The packet life time used in the simulations was 200s, all packets shown to have arrived at 300s in the graphs in fact never reached the destination but received that value in order to be visible in the graphs. The logarithmic scales have been provided for a more in detail view of the smaller differences between measurements.

1) The influence of the beaconing interval: As shown in figures 4 and 5, the difference between 1.0s and 0.5s or 0.25s beaconing interval is quite big and results in packets not arriving at the destination The difference between 0.5s and 0.25s interval is already a lot smaller. In fact most results collide and only for the most dense scenarios there is a little difference. It is true that alert

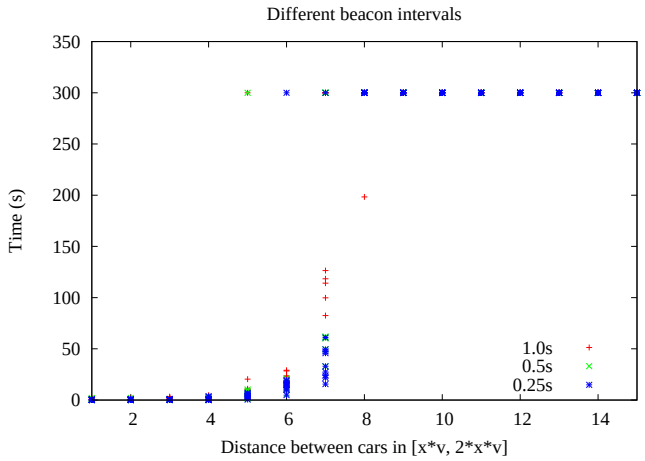

Fig. 4. Influence of the beaconing interval

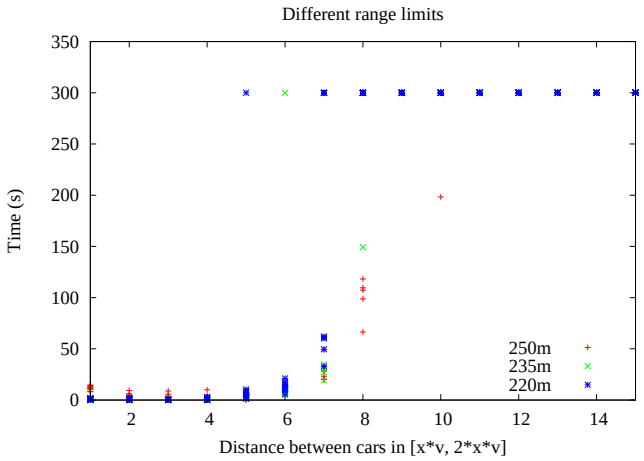

Fig. 6. Influence of the range limiting

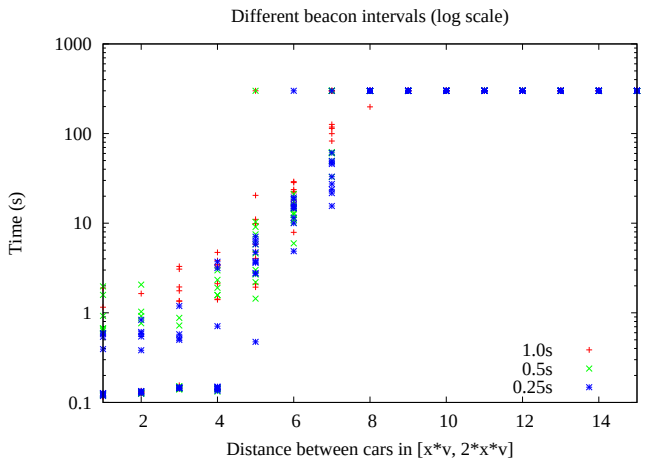

Fig. 5. Influence of the beaconing interval (logarithmic scale)

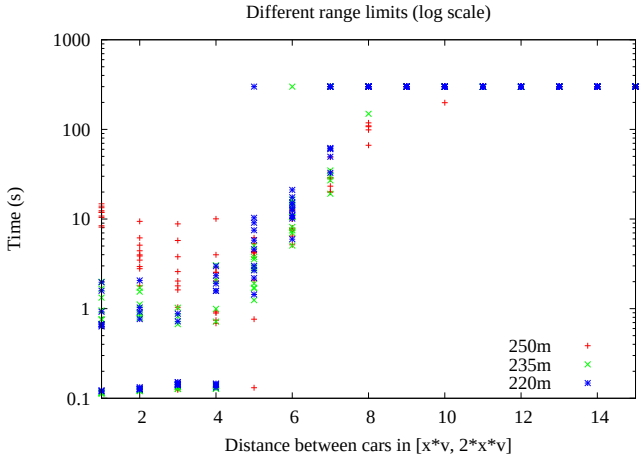

Fig. 7. Influence of the range limiting (logarithmic scale)

messages must travel fast through the vehicular network, but tenths of seconds of difference over 8000m are in fact negligible. This is a quite nice result as this says that extremely fast beaconing (magnitude of milliseconds) is not necessary.

In fact, the biggest part of the delay is caused by gaps in the topology and closing these gaps takes a lot more time than can be lost on waiting for beacons. Faster beaconing will not make the cars move any faster.

2) The influence of range limiting: As shown in figures 6 and 7, range limiting can help to avoid transmissions to a next hop that is not anymore in the transmission range. However, if you limit the range from 250m (transmission range) to 235m or 220m, it is possible that you do not see a next hop that you would see when using the full transmission range. In sparse scenarios, i.e. scenarios where the distance between cars (same or opposite direction) is bigger than the limited range, it is thus better to use the full transmission range in the next hop election algorithm.

IV. Future Work

As shown in the Simulation results (section III-B), the limited range used in the algorithm influences the time to destination greatly. An adaptive algorithm instead of a fixed range limit should be used here. A smaller gain on the time to destination is probably possible by changing beaconing interval according to the needs (e.g. Faster beaconing when you want to send an alert message but there is no better next hop and instantaneous reply of new neighbors upon receiving the beacon).

The adaptations of REACT, mostly in the way the next hop is calculated and the trajectory is described, to other scenarios like a Manhattan city model is also very

important in future work.
Furthermore, better MAC protocols for vehicular networks and the ways of using the REACT protocol with them should be investigated, more in particular the possibilities of cross-layering with these MAC protocols.

V. Conclusions

This paper presents a first version of the REACT routing protocol for vehicular networks. The protocol takes into account both the special properties of the vehicular network and the requirements of the emergency alert application using geographical and topology information. The results are promising as REACT is able to deliver packets fast and with high certainty even if the network topology is not fully connected at some moment in time. The ability to use vehicular mobility to overcome topology gaps is very important here.
The particularity of both the network and the application do not allow the use of the more common MANET routing protocols and thus a specialized protocol like REACT is necessary.

References

[1] Ns-2 network simulator.

[2] Subir Biswas, Raymond Tatchikou, and Francois Dion. Vehicle-to-vehicle wireless communication protocols for enhancing highway traffic safety. *IEEE Communications Magazine*, pages 74–82, January 2006.

[3] Car 2 car communication consortium. http://www.car-to-car.org/.

[4] T. Clause and P. Jacquet. Optimized link state routing protocol (olsr). Technical report, Project Hipercom, INRIA, October 2003. RFC 3626.

[5] Dedicated short range communications. http://www.leearmstrong.com/Dsrc/DSRCHome.htm.

[6] Fleetnet. http://www.et2.tu-harburg.de/fleetnet/index.html.

[7] Holger Füssler, Jörg Widmer, Martin Mauve, and Hannes Hartenstein. A novel forwarding paradigm for position-based routing (with implicit addressing). In *Proc. of IEEE 18th Annual Workshop on Computer Communications (CCW 2003)*, pages 194–200, October 2003.

[8] M. Heissenbüttel and T. Braun. Blr: Beacon-less routing algorithm for mobile ad-hoc networks. *Elsevier's Computer Communications Journal (Special Issue)*, 2003.

[9] David. B. Johnson, David A. Maltz, and Yih-Chun Hu. The dynamic source routing protocol for mobile ad hoc networks (dsr). Technical report, Rice University and Mellon University, July 2004. Internet Draft.

[10] B. Karp and H.T. Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *6th Annual ACM/IEEE International Conference on Mobile Computing and Networking*. MobiCom2000, 2000.

[11] Mobile ad hoc networks. http://www.ietf.org/html.charters/manet-charter.html.

[12] S. Omar and C. Rizos. Incorporating gps into wireless networks: Issues and challenges. In *The 6th International Symposium on Satellite Navigation Technology Including Mobile Positioning & Location Services*. SatNav 2003, July 2003.

[13] Charles E. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing. Technical report, Nokia Research Center and University of California and University of Cincinnati, February 2003. RFC 3561.