HHLS: A Hybrid Routing Technique for VANETs

Ayaida Marwane, Barhoumi Mohtadi, Yacine Ghamri-Doudane, Afilal Lissan

▶ **To cite this version:**

Ayaida Marwane, Barhoumi Mohtadi, Yacine Ghamri-Doudane, Afilal Lissan. HHLS: A Hybrid Routing Technique for VANETs. IEEE Global Communications Conference, IEEE GLOBECOM'12, 2012, United States. pp.44 - 48. hal-00794619

HAL Id: hal-00794619
https://hal.science/hal-00794619

Submitted on 7 Mar 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HHLS: A Hybrid Routing Technique for VANETs

Marwane Ayaida*, Mohtadi Barhoumi*, Hacène Fouchal*, Yacine Ghamri-Doudane** and Lissan Afilal*
*Centre de recherche CReSTIC,
Université de Reims Champagne-Ardenne:
51687 REIMS Cedex 2, France
*{marwane.ayaida, mohtadi.barhoumi, hacene.fouchal, lissan.afilal}@univ-reims.fr
**Université Paris-Est, LIGM, 77454 Marne-la-Vallée Cedex 2, France,
**yacine.ghamri@univ-mlv.fr
**ENSIIE, 1 Square de la Résistance, 91025 Evry Cedex, France,
**yacine.ghamri@ensiie.fr

***Abstract*—In this paper, we propose a combination between a routing protocol Greedy Perimeter Stateless Routing (GPSR) and Hierarchical Location Service (HLS) that we denote Hybrid Hierarchical Location Service (HHLS). HLS and GPSR used to be combined in the original work with a direct method, i.e. GPRS takes care of routing packets and HLS is called to get the destination position when the target node position is not known or is not fresh enough. When a destination is quite far away from the sender, the exact position of the target is calculated, and an extra overhead is generated from sender to receiver. Our main purpose is to reduce this overhead in HHLS. We suggest to proceed as follows: when a packet has to be sent to the destination, it will be sent directly to the former position of the target instead of requesting for the exact position. When the packet is approaching the former position, the exact position request is then sent.**

We have proposed a patch over the NS-2 simulator for HHLS according to our proposal. We have conducted experimentations which show promising results in terms of latency, packet delivery rate and overhead.

***Index Terms*—VANETs; Location-based Services; Geographic Routing Protocols.**

I. Introduction

VANETs (Vehicular Ad-hoc NETworks) are a special case of *MANETs (Mobile Ad-hoc NETworks)*. Their major features is the high mobility of nodes. The immediate consequences are: topology changes and link disconnections.

Usual topology-based routing protocols have limited performances in such networks. *Geographic routing protocols* were designed to provide better performances for such networks. The main principle adopted by these protocols is that each node has to care about its actual geographic position (often achieved by a GPS) and the position of the targeted node to reach. With these protocols, the paradigm position-to-position is used. The *Location-based Services* is required to catch the destination position. The combination of this service with routing is quite natural in order to guarantee interesting performances. This combination called *Hybrid Hierarchical Location Service (HHLS)* was made between *Greedy Perimeter Stateless Routing (GPSR)* as a geographic routing protocol and *Hierarchical Location Service (HLS)* as a location-based service. Several experimentations were carried out over NS-2 network simulator. These experimentations demonstrate that the efficient combination between the geographic routing protocols and the location-based services enhanced the network performances while reducing the location overhead.

The remainder of the paper is organized as follows. Section II is dedicated to related works. Section III details our HHLS combination algorithm about GPRS and HLS, it describes also our experimentations and the obtained results. Section IV concludes the study and gives some hints about future works.

II. Related works

A. Location Service

The location-based services can be classified into two classes: *Flooding-based* and *Rendez-vous-based*. The first class is composed of reactive and proactive services. In the proactive flooding-based location-based service, every node floods its geographic information through all the network periodically. Thus, all the nodes are able to update their location tables. Since this approach uses flooding and may surcharge the network by location update messages, several techniques to reduce the congestion were used. One of them is to tune the update frequency with the node mobility (the more nodes are moving fast, the higher update location frequency is used). An example of proactive flooding-based location-based service is the Distance Routing Effect Algorithm for Mobility (DREAM)[1]. This routing includes a location-based service. Behind DREAM there are two ideas: the first one is, if a node is far it appears moving slower than a neighbor having the same speed. Therefore, the update frequency decreases with the distance to the node. The second idea is, a node with high mobility sends more update location packets. As a result, there are less packets than a simple flooding scheme without affecting the network performances. For the second group (i.e the reactive flooding-based location-based service), the location response is sent when receiving a location request. This avoids the overhead of useless location information of some nodes updated and never used. But, it adds high latencies not suitable in VANETs. As a good example, we cite the Reactive Location Service (RLS) [2].

In the second class (rendez-vous-based location service), all the nodes agree on a unique mapping of a node to other specific nodes. The geographic information are disseminated

through the elected nodes called the "location servers". Thus, the location-based services proceed with two main operations:

1) Location Update: A node has to recruit location servers (chosen from other nodes) and needs to update its location through theses servers. The location servers are responsible of storing the geographic data of the relating nodes.
2) Location Request: When a node needs to know the location of another node, it broadcasts a location request. The location server will replay as soon as it receives this request.

There are two major approaches on the rendez-vous-based location services. In the quorum-based approach, as in [3], the location update is sent to a group of nodes (update quorum). The location query is sent to the same or a different group (query quorum). These two groups have not to be disjoint. Thus, the query reaches the first group which replies immediately. The challenge of this approach is to choose how to generate the query system [3].

The second approach is the hierarchical approach. The network is split into several levels. At each level a node selects location servers. The location query is forwarded up and down in the hierarchy. This principle reduces the forwarded packets and avoids flooding. The two major hierarchical services are: the Grid Location Service (GLS) [4] and the Hierarchical Location Service (HLS) [5].

As GLS, HLS shares the network into several subsets called regions which are subdivided in hexagonal cells. Close regions are grouped into region level. This partition is fixed and known by all the participating nodes. The cell dimensions must be less than the range transmission. Thus, a node may be able to broadcast a message to all nodes in the same cell. Unlike GLS, a hashing function depending not only on the ID but also on the current position of the node is used to choose the responsible cell (where a node must select its location servers) at each region level. There are two different update methods:

- The direct method: to update its location information, a node sends frequently packets to the responsible cells. A responsible cell may contain more than one location servers. This is the case only for close location servers (at the same region level). For others location servers the second methods is used.
- The indirect method: To reduce the traffic, only the location of the responsible cell at level N-1 where is located the node is sent to the higher region level N. Instead of sending the node's geographic information, the coordinate of the responsible cell are sent when the node moves from a region level to another. Consequently, the traffic congestion generated by the node's movement is local at the first level and a few multi-hop long-distance location packets are sent to the top levels.

If a node A needs to send data to a node B, A broadcasts a query location. When this location reaches a B's responsible cell and if A and B are in the same region level, a location server in the B's responsible cell sends a location reply. If they are not in the same region level, the location server forwards the packet toward the B's responsible cell in the lower level and so on until reaching the first level responsible cell. The latter generates a reply and sends it to the node A.

The location-based service used here for the combination is HLS since it has the best performances and the lowest overhead and complexity compared to GLS and RLS in [6].

B. Geographic Routing Protocols

Routing protocols algorithms must choose some criteria to make routing decisions, for instance the number of hops, latency, transmission power, bandwidth, etc. The topology-based routing protocols suffer from heavy discovery and maintenance phases, lack of scalability and high mobility effects (short links). However, geographic routing are suitable for large scale dynamic networks. The first routing protocol using the geographic information is the *Location-Aided Routing (LAR)* [7]. This protocol used the geographic information in the route discovery. This latter is initiated in a *Request Zone*. If the request doesn't succeed, it initiates another request with a larger *Request Zone* and the decision is made on a routing table. The first real geographic routing protocol is the *Greedy Perimeter Stateless Routing (GPSR)* [8]. It is a reactive protocol which forwards the packet to the destination's nearest neighbor (Greedy Forwarding approach) until reaching the destination. Therefore, it scales better than the topology-based protocols, but it does still not consider the urban streets topology and the existence of obstacles to radio transmissions. Another geographic routing protocol is the *Geographic Source Routing (GSR)* [9]. It combines geographical information and urban topology (street awareness). The sender calculates the shorter path (using Djikstra algorithm) to the destination from a map location information. Then, it selects a sequence of intersections (anchor-based) by which the data packet has to travel, thus forming the shortest path routing. To send messages from one intersection to another, it uses the greedy forwarding approach. The choice of intersections is fixed and does not consider the spatial and temporal traffic variations. Therefore, it increases the risk of choosing streets where the connectivity is not guaranteed, and then losing packets. Like GSR, *Anchor-based Street and Traffic Aware Routing (A-STAR)* [10] is anchor-based. However, it reflects the streets characteristics. A connectivity rate is assigned to the roads depending on the capacity and the number of bus using it. This metric is used in addition to traditional metrics (distance, hops, latency) when making routing decisions. As a consequence, the streets taken by busses are not always the main roads where connectivity is ensured and the greedy approach does not consider the speed and direction for the next hop selection. This is why *improved Greedy Traffic Aware Routing (GyTAR)* [11] was designed as a geographical routing protocol adapted to urban environments and managing the traffic conditions. A sender selects dynamically an intersection (depending on the streets connectivity) through which a packet must be forwarded to reach the destination node. Between intersections, an improved greedy approach to forward packets between two intersections is used. This latter is based on the neighbors' speeds and directions. Despite GyTAR takes advantage from the urban roads characteristics, selects robust paths with high

connectivity and minimizes the number of hops to reach an intersection; the main GyTAR drawbacks are that the connectivity information may be maintained by the infrastructures (RSSU: Road Side Service Unit) and it has a weak performances in spare networks. Besides, *Geographic and Delay Tolerant Network with Navigation Assistance (GeoDTN+Nav)* [12] proposes a solution to the last problem. It switches between DTN (Delay Tolerant Network) and non DTN mode depending on the roads connectivity. In DTN mode and for a sparse network, the vehicle uses the *carry-and-forward* scheme. The packet will be stored until finding a possible forwarder. The main disadvantage of this protocol is that the performances are affected (high latencies) in a sparse network. This is due to the fact that the protocol tries constantly to switch between the DTN and non DTN mode when forwarding packets. In [13], authors introduced a scheme which enhances the stability of Intervehicular communications (IVC) and road-vehicle communications (RVC) communications in VANET networks. The key idea behind the proposed scheme is to group vehicles according to their moving directions. [14] proposes a routing protocol adapted for mobile networks which could be used in the case of VANETs.

We have used the Greedy Perimeter Stateless Routing (GPSR) as the geographic routing protocol for the combination. However, the work still available with other geographic routing protocols.

III. HHLS: an Efficient GPRS & HLS Combination

A. Description

In order to reduce the overhead of HLS and GPSR, we combine them in HHLS algorithm. The major difference between HHLS and HLS/GPSR is implemented in three functions. The first function *Poslookup* (defined in algo. 1) handles the querying of destination's position, it looks into the local cache memory of the current node and updates the packet information with the destination's position. Then, the second function *GPSREmit* (defined in algo. 2) manages the creation and emission of new packets, it verifies at first, whether the sender has fresh or non-fresh information about the destination's position and then starts the routing of packets. If not, the function starts a new position query and places the packet into a buffer while the query is taking place. The function *forwardPacket* (defined in algo. 3) handles the forwarding of packets, it is called whenever a packet reaches a intermediary node, it verifies whether this node has a fresher position of the target and eventually updates the packet's information with it. Otherwise, if the reached node is in the same region of the destination, we must launch a new query to retrieve the new position of the target.

Algorithm 1 HHLS :: Poslookup

```
1: cacheThreshold ← LocationCacheMaxAge;
2: enum freshness {NOPOSITION = 0, FRESH = 1, NONFRESH = 2};
3: procedure POSLOOKUP(packetToSend)
4:     if (! Location information) then
5:         freshness ← NOPOSITION;
6:         PreparePacket(packetTosend);
7:     else
8:         if (Destination Location age < cacheThreshold) then
9:             freshness ← FRESH;
10:            PreparePacket(packetToSend);
11:        else
12:            freshness ← NONFRESH;
13:            PreparePacket(packetTosend);
14:        end if
15:    end if
16: end procedure                          ▷ End of Poslookup
```

Algorithm 2 GPRS :: GPRSEmit

```
1: procedure GPSREMIT(packetToSend)
2:     PreparePacket(packetToSend);
3:     Poslookup(packetToSend);
4:     if (freshness == FRESH) then
5:         ForwardPacket(packetToSend);
6:     else
7:         if (freshness == NONFRESH) then
8:             PreparePacket(packetTosend);
9:             ForwardPacket(packetToSend);
10:        else
11:            LaunchPositionQuery(destination);
12:            StickToBuffer(packetToSend);
13:        end if
14:    end if
15: end procedure                          ▷ End of GPSREmit
```

B. Experimentations

1) Working Environment: The simulations were performed using the NS-2 simulator 2.33. Greedy Perimeter Stateless Routing (GPSR) [8] is the used geographic routing protocol. The area chosen is a 2x2 km^2 of a real map representing a part of the French city *Reims*. This area is extracted from Open Street Map. The MAC layer used is 802.11p. The parameters used in the simulation are summarized in Table. I.

At each simulation, each node starts 4 CBR traffics of 10 packets with a size of 128 B to 4 random destination nodes with one second of interval between each sending instant. The CBR traffic simulates, for example, an audio or a video streaming process. It may be used in monitoring applications for example, such as viewing the video stream from a camera located on a bus by the police car or the security agent vehicle. Also, this traffic could be used in entertainment applications to connect to the Internet or to play online video games.

2) Experimentation Results: The main result in our experimentation is the number of sent location requests shown in Figure. 1. Clearly, the number of location requests is reduced in HHLS. For example, there are 93% less location requests in HHLS compared to HLS with 20 nodes. This difference decreases when the number of nodes increases(25% with 100 nodes and 22% for 120 nodes). This is due to the traffic overload when the number of nodes increases.

Algorithm 3 GPSR :: ForwardPackets

```
1:  procedure FORWARDPACKET(packetToSend)
2:      if (freshness == FRESH) then
3:          ChooseNextBestHop(packetToSend);
4:          ForwardNextHop(packetToSend);
5:      else
6:          if (Current Node freshness == FRESH) then
7:              UpdatePacketDestination(packetToSend);
8:              ChooseNextBestHop(packetToSend);
9:              ForwardNextHop(packetToSend);
10:         else
11:             if (Current Region == Target Region) then
12:                 LaunchPositionQuery(destination);
13:                 StickToBuffer(packetToSend);
14:             else
15:                 ChooseNextBestHop(packetToSend);
16:                 ForwardNextHop(packetToSend);
17:             end if
18:         end if
19:     end if
20: end procedure
```

Parameters	Value
Channel type	Channel/WirelessChannel
Propagation model	Propagation/TwoRayGround
Network interface	Phy/WirelessPhyExt
MAC layer	802.11p
Interface queue type	Queue/DropTail/PriQueue
Link layer	LL
Antenna model	Antenna/OmniAntenna
Interface queue length	512 packets
Ad-hoc routing protocol	GPSR
Location-based service	HLS
Area	2x2 km^2
Number of nodes	20, 40, 60, 80, 100 and 120
Simulation time	300 s
GPRS beacon interval	0,5 s
CBR traffic	4 x 10 packets / node
CBR packet size	128 B
CBR sent interval	1 s

Table I
THE SIMULATION PARAMETERS

As a consequence of this location traffic, there are a lot of sent packets for location updates, requests, replies, handovers, etc. This produces an extra overhead. We measured the location-based service overhead in term of bandwidth consumption. The bandwidth consumed in the routing layer is presented in Figure. 2 and the one consumed in the MAC layer is shown in Figure. 3. In all these experimentations, the bandwidth needed for the routing layer and for the MAC layer is lower in HHLS than in HLS. This is due to the efficiency of the HHLS mechanism which launches the location query as late as possible.

Our main aim is to reduce the location overhead without affecting the network performances such as the packet delivery rate (PDR) and the latency. The PDR is defined as the ratio between the CBR packets received and the ones sent. It is depicted in Figure. 4. The PDR is almost better in HHLS than

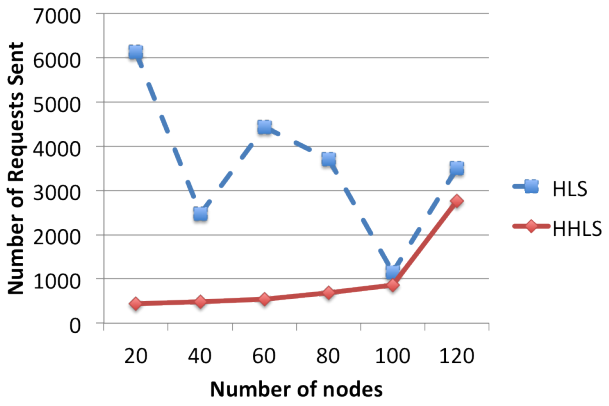

Figure 1. HLS requests Vs. HHLS requests

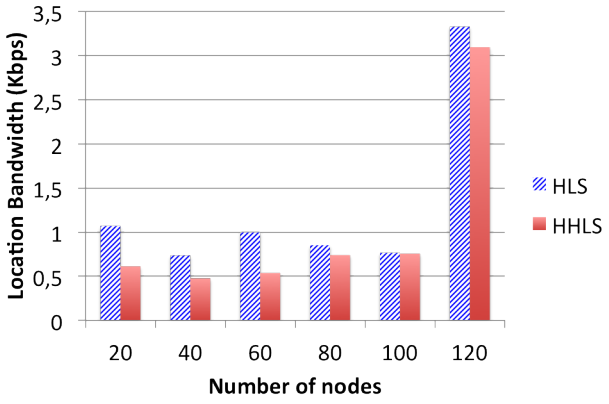

Figure 2. HLS bandwidth Vs. HHLS bandwidth in routing layer

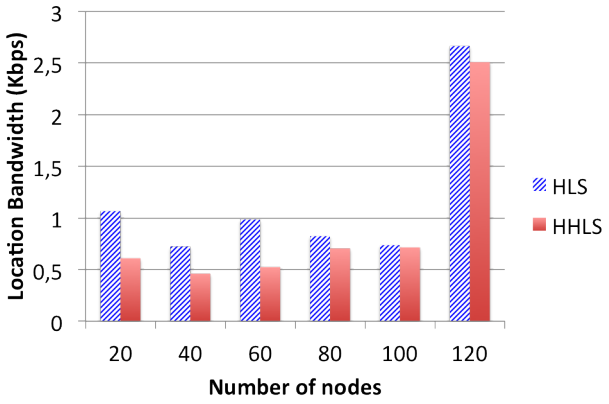

Figure 3. HLS bandwidth Vs. HHLS bandwidth in MAC layer

in HLS even it is improved by about 10% with 120 vehicles. The PDR is at its lowest rate for HLS and HHLS with 20 nodes. This could be explained by the insufficient number of nodes and hence the weak connectivity of the network. Then, the PDR increases with the number of nodes up to 120 nodes, where the network is overloaded and then the PDR decreases again because of packet collisions.

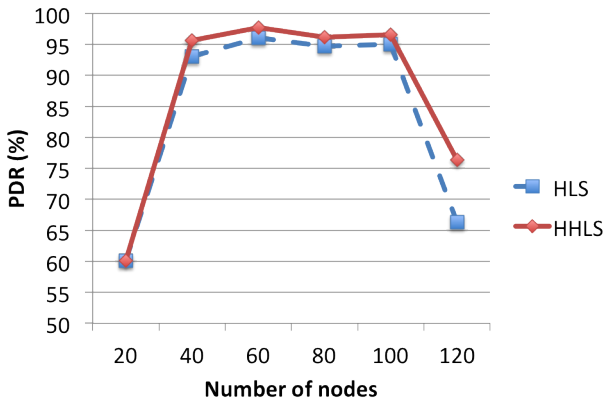

Figure 4. HLS PDR Vs. HHLS PDR

The latency is measured as the time between the moment of dispatch and the moment of receiving the CBR packets. The average latency is detailed in Figure. 5. It is lower in HHLS compared to HLS since the CBR packets are already sent even if the position is not fresh enough.

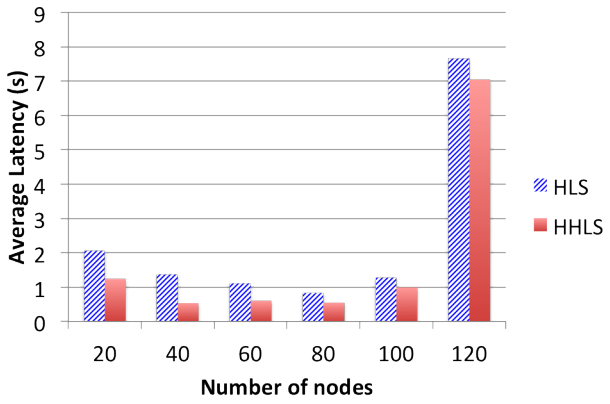

Figure 5. HLS Latency Vs. HHLS Latency

Another performance criterion is the average number of packet hops. The results of the average hops were not included because they was almost the same in both approaches HLS and HHLS.

Considering the results of these experimentations, we conclude that HHLS scheme has not only a lower cost (number of requests and consumed bandwidths) than HLS, but it enhances the network performances (PDRs and latencies). This confirms that mixing the geographical routing protocols and the location-based services reduces the overhead and also improves the performances instead of dealing with each issue alone.

IV. Conclusion & Future Works

We have proposed a hybrid approach HHLS to handle the location service and to make the routing decisions in VANETs. In fact the approach provide one unique process which includes on one hand routing and on another hand location service. We implemented it in the NS-2 framework by means of an appropriate patch. We have conducted many experimentations with this patch in order to observe network performances. We have shown in this paper that a smart combination of HLS with GPRS could provide better results in terms of network performances in particular for the packet delivery rate, the latency and the overhead. As a future work, we intend to improve these performances by adding mobility prediction. Instead of sending the packet to old locations. We will estimate the new position using additional information such as speed, direction, etc. We intend also to improve the process by considering location updates after reaching target nodes when packets are routed.

References

[1] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A distance routing effect algorithm for mobility (dream)," in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking (MobiCom'98)*, New York, NY, USA, 1998, pp. 76–84.

[2] M. Kasemann, H. Hartenstein, and M. Mauve, "A reactive location service for mobile ad hoc networks," *Department of Computer Science University of Mannheim Tech Rep TR02014*, pp. 121–133, 2002.

[3] Z. J. Haas and B. Liang, "Ad hoc mobility management with uniform quorum systems," *IEEE/ACM Transactions on Networking*, vol. 7, pp. 228–240, April 1999.

[4] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom'00)*, New York, NY, USA, 2000, pp. 120–130.

[5] W. Kiess, H. Fussler, J. Widmer, and M. Mauve, "Hierarchical location service for mobile ad-hoc networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 8, pp. 47–58, October 2004.

[6] M. Ayaida, H. Fouchal, L. Afilal, and Y. Ghamri-Doudane, "A comparison of reactive, grid and hierarchical location-based services for vanets," in *Vehicular Technology Conference (VTC Fall), 2012 IEEE*, sept. 2012.

[7] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (lar) in mobile ad hoc networks," *Wirel. Netw.*, vol. 6, no. 4, pp. 307–321, Jul. 2000. [Online]. Available: http://dx.doi.org/10.1023/A:1019106118419

[8] B. Karp and H. T. Kung, "Gpsr: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking (MobiCom'00)*, New York, NY, USA, 2000, pp. 243–254.

[9] C. Lochert, H. Hartenstein, J. Tian, H. Fuessler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," in *In Proceedings of the IEEE Intelligent Vehicles Symposium*, 2003, pp. 156–161.

[10] B.-C. Seet, G. Liu, B.-S. Lee, C.-H. Foh, K.-J. Wong, and K.-K. Lee, *A-STAR: A Mobile Ad Hoc Routing Strategy for Metropolis Vehicular ommunications*, 2004. [Online]. Available: http://dx.doi.org/10.1007/b97826

[11] M. Jerbi, S.-M. Senouci, R. Meraihi, and Y. Ghamri-Doudane, "An improved vehicular ad hoc routing protocol for city environments," in *Communications, 2007. ICC '07. IEEE International Conference on*, june 2007, pp. 3972 –3979.

[12] P.-C. Cheng, J.-T. Weng, L.-C. Tung, K. C. Lee, M. Gerla, and J. Hårri, "Geodtn+nav: A hybrid geographic and dtn routing with navigation assistance in urban vehicular networks," 5 2010.

[13] T. Taleb, E. Sakhaee, K. Hashimoto, A. Jamalipour, N. Kato, and Y. Nemoto, "A stable routing protocol to support its services in vanet networks," *IEEE Transactions On Vehicular Technology*, vol. 56, pp. 3337–3347, 2007.

[14] B. Yahya and J. Ben-Othman, "A peer-to-peer based naming system for mobile ad hoc networks," in *The 35th Annual IEEE Conference on Local Computer Networks, LCN 2010, 10-14 October 2010, Denver, Colorado, USA, Proceedings*. IEEE, 2010, pp. 821–826.