Two-level Trajectory-Based Routing Protocol for Vehicular Ad Hoc Networks in Freeway and Manhattan Environments

Floriano De Rango, Fiore Veltri, Peppino Fazio, Salvatore Marano
University of Calabria, D.E.I.S. Dept., Rende (CS), 87036, Italy
e-mail:{derango, fveltri, pfazio, marano}@deis.unical.it

***Abstract-* This paper focuses on the routing protocol issue in two important environments for Vehicular Ad Hoc Networks (VANET): Manhattan and the Freeway. A novel protocol called *Two-level Trajectory Based Routing* (TTBR) protocol is proposed. Deterministic vehicles movement permits advantage to be taken of the map info to build a specific local trajectory to reach the destination node. However, in order to offer network scalability also a high level cell-based trajectory is applied to have a coarse knowledge of the cell where the destination node is moving. Our proposal needs Peer Servers and Grid subdivision of the space. Simulation results were assessed to show the improvements and scalability offered by TTBR in comparison with other Ad Hoc networks protocols such as AODV and GPSR. Performance Evaluation was evaluated in terms of Normalized Control Overhead and Data Packet Delivery Ratio. TTBR is more performing than AODV for a high speed and high density scenario for both the Manhattan and Freeway scenarios.**

***Index term-* VANET, Trajectory based Routing, GRID, GPSR.**

I. INTRODUCTION

Mobile Wireless Ad Hoc Networks (MANET) are wireless networks that work in a decentralized way without the need for a pre-determined infrastructure. This kind of network is based on a contention-based distributed MAC protocol such as IEEE 802.11 in a Distributed Coordination Function (DCF) operative mode or other no contention -based distributed MAC, based on TDMA schemes such as [1]. When a mobile node desires to communicate with some other node that is far away, it can use other mobile nodes on the fly that fall into its transmission range to make progress towards the destination node. For this reason, a multi-hop forwarding-based scheme routing is mandatory. Many issues regarding the Ad Hoc paradigm have been presented in the literature and a lot of research has been addressed to the routing, MAC and TCP over MANET [1-4]. These networks are very flexible and they are able to auto-organize themselves so that it is possible to use them for many purposes such as rescue operations, disaster recovery, and military operations and so on. In the last few years a novel application of these networks has been found. A context of vehicular communications where mobile nodes identified as car, bus or other mobile vehicles have been considered and new research has been focused around this issue. The network that supports vehicular communications is called Vehicular Ad Hoc Network (VANET) and it represents a specific typology of Wireless Ad Hoc Network. There is a wealth of desirable applications for ad-hoc communication between vehicles ranging from emergency warnings and distribution of traffic as well as road condition information to chatting and distributed games. As a consequence many vehicle manufacturers and their suppliers are actively supporting research on how integrate mobile ad-hoc network into vehicles ([1-7]). However, the VANET networks are affected by problems related to the high mobility of the nodes and to the urban scenario leading to very strong multipath fading. In this context, traditional Ad Hoc routing approaches (e.g. AODV) are not efficient due to their lower degree of scalability, so it is very important to develop scalable protocols [8]. Many works in literature show that routing protocols based on position strategies (that is to say protocol using node geo-coordinates to forward the packets) can be very efficient in these terms. Some of these are the GPSR and GEDIR protocols described respectively in [9] and [10]. However, Lochert et al. have been proved as the position-based approach can be affected by the problem of local optimum: a packet could be forwarded to a intermediary node having neighbors closing with the destination lesser than the intermediary itself [11]. This problem can be solved making routing decision employing information on the streets and crossing points instead of node information [11]. Maihöfer et al. approached the problem differently: a cache is introduced and the packets undeliverable are stored by intermediary node until a new discovery or moved neighbor events occur (in this case a new research could allow us to forward the stored packets to destination) [12]. To solve previous problem, Niculescu et al. proposed another approach based on the trajectory concept [13]. In accordance with [13], we propose to use the map info to build the trajectory where the geographic forwarding can be adopted. On the contrary to protocols such as GPSR or GEDIR, Trajectory-based forwarding offers a better performance, because in the VANET it is possible to build a trajectory that can account for obstacles or voids that can produce a long detour of the data packet. Moreover, we subdivide the space in the grid introducing a multi level hierarchical trajectory-based routing, in this way we avoid the increasing of data packet header. In this work, we limit our analysis to a two-level trajectory-based routing: the first level is to build a specific trajectory, based on the map info distributed in the neighborhood of mobile nodes; the second level is useful to build a cell-trajectory where a coarser knowledge of the path is used. The higher level cell based trajectory uses a principle similar to Terminodes routing or to the Geo-LANMAR ([8,14]).

The paper is organized as follows: Section II gives an overview of the related work on VANET and routing protocols; in Section III the working steps of the TTBR protocol are shown; mobility models adopted in this work are presented in Section IV; trajectory based forwarding scheme (local and high level) is explained in Section V; location system and map coding are introduced respectively in Section VI and VII; Section VIII describes the Server Points calculation and Peer Servers election procedures; start-up phase of TTBR is briefly introduced in Section XI; node selection procedures and cell switching phase are summarized in Section X and XI; then simulation results are presented in Section XII; finally, conclusions are presented in the last Section.

II. RELATED WORK

Thanks to their potentialities many vehicle manufacturers and their suppliers are actively supporting research on how integrate mobile ad-hoc network into vehicles ([1-7])

VANET, contrarily to a generic MANET, present many problems associated with high vehicle mobility, including the context where the mobile nodes can move (city environment with high grade of multipath fading), or high nodal density and number of nodes involved in communication in a dense urban environment. The high mobility speed can produce frequent link breakage. This means that if a source-based on-demand routing protocol such as AODV is applied, a lot of route discovery procedures need to be started, producing a greater amount of control packet overhead. Multipath fading produces a frequent fluctuation of the channel and this means that adaptive modulation techniques with channel estimation should be applied. The high nodal density creates a lot of problems for the protocol scalability. It is important to develop protocols that are able to offer scalability in terms of state info stored in the mobile nodes, in terms of a lower number of discovery or recovery procedures and in terms of reduced traffic load for increasing traffic demand [8]. In this perspective it is important to find routing strategies that try to meet the above requirements. To this purpose a new protocol that uses advanced behavior inherited from some important protocols proposed in the last few years, is proposed. In order to offer state scalability a geographic routing scheme is applied. In this way without the need for IP packet forwarding, the use of the geo-coordinates reduces the state info stored in the routing table of each node, such as shown in many papers presented in the literature ([7,9]). In particular, in [7] a position-based routing approach is compared with non-position-based ad hoc strategies such as AODV: the authors proof that position-based approach outperforms other strategies thanks to its very high scalability. Lochert et al. deal with problem of local optimum that can occur when Greedy strategies are applied in position-based routing: this can lead to forward a packet to a node whereby there is not a neighbor which is closer to the destination than itself [11,15]. Furthermore, they show as repair strategy based on graph planarization algorithms can be inefficient and so they propose a new approach: Greedy Perimeter Coordinator Routing (GPCR) employing repair strategies in which routing decisions are made on the basis of streets and junctions (contrarily to traditional approach based on individuals nodes and their connectivity).

Yuksel et al. studied various implementation issues of Trajectory-Based Routing(TBR) for stateless routing in ad-hoc networks and proposed to use Bezier curves for defining trajectories in TBR [16]

A static-node assisted Adaptive data Dissemination protocol for Vehicular networks, which reduces the data delivery delay through three mechanisms, was proposed in [17].

Instead, in [12], this problem is approached differently: the authors introduce a small cache to the network layer in order to hold those packets that cannot be immediately forwarded due to local optimum problem. Then, when a known neighbor changes its position, the cache can check all its stored messages and it can try to forward them to a new discovered or moved neighbor in order to finally reach the destination node. The proposed protocol is called Cached Greedy Geocast (CGGC). However, in our work, to avoid the issue associated with the geo-forwarding such as GPSR, GEDIR or above mentioned approach ([9-12]), a source routing technique based on the trajectory concept is applied. A fully geographic forwarding criteria can determine some longer route due to the problem of the local maximum and to the perimeter forwarding technique such as explained in ([8,14]). For this reason it cannot be suitable for the context of VANET. In a VANET, for different node density the fully geographic routing protocol performance can degrade trough the selection of sub-optimal route. Specifically, a trajectory-based routing in accordance with work presented in [8] is considered with some modifications for the VANET environment. In VANET networks mobile nodes move on the streets and this offers a greater determinism in the link-life time calculation and on the forwarding strategies. Using the map info offered by GPS system it is possible to know specific info such as streets, crossover points, location info and so on. All this info can be applied to make a more suitable choice in route finding. We propose to use the map info to build the trajectory where the geographic forwarding can be adopted. On the contrary to protocols such as GPSR or GEDIR, Trajectory-based forwarding offers a better performance, because in the VANET it is possible to build a trajectory that can account for obstacles or voids that can produce a long detour of the data packet. Other mechanisms can be used such as those explained in ([8,14]), but for the vehicular context the map info offers other possibilities of routing strategies.

The main contributions introduced in this work are listed in the following:

- Considering the issues associated with trajectory -based routing owing to coding and insertion of the coded trajectory in the data packet, a GRID structure such as suggested in [13] is applied.
- Trajectory coding and storing can limit the protocol scalability, because for a longer path a greater number of points needs to be stored in the data header. In order to avoid the data packet header increase, a subdivision of the space in the grid is considered and a two-level Trajectory-based routing is applied.
- Our approach is general and it can be applied also to more hierarchical levels. In this case we limit ourselves to two hierarchical levels: the first level is to build a specific trajectory, based on the map info distributed in the neighborhood of mobile nodes; the second level is useful to build a cell-trajectory where a coarser knowledge of the path is used.
- The higher level cell-based trajectory uses a principle similar to Terminodes routing or to the Geo-LANMAR

([8,14]). When a node approaches destination it can use the most specific low level (map-based) trajectory. A similar approach offers higher network scalability and more longevity to the high level route. Since our proposed routing for VANET is based on two hierarchical levels, it is called Two -level Trajectory Based Routing (TTBR) protocol. This contribution is an extension of a previous our contribution [18] where more simulations and the extension to Freeway scenario is introduced and more explanation about the street map coding and mobility model are given.

III. WORKING STEPS OF THE TRAJECTORY-BASED ROUTING PROTOCOL

The working scheme of a trajectory-based routing protocol is the following [13]:

1) Each node in the network performs the start-up phase through the Hello packet exchange in order to get the info about its neighborhood; this phase permits the neighbor entry to be inserted and the Neighbour Table to be populated.
2) The next phase is the acquisition of the local cell map where the mobile node is moving. This map can be requested from some map server disseminated in the VANET through some map distribution protocol, which will not be specified in this work.
3) After the acquisition of the cell map, registration on the Peer Server (called also *Location Servers*) needs to be performed. These servers are called Peer Servers in our protocol and they are distributed on a Grid basis. This procedure permits distribution of the node location info, so that after a query packet in the network, its exact location can be found. This approach is typical of any fully geographic ad hoc routing protocol [19]. The beauty of this approach is that trajectory based forwarding is always applied to reach the Peer Servers. In this case a reference point is needed that is calculated on the Grid and the registration packet is sent towards this reference point, called Server Points, building a high level trajectory. Details of this phase will be given in the next paragraph.
4) The Peer Server, receiving the registration packet, will send back an Ack Update packet to the sender node and will broadcast the registration packet to all its neighbor nodes in order to update their Client Table. This packet forwarding is important to guarantee a greater resilience to the *Peer Server* breakdown, because more nodes near the Server Point can be used as Peer Servers. The registration phase will be completed with the reception of the Ack Update packet of the sender, which will stop the timer at the source activated during the registration phase.
5) Each node that wants to communicate with the registered node can calculate the Server Point through the Hash function, which will receive the destination node identification as an input parameter. Among the Server Points selected by the Hash Function, the nearest to the requesting node will be selected. Thus, the requesting node will start a Location Discovery Phase to build the high-level trajectory up to the Server Point, in a way similar to the Registration/Update procedure. This phase provides the forwarding of a Destination Request packet to some Peer Servers in the neighborhood of the Server Point that will send back the info stored in the Client Table in a Destination Response packet. Destination Request and Response will be always sent applying the trajectory based forwarding strategy. The high-level cell-based trajectory will not be changed until the source and destination node moves in the same cell, while the local trajectory can be dynamically changed on the traffic basis or on some other optimization criteria.

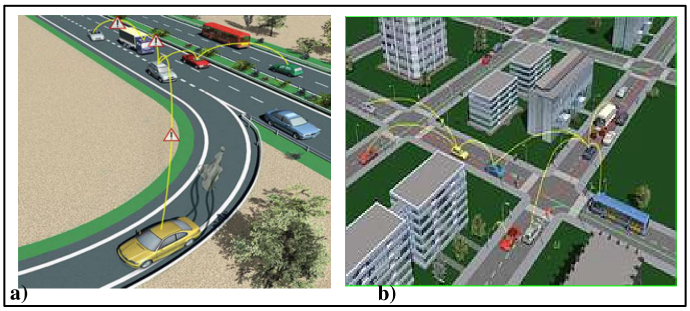


Fig.1. Freeway and Manhattan environment where VANET can operate.

Furthermore, mobility models are another important issue, because protocol performances are greatly influenced by the selected mobility model. This means that it is important to select the right mobility model for the specific conditions. The Freeway and the Manhattan scenario are respectively shown in Fig.1a and Fig.1b.

Regarding the VANET, in accordance with the work of the authors in [23], the IMPORTANT tool was applied in order to model the vehicles movement for Manhattan environment. These models are rather different from the well-know Random Way Point (RWP) mobility model because the node movement is more deterministic. Two examples of trajectories for the Manhattan and Freeway environment are presented in Fig.2a and Fig.2b.

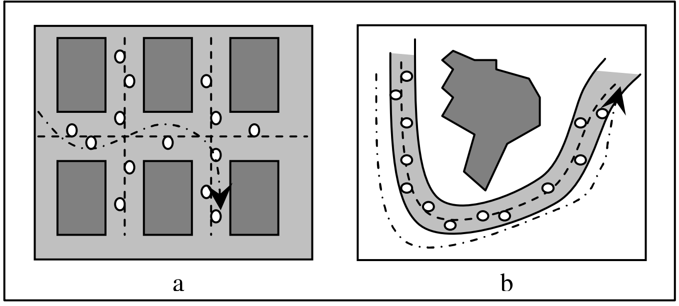


Fig.2: Trajectory based routing a) c Manhattan case; b) Freeway case

IV. TRAJECTORY-BASED FORWARDING

The path from a source to a destination vehicle, such as previously expressed, follows two phases: a specific local trajectory-based forwarding and a coarser high-level trajectory-based forwarding. In the following some indications about the building of these two trajectories for the hierarchical trajectory-based forwarding will be given.

A. *Local Trajectory Forwarding*

The local forwarding based on the local trajectory permits the most specific path to be chosen to send the data packet. This mechanism offers the possibility of selecting the path, using some optimality criteria, and of avoiding possible obstacles or holes that can be found in a position-based routing. The building of the local trajectory is realized through the acquisition of the local *map* present in the current cell where the mobile node is moving. Therefore, the local trajectory is constituted of street segment lists and cross points. The trajectory will be a reference for the mobile nodes that are involved in the data packet forwarding and the node selection will be dynamic on the basis of the metric selected to define the belonging or the proximity to the local trajectory. Each node selects the neighbor node nearest to the next point on the trajectory. The node that receives the data packet will delete the current point on the curve and will consider the next point on the curve (on the street) as a reference point for the next hop selection and so on. The step is repeated until the destination is reached, if it resides in the current cell, or if the border node in the current cell is reached (see respectively Fig.3a and Fig.3b). At this time, if the destination is not reached, the switching on the next cell is performed and the global cell-based trajectory needs to be used. The destination point in the current cell for the local trajectory will be selected using the *Neighbor Maps* list. Specifically, the point that presents the lowest traffic level permitting the neighbor cell to be reached can be selected as the local destination point on the cell border.

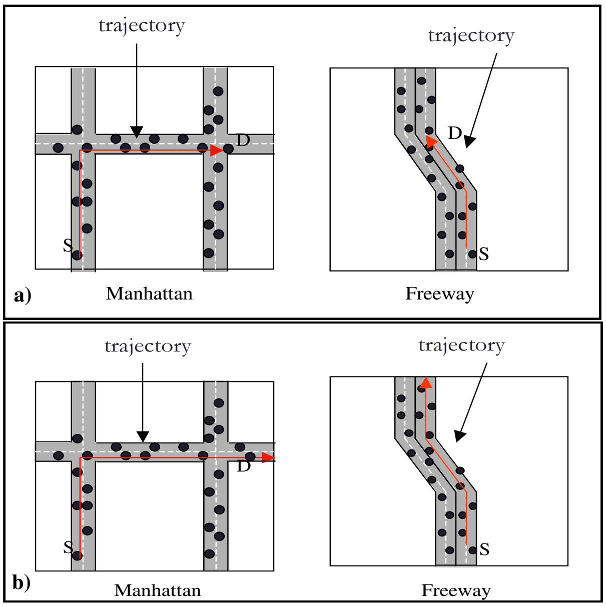


Fig.3 : a)Trajectory with source and destination in the same cell; b) trajectory until the border of the cell

B. *High-Level Trajectory forwarding*

High-level trajectory-based forwarding permits a coarse path to be found from source to destination where just the cell trajectory is specified. Since each node has information just about the map where it is currently moving, it can use the cell info that belongs to its current map. Thus, in order to know all the cells that belong to a path toward a destination, a cell discovery procedure needs to be performed. This discovery procedure is more scalable than other procedures adopted in other protocols such as AODV, DSR or more scalable protocols such as [18], because the cell discovery packet is sent in the neighbor cell to a given cell where the mobile node is moving and the neighbor cells propagate this packet to their neighbor cells and so on. When the packet propagates among the cells, it increases its size because the *cell id* is inserted in its header. However, this increase of the discovery packet is always lower than the increase of other on-demand source-based routing protocols, which insert all mobile nodes *id* that are traversed. The cell packet discovery propagates up to the destination cell (where the destination vehicle is found) is reached. The number of cells is much lower than the number of mobile nodes in the wireless ad hoc networks and this characteristic offers more scalability in the building of the global trajectory. In order to forward the cell discovery packet to the neighbor cell, the mobile node within a specific cell can use the local trajectory in the packet to reach the border nodes in the neighbor cells. After arriving at the border of the cell, the local trajectory of the discovery packet is changed by the first node in the next cell. Specifically, the border node can use its new local trajectory to reach other neighbor cells. For each traversed cell the *cell id* is inserted in the header for the high lever trajectory. This approach permits advantage to be taken of the trajectory-based forwarding in the current cell without storing a lot of points of the curve to reach the final vehicle destination, because a coarser high-level cell-based trajectory is applied. After reaching the destination cell, the first mobile node in the last cell that receives the cell discovery packet can put the last *cell id* in the list, it can invert the list in the discovery packet header and it can reverse the *cell-based path forwarding*. This approach permits the cell-based coarse trajectory toward destination to be built at the source.

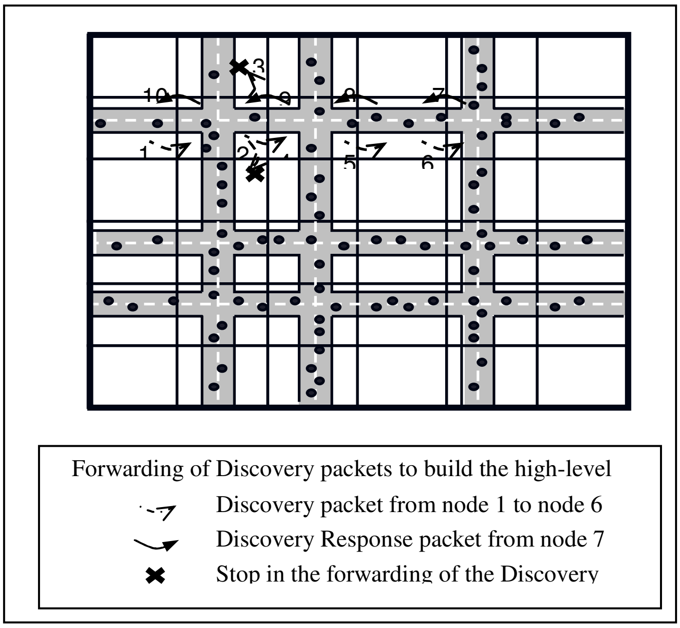


Fig.4 : Discovery Messages

The discovery procedure and messages involved in the high level cell-based trajectory building are depicted in Fig.4. The high-level trajectory could be re-built if, during the communication, source and destination node go out of the cell

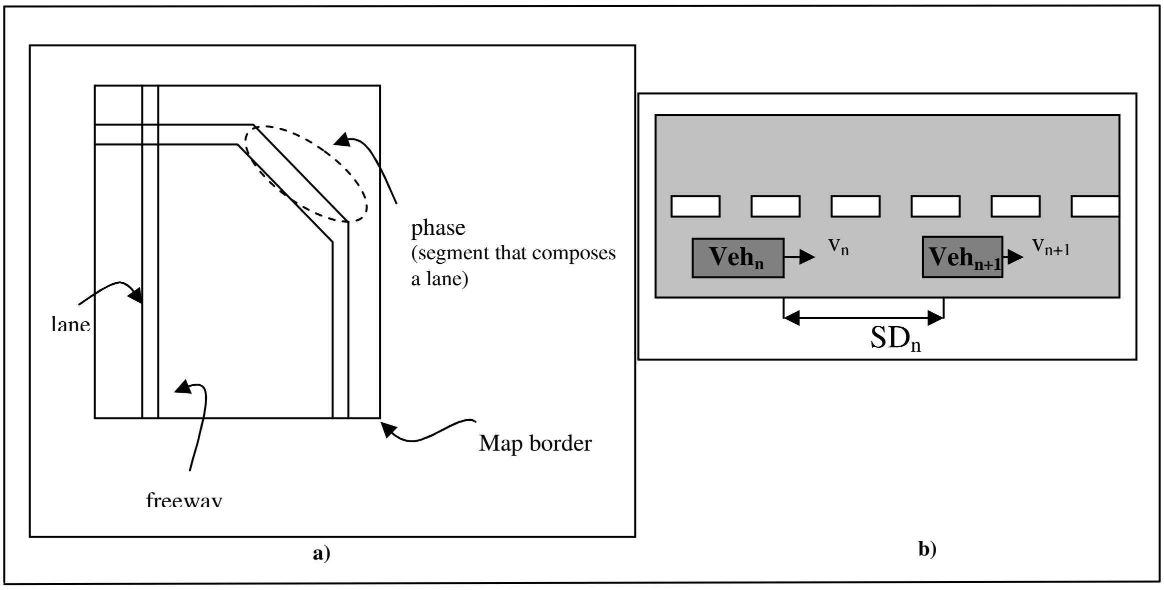


Fig.5: a) freeway mobility model; b) acceleration and deceleration associated with the security distance s.

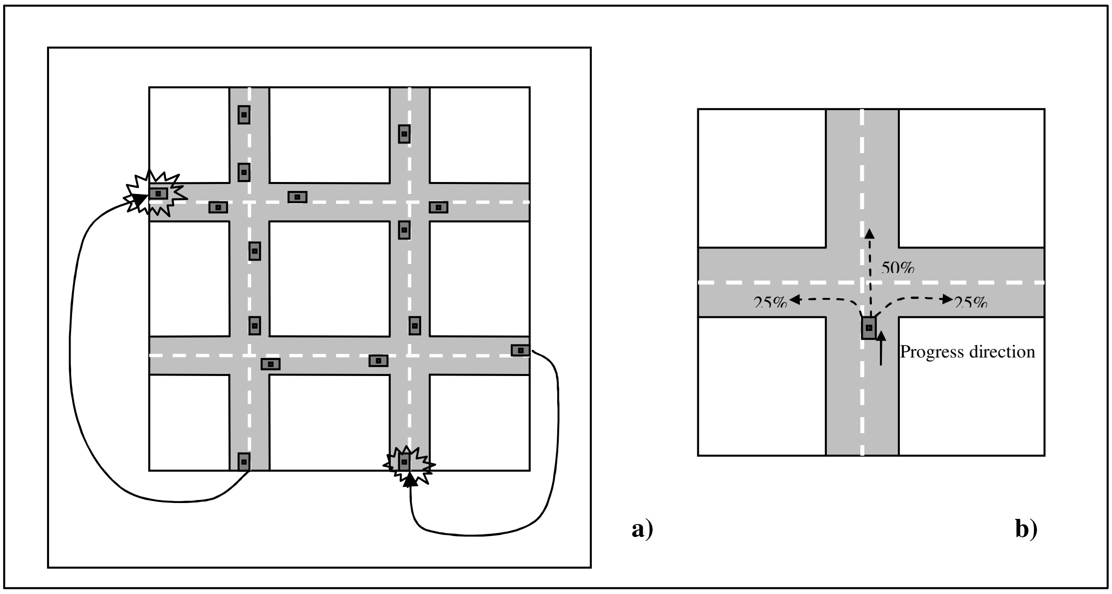


Fig.6 : a) Switching of the cell at the grid border; b)choice of the progress direction

where they are currently moving. If the source node moves out it should perform a novel *Trajectory Discovery* phase to obtain the high level trajectory. On the other hand, if the destination node is moving out it sends a control packet to inform the source of the cell change and this permits a novel Trajectory Discovery Phase to be activated. In this last case the source node does not need to send a Destination Request to the destination Peer Servers, because it knows the new cell location.

V. MOBILITY MODELS

Mobility models are an important issue, because protocol performances are greatly influenced by the selected mobility model. This means that it is important to select the right mobility model for the specific conditions. Regarding the VANET, in accordance with the work in [23], the *IMPORTANT* tool was applied in order to model the vehicles movement for the *Freeway* and *Manhattan* environment. These models are rather different from the well-know *Random Way Point* (RWP) mobility model because the node movement is more deterministic.

A. *Freeway*

A Freeway environment is characterized by long lane traversed by vehicles with high mobility speed and a few direction change frequencies such as depicted in Fig.5a and Fig.5b.

In our model for each map it is possible to have also more than one freeway. Each freeway can be composed of several lanes up to a maximum number of three lanes. The lane is represented as a segment and it is coded as expressed in Section VII.

A freeway implementation is modeled as follows: each node (vehicle) randomly selects a lane and the position in the lane; at this point the node starts to move toward the end of the lane; during the movement the mobile node can change its speed in accordance with the rule of the uniformly accelerated motion; so, an acceleration is fixed and the node velocity is temporally dependent on its previous velocity; the node needs to reduce its speed if it overcomes a *security distance* (SD) from the node that is in front of itself, because the following node cannot exceed the velocity of the preceding node. The mobility model also imposes strict geographic restrictions on the node movement by not allowing a node to change its lane. An example of the implemented Freeway mobility model is presented in Fig.3.a. For more details about this mobility model see [11,22].

B. Manhattan

The Manhattan mobility model is typical of city environments characterized by streets, corners and cross points such as depicted in Fig.6. A *map* is used in this model too. The map is composed of a number of horizontal and vertical streets (see Fig.6a). Each street has two lanes for each direction (north and south direction for vertical streets, east and west for horizontal streets). The mobile node is allowed to move along the grid of horizontal and vertical streets in the map. A node that moves on a street in the proximity of a cross point can decide to change street or to maintain its direction (see Fig.6b). In a similar way to the Freeway case, the node can change its speed or stop on the basis of the obstacles (other vehicles or traffic lights) that are met during the movement.

The Manhattan mobility model is implemented as follows: a mobile node randomly selects the street and consequently the movement direction on the street; then, the mobile node starts to move toward the next cross point according to a uniformly accelerated motion; when a cross point is reached a direction of movement (it can be the same or can be changed) is selected through a uniform distribution. The probabilities associated with direction change are respectively 50% to maintain the same direction, 25% to change in the other two possible directions such as depicted in Fig.4b. When the node reaches the end of the GRID, another street is randomly selected such as presented in Fig.4a and it is considered as a new node that enters in the network. For more details about this mobility model see [9-11]. The Manhattan mobility model is also expected to have high spatial dependence and high temporal dependence. However, the Manhattan model differs from the Freeway model in giving a node some freedom degree to change its direction.

VI. LOCATION SYSTEM

In order to send a data packet toward destination it is important to know the destination position for any location-based routing strategy. This means that a Location System is an important issue. In this work a Grid Based Location Service was applied such as in [20,21]. The adopted Location System is based on the Grid concept that permits subdivision of the space in further cells. Specifically, each region is divided into first level grids. These grids can be subsequently divided into second level grids or cells and so on (see Fig.7). So a hierarchical grid structure can be used such as referred to in [22]. This system permits to know where the mobile node is positioned and it also permits a mobile node to be registered at the Peer Server through the cell location info. For our purpose it is not necessary to know the exact position of the destination node, but the current destination cell where the mobile node moves. This is owing to the coarse knowledge that can be applied in long distance trajectory forwarding in order to offer network scalability. Only when the data packet arrives in the last cell, more specific local trajectory forwarding is applied. The coarse knowledge of the destination location and the registration of the destination cell rather the destination node position permits reduction of the high control overhead owing to the location updating of node on the *Peer Servers*. An enhanced version of the GRID Location Server (HIGH-GRADE) is considered in accordance with [21].

In this work, we proposed a two-level hierarchy, in which the first-level grid is applied to fix in the space the *Server Points* where to elect the location servers called *Peer Servers* in our protocol. In the next paragraph an example of the Server Points selection in the first-level grid is presented. On the other hand, the second-level grid permits subdivision of the streets map into small maps for each sub-grid and in this way map knowledge of the specific streets and cross points in the neighborhood of a given node can be distributed. This work does not address the issue of map distribution, but it assumes that the mobile node can know the local map in each sub-grid to build a local trajectory toward the border of the current cell. This assumption can be made because it is possible to think to *map distribution* protocols or to the *map server* election procedures usage, where the *map info* are distributed and requested by mobile nodes. We are not interested in the time and overhead introduced by the *map distribution,* but we want to see the advantage gained after the acquisition of the *local map*. However, it is important to observe that the sub-map size is maintained low, because this can produce a lot of overhead and can increase the interference at the MAC layer if a contention-based MAC such as IEEE 802.11 is applied. The *Server Points* selection that represents virtual geographical locations in the space are useful to elect more *Peer Servers* (*Location Servers*) distributed in the space. This can guarantee reliability if a *Peer Server* fails and the query propagation range can also be reduced to get the destination node position info or destination cell location.

VII. CELL MAP

The *map* represents a set of streets and cross points that the mobile node can potentially visit. Differently from the general ad hoc networks, where the movement of a node can follow a random direction such as the RWP model or other stochastic mobility model, in this case the vehicle (mobile nodes) can move following the street where is located. In this perspective it is possible to see the importance of knowledge of the streets and cross points in order to make the best choice of trajectory points. The map that is composed of streets and crossing points needs to be coded in parallel and/or crossing segments where the coordinates of the initial and end point of the segment is stored. It is possible to use one or more packets to put this information, which can be distributed through some map distribution protocols. In this work the local map knowledge is supposed and the local map is coded in the

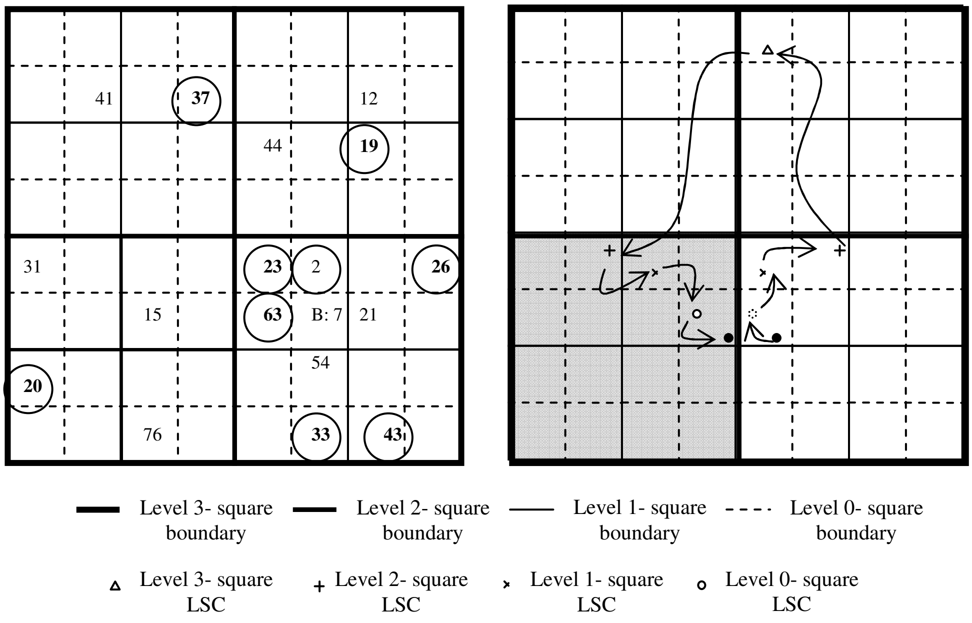
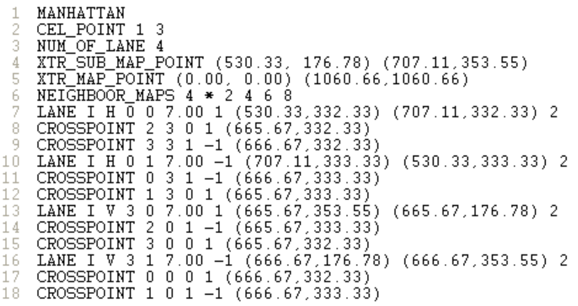


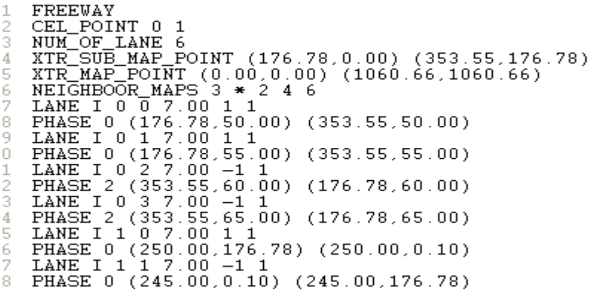
Fig.7: Grid and High Grade Location Services.

mobile nodes on the basis of the mobility model applied for the simulations. Two map types are presented in the following where the same coding criteria are applied. The map files will be coded in text format and they will be loaded onto each node that belongs to a specific cell. A problem of this approach can be the frequent MAP distribution of the MAP, especially in the case of the movement around the cell boundary of the mobile user. In this case the problem can be reduced through a correct dimensioning of the cell sizes in order to reduce the cell change frequency. Another approach can be the use of further hierarchical levels introduced to cover different details of the *map* at different levels of the hierarchy. Anyway, this aspect is not addressed in this work and it can be seen as a future research issue in the vehicular ad hoc networking field.



```
MANHATTAN
CEL_POINT 1 3
NUM_OF_LANE 4
XTR_SUB_MAP_POINT (530.33, 176.78) (707.11,353.55)
XTR_MAP_POINT (0.00, 0.00) (1060.66,1060.66)
NEIGHBOOR_MAPS 4 * 2 4 6 8
LANE I H 0 0 7.00 1 (530.33,332.33) (707.11,332.33) 2
CROSSPOINT 2 3 0 1 (665.67,332.33)
CROSSPOINT 3 3 1 -1 (666.67,332.33)
LANE I H 0 1 7.00 -1 (707.11,333.33) (530.33,333.33) 2
CROSSPOINT 0 3 1 -1 (666.67,333.33)
CROSSPOINT 1 3 0 1 (665.67,333.33)
LANE I V 3 0 7.00 1 (665.67,353.55) (665.67,176.78) 2
CROSSPOINT 2 0 1 -1 (665.67,333.33)
CROSSPOINT 3 0 0 1 (665.67,332.33)
LANE I V 3 1 7.00 -1 (666.67,176.78) (666.67,353.55) 2
CROSSPOINT 0 0 0 1 (666.67,332.33)
CROSSPOINT 1 0 1 -1 (666.67,333.33)
```

Manhattan MAP coding



```
FREEWAY
CEL_POINT 0 1
NUM_OF_LANE 6
XTR_SUB_MAP_POINT (176.78,0.00) (353.55,176.78)
XTR_MAP_POINT (0.00,0.00) (1060.66,1060.66)
NEIGHBOOR_MAPS 3 * 2 4 6
LANE I 0 0 7.00 1 1
PHASE 0 (176.78,50.00) (353.55,50.00)
LANE I 0 1 7.00 1 1
PHASE 0 (176.78,55.00) (353.55,55.00)
LANE I 0 2 7.00 -1 1
PHASE 2 (353.55,60.00) (176.78,60.00)
LANE I 0 3 7.00 -1 1
PHASE 2 (353.55,65.00) (176.78,65.00)
LANE I 1 0 7.00 1 1
PHASE 0 (250.00,176.78) (250.00,0.10)
LANE I 1 1 7.00 -1 1
PHASE 0 (245.00,0.10) (245.00,176.78)
```

Freeway MAP coding

VIII. SERVER POINTS

In order to offer location services we need *Peer Servers* that can be distributed among the network through a Grid-based criterion. The selection of *Peer Servers* is performed through the fixing of some reference points called *Server Points*. These Server Points are located through a Hash function. Specifically, in this work a Hash function was applied to calculate four server points in the first-level grid such as depicted in Fig.8, Fig. 9 and Fig.10. The first grid (level 0 grid) is divided into four square regions where 4 Server Points are calculated. Each node will have different Server Points in the Square region and this permits different Peer Servers to be selected in accordance with the work presented in [21].

After the selection of four *Server Points*, some nodes have to be elected as *Peer Servers* and then the mobile node can start a Trajectory Discovery phase to find the high level trajectory. In the following, three procedures for the Peer Servers election are presented: Server Point calculation, Peer Server election and Peer Server registration.

A. *Server Points Calculation*

The server points calculation is realized through a Hash function that is able to distribute the server points in the space on the basis of a unique node identifier as input parameter. Using this function each node is able to obtain, in our specific case, four geo-coordinates that represent the four Server Points that will be used by a node to make the registration.

As previously referred, the first-level grid is applied to collocate the four Server Points such as depicted in Fig.8. The sbc_x and sbc_y variables that are used in the hash function are referred to the central point in the quadrant in which the region has been subdivided through the first-level grid such depicted in Fig.8.

The applied hash function is presented in the following equation:

$$\boxed{\begin{aligned} sp_x &= \left[\left(\frac{node\_id}{rx} \right) * 10 \right] + sbc_x \\ sp_y &= \left[\left(\frac{node\_id}{rx} \right) * 10 \right] + sbc_y \end{aligned}} \qquad (1)$$

where *node_id* is a unique node identifier and *rx* is the range in which locate the locations servers of this mobile node. This approach permits to select four geographical coordinates for each node (Server Points) that is translated by a certain amount with respect to the middle of the cells (sbc_x , sbc_y) with i = 1,2,3,4.

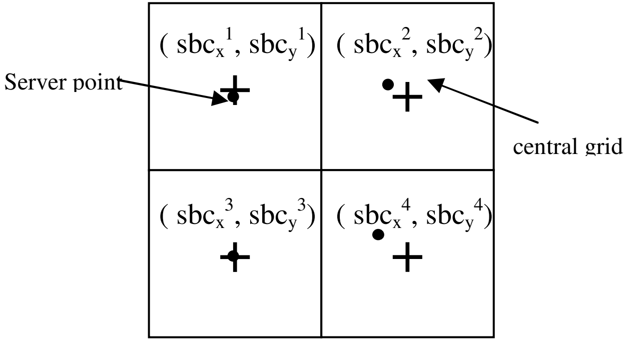


Fig.8 : Four server points.

B. Peer Server Election

Peer Server election is based on the location of a generic mobile node in the grid. Specifically, no qualitative or resource-based criterion is applied in the selection, but just the distance of the mobile node from the Server Point in a specific time instant is applied.

When a node needs to start a registration phase, it makes use of the hash function to know the Server Point and it sends the registration packet toward this geo-coordinate. However, the Server Point is only a reference point and it is possible that it is not associated with a specific node. Thus the reference point is just useful to represent a region where some *Peer Servers* can be elected. In this specific case, a criterion based on the distance between the reference point (Server Point) and the mobile nodes within the cell where the reference point is fixed is applied. All the nodes that verify the condition expressed in (2) are elected to be Peer Servers:

$$\sqrt{(x_{SP} - x_N)^2 + (y_{SP} - y_N)^2} < r \qquad (2)$$

where (x_{SP},y_{SP}) and (x_N,y_N) are respectively the coordinates of the *Server Point* and the mobile nodes in the same cell; *r* is threshold value to consider a mobile node a possible candidate to the *Peer Server* election.

Thus after the range definition where it is possible to elect *Peer Servers*, inside this range each node that receives the registration packet coming from the requesting node can store the location info and can send back a reply toward the source node. The elected Peer Servers continue to be Location Servers until a Peer Server goes out of the preset range. In this case the Peer Server transfers a part of its Client Table, trough a control packet, to other nodes in region *A,* as referred in Fig.9, in order to find another Peer Server.

The first node in the region that receives this packet adds client entries stored in this packet and it absolves the function of a novel *Peer Server*.

C. Peer Server Registration

The registration of mobile nodes on the Peer Servers is the last phase before communication between source and destination vehicles can start. In order to avoid an increase in the control overhead, a reactive scheme regarding the registration phase has been adopted. Specifically, after the Server Point election the mobile node sends a Registration/Update packet toward the Server Point when it needs to send a data packet toward a destination of which it does not know the location. The local forwarding toward the Server Points is made through the local trajectory-based routing and each node that receives the Registration/Update packet checks whether it is a Peer Server or a common node. Each intermediate node that is not a Peer Server sends the control packet on the local cell until the cell border is reached and then a cell switching procedure is applied. The local trajectory forwarding is based on the reference point stored in the control packet sent on the network. Thus control and data packet present two lists in their header: local trajectory and global trajectory. The local trajectory is represented through a list of segments (streets and crossing points) that can be traversed by packets. The global trajectory is constituted of a set of cell *id*s (*row*, *col*) that should be traversed by the control or data packet. During the cell switching phase the local trajectory is updated with the new local trajectory of the next cell and the global trajectory is increased with the new cell id of the visited cell. When a Peer Server is met in the propagation of the Request/Update packet, it stores the new location info or updates the previous stored location and it sends back an acknowledgment packet to the sender and to the other Peer Servers. A typical Client Entry that is created or updated is of the following type (*client_ip, cel_row, cel_col , sp_x, sp_y, node_type*) and an example of some values of these entries are listed in Table I, where the *node_id* is the node identification, cell_*row* and cell_*col* are useful to identify the cell, *Sp_x* and *Sp_y* are used to know the Peer Server location and node type indicates if the node is a mobile node or an access point (fixed node).

TABLE I : CLIENT TABLE OF A NODE.

Node_id	Cel_row	Cel_col	Sp_x	Sp_y	Node_type
0	**0**	**0**	**410.9**	**320.1**	**1**
2	**1**	**1**	**430.2**	**298.5**	**0**
3	**2**	**0**	**430.2**	**320.1**	**0**

IX. START-UP PHASE

In order to use the routing scheme to forward the data packet toward a destination, a start-up phase to populate the

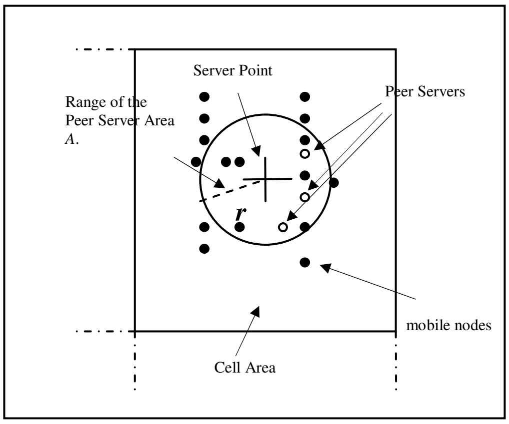


Fig.9: Peer Servers.

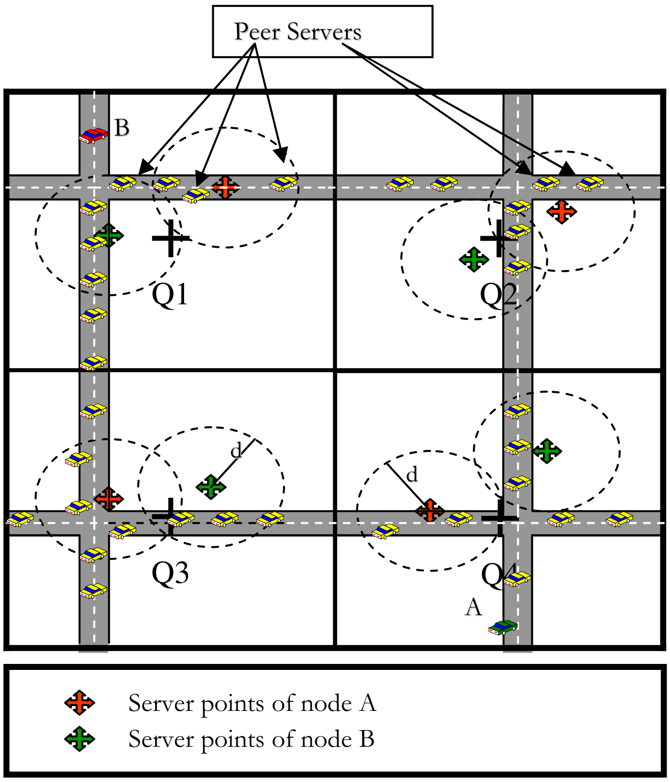


Fig.10: Peer Server distribution on the level 0 GRID. Each node uses a set of Peer Server associated with the specific Server Points.

routing table needs to be performed. Specifically, the *Neighbor table* and the *Client Table* have to be built before using the local and global trajectory to forward the data packets. The first step is the exchange of the *Hello* packet to populate the *Neighbor Table*. This phase is not so expansive because the control packet forwarding is limited to the range of the local neighborhood.

In the *HELLO* packet it is important to insert the node location info in order to use the trajectory based geo-forwarding later. An example of the *Neighbor Table* is given in Table II.

Table II. Neighbor Table of a mobile node 1.

Node_id	**X**	**Y**	**Node_type**
0	**320.1**	**430.2**	**1**
2	**430.2**	**298.5**	**0**
3	**430.2**	**320.1**	**0**
4	**320.1**	**298.5**	**0**
5	**430.2**	**298.5**	**0**

The sending rate of the *HELLO* packet can follow different rules by which the speed and position of mobile nodes can be also used in order to offer a dynamic sending rate associated with the mobility grade of the network (i.e. low frequent sending rate for low mobility and more frequent hello packet exchanges for high mobility). However, in this work a fixed timer-based approach is applied. After the building of the *Neighbor Table*, it is important to get the local *map* in order to build the local specific trajectory. In this case the mobile node can ask for the local map to the nearest neighbor in its cell. Details about the *map acquisition* or *map distribution* protocols will not be given in this work because we are interested in exploiting the *map* usage and the cell-based forwarding when this info is obtained. It is just pointed out how the mobile node can ask about the local *map* to some neighbor, but it can get an answer if the neighbor does not have this info. In this case it is possible to extend the research of the local *map* by making more trails. Obviously, the better the deployment of an efficient *map* distribution scheme, the lower the time to get this information. Another important consideration is associated with the *map size*. The greater the map size, for example, because there are a lot of streets and crossing points or because the grid size is larger, the more packets where the *coded map* is inserted need to be transmitted. In this case it may be possible to adopt some scheme where just a partial info about the MAP can be given in order to reduce the excess of control overhead, the MAC collision and to reduce the latency time in getting the entire *map*. These issues are not the object of this work and can be seen as further aspects to investigate for future research staff. In our protocol the assumption is to have the local MAP and we try to delineate how this info can be efficiently used.

X. PEER SERVERS-BASED NODE SELECTION

In order to reach the destination node two Route Discovery procedures should be preformed (see Fig. 11). The first one is to get the destination location and make the Peer Server registration. Since the Peer Server registration can involve just some *Peer Servers* that can be also far from the destination, it is important after getting the destination location, to perform another on-demand cell-based trajectory discovery in order to build the most updated destination location and destination cell. During this second Route Discovery phase, the local trajectory-based forwarding is

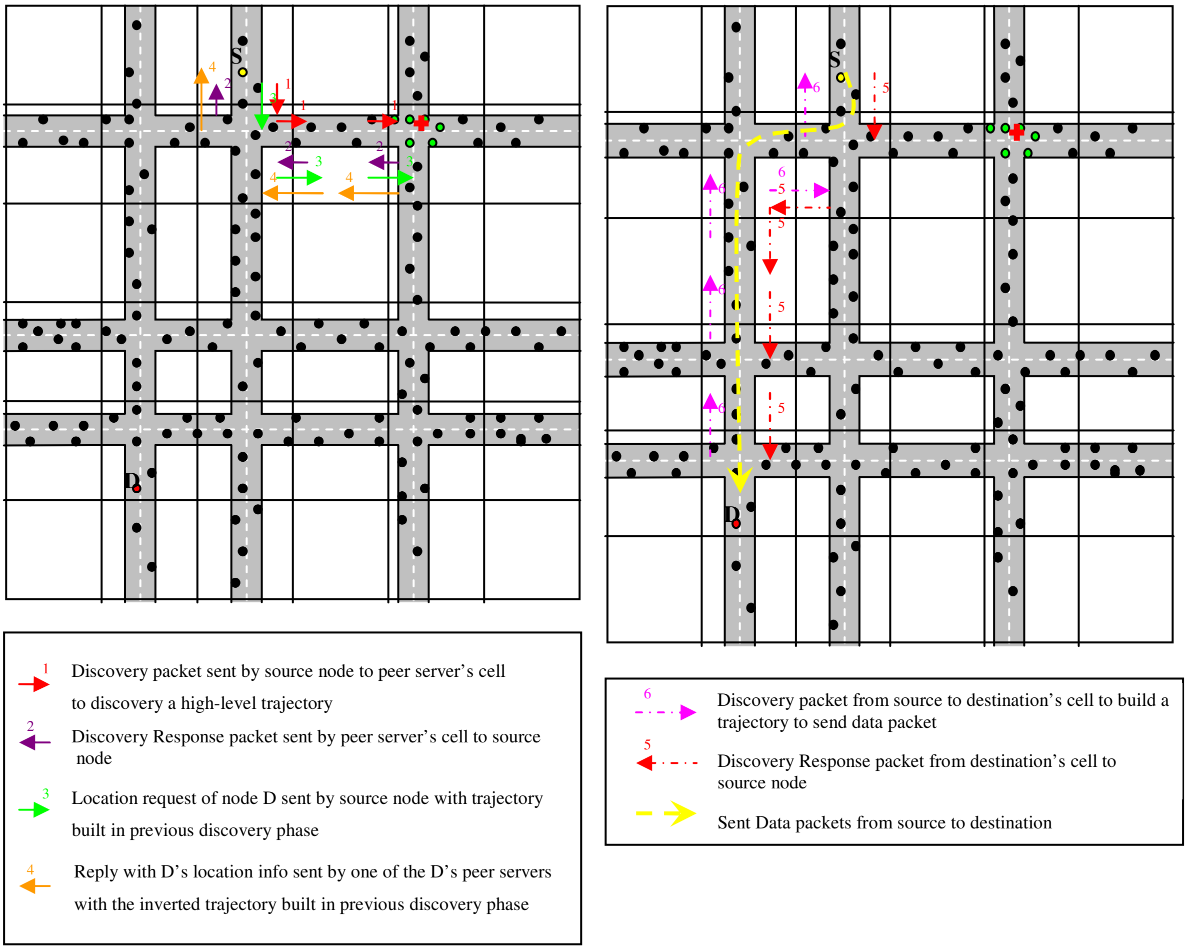
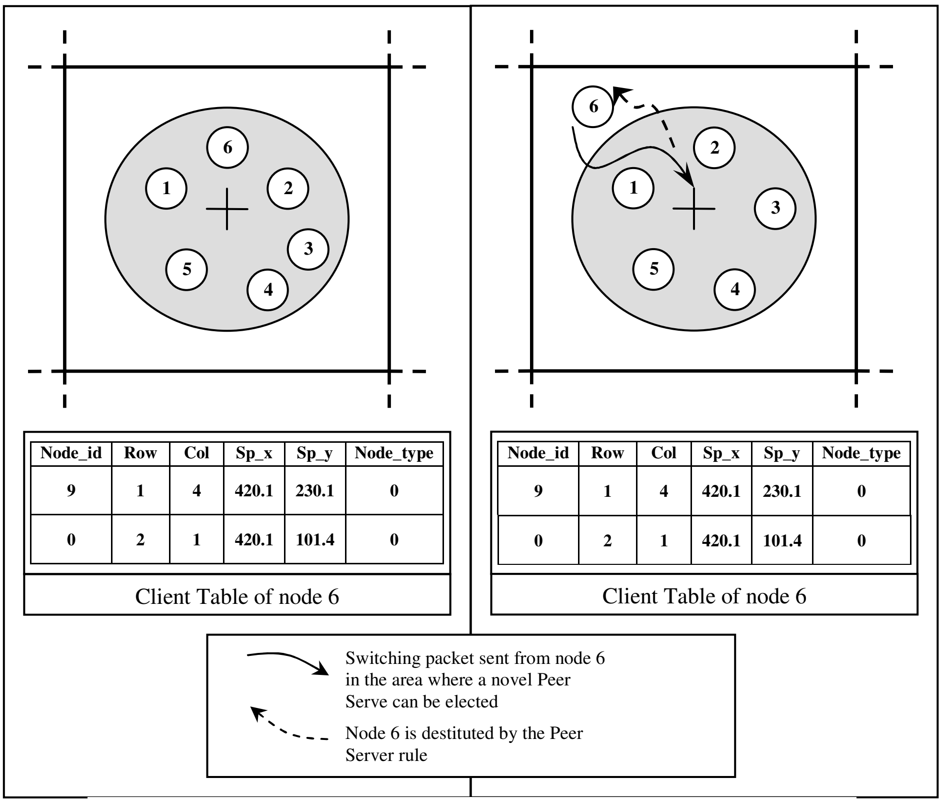


Fig.11: Forwarding of Discovery packets to build high- level Trajectory.



Node_id	Row	Col	Sp_x	Sp_y	Node_type
9	1	4	420.1	230.1	0
0	2	1	420.1	101.4	0

Client Table of node 6

Node_id	Row	Col	Sp_x	Sp_y	Node_type
9	1	4	420.1	230.1	0
0	2	1	420.1	101.4	0

Client Table of node 6

Fig.12 : Client Table inheritance from mobile nodes within the Peer Server Area.

applied and the traversed cells are inserted in the route discovery header in order to build the cell-trajectory for the source node. When the global cell-trajectory is built, the data packet is forwarded following the trajectory-based routing scheme until the destination cell is reached. Then again a local trajectory scheme is applied to reach the final destination in the best possible way.

XI. MAINTENANCE AND CELL SWITCHING PHASE

After the registration phase, maintenance and cell switching procedure need to be performed. The maintenance procedures are summarized below:

- Selection of the no more valid entry in the *Client Table*.
- Fragmentation of the Location Request if the entry in the table cannot be inserted in one packet.
- Forwarding of the partial *Client Table* to the neighbor so to elect a new Peer Server with the consistent location info.
- Erasing of the selected client entries.

The choice of the obsolete Client Entry is made considering the values storied in the *sp_x* and *sp_y*. These last values represent the Server Point coordinates associated with the entry. In Fig.12, node 6 goes outside the Peer Servers Area and it sends its table to the neighbor in the Peer Server Area. This assures that the info stored by a Peer Server continue to be manage in the network also if the Peer Server dynamically changes.

XII. SIMULATION RESULTS

An extensive simulation campaign was performed by using a software tool based on the Network Simulator (NS-2) [24]. NS-2 has been integrated with the IMPORTANT mobility tool, available at [25], in order to create the Manhattan scenario. Furthermore, in this work the 802.11 MAC was employed (for this purpose, the recommended patch was applied to N-S2 802.11 MAC implementation [26]). In the Manhattan model the number of Access Points (APs) situated on the streets are changed. An APs number in the range [10-40] is selected to create a lower or more stable scenario. The node speed is also changed in the range [10-90] Km/h and the number of nodes is increased from 50 to 250 in order to test the protocol performance in a high density and dynamic urban scenario. The level 0 grid is 2000mx2000m and it is divided into four grids at level two. A set of maps where the streets, lane and cross points are coded is distributed among the mobile nodes. The modified GRADE location server [21] is applied in order to get the destination info of the mobile nodes. Cell sizes are considered to be $d = 2r/\sqrt{2}$ (where r is the transmission range) that, in accordance with [21], permits mobile node to reach any other node at the cell border. The packet size has been fixed to 512bytes. Finally, each point in the curves is averaged over 30 runs all fallen in a 95% confidence bound.

A. Manhattan environment

The NCO, calculated as the ratio between the number of received control packets and the number of received data packets vs. number of nodes is depicted in Fig. 13. It is possible to see the increase of the control traffic when the number of nodes increases. This is owing to the higher number of MAC collisions associated with the transmission of data and control packets. AODV reaches channel saturation before TTBR, because the broadcast-based Route Discovery is resource consuming. The GPSR, not shown in the figure, presented also a reduced control overhead due to the local beacon messages exchange. In particular a lower overhead (around 5% less) has been found in comparison with TTBR. However, the data packet delivery ratio resulted lower than TTBR and the average and-to-end delay higher such as explained in the following. Also if the VANETs are not interested in energy saving, it is important to reduce the control overhead to reduce the data packet delivery latency and the bandwidth consumption. On the contrary, TTBR is more scalable and it drastically reduces the control overhead guaranteeing an improvement of 60% in comparison with AODV. It is possible to see also how the TTBR is better performing for higher speed (10-50 Km/h). In this case and when the number of the node is high (>100) the improvement of TTBR is more accentuated. This is owing to the better network state management of the TTBR. The Trajectory usage and the Grid Location Server offer more scalability and reduce the number of control packets. Experiments were conducted also changing the number of APs positioned on the street of the Manhattan scenario (see Fig.14). In this case we wanted to test the protocol performance when the topology and network connectivity changed. We observed an improvement of AODV and TTBR when a higher number of APs is applied. This is owing to more stable paths that can be selected when the APs are applied. If a higher number of fixed nodes is selected, the path is more stable because few routes will be composed of mobile nodes that can move also in the opposite direction on the street (very low link life time). Anyway, the TTBR continues to perform better than AODV but the difference is not marked. This means that the high mobility grade impacts differently if some fixed nodes can be applied. Anyway, the differences between AODV and TTBR are great when a high number of nodes is involved in communication, such as depicted in Fig.13. Thus. also if some criteria to select a stable route can be applied when more fixed nodes are present in the network, this aspect does not assure a good protocol performance if the nodal density or network size increases. It is important to offer network scalability and TTBR deals with this issue. Data Packet Delivery Ratio, for the TTBR, AODV and GPSR protocols, is presented in Fig.15, Fig.16, Fig.17 and Fig.18 respectively as function of maximum vehicles speed (the first two) and APs number in a high populated scenario (the last two figure). We must underline that the increasing number of nodes significantly impacts on the Data Packet Delivery Ratio reduction both for the AODV and GPSR protocols specially in presence of high mobility. In particular, for the AODV, this is owing to the bandwidth wastage owing to the high control overhead. TTBR, on the other hand, is better performing, because it puts lower control packets and it can make use of the local map-based and long-distance cell-based trajectory that takes advantage of the street layout and node movement, so the performance differences between three protocols are emphasized in presence of a great number of nodes (that is the scenario depicted in Fig.14 and Fig.17). Higher mobility speed produces decreasing delivery ratio (see Fig.16). This is owing to the frequent link breakage and to the high number of path discovery procedures. Also in this case, TTBR is able to delivery over 70% in comparison with AODV (30%) and GPSR (40%) because it limits the control overhead to the Location Discovery on the Peer Servers and to the Trajectory coded in the data packet. However, the trajectory coding impact is reduced through the cell-based trajectory forwarding. Data Packet Delivery Ratio for increasing number of APs is presented in Fig.17 and Fig.18. In this case, the higher APs number offers a more stable path, increasing the Data Packet Delivery rate. Similar behavior in the curve is observed for AODV, GPSR and TTBR. However, TTBR offers a high throughput for its better capability to exploit the map and cell knowledge. Average end-to-end data packet delay has not been presented for space limitation. However TTBR outperformed AODV (around 60% lower) and also GPSR (around 30% lower). In particular GPSR, also presenting a lower control overhead, is not able to select always the shortest route due to the perimeter forwarding techniques that is applied when the local maximum is found such as referred also in [8]. TTBR, instead, is able to better manage the map info and to select better route reducing the void. Higher mobility speed produces decreasing delivery ratio. This is owing to the frequent link breakage and to the high number of path discovery procedures. Also in this case, TTBR is able to delivery over 70% in comparison with AODV because it limits the control overhead to the Location Discovery on the Peer Servers and to the Trajectory coded in the data packet. However, the trajectory coding impact is reduced through the cell-based trajectory forwarding.

Data Packet Delivery Ratio for increasing number of APs is presented in Fig.16. In this case, the higher APs number offers a more stable path, increasing the Data Packet Delivery rate. Similar behavior in the curve is observed for AODV and TTBR. However, TTBR offers a high throughput for its better capability to exploit the map and cell knowledge.

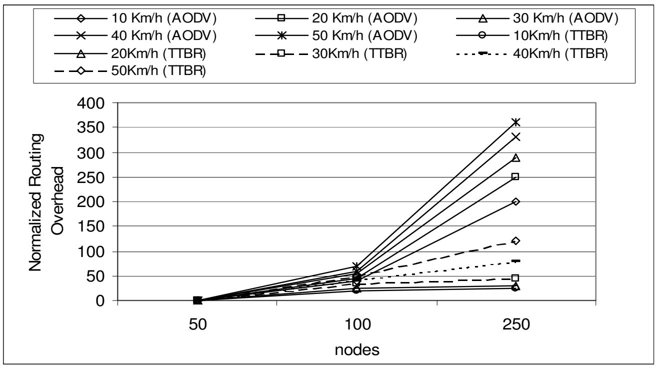


Fig.13: Normalized Control Overhead v Number of Nodes in Manhattan scenario.

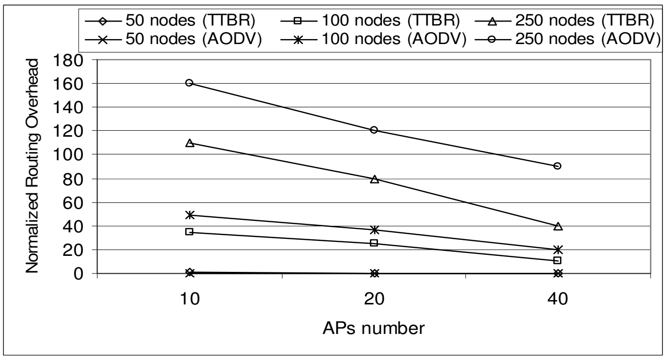


Fig.14: Normalized Control Overhead v increasing APs number in Manhattan scenario.

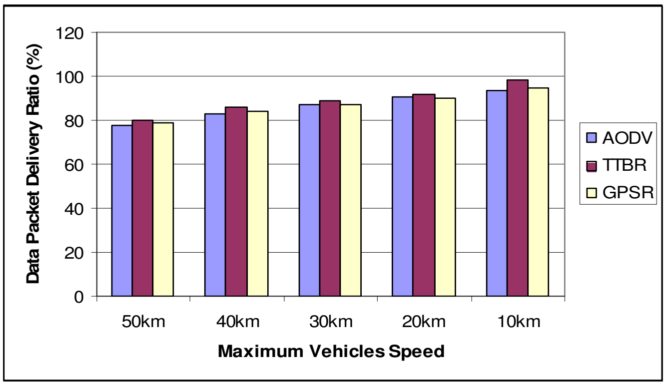


Fig.15: Data Packet Delivery Ratio v Number of Nodes in Manhattan

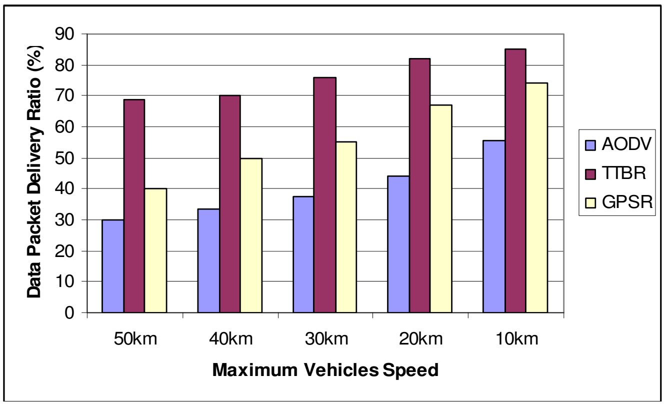


Fig.16: Data Packet Delivery Ratio v Number of Nodes in Manhattan scenario.

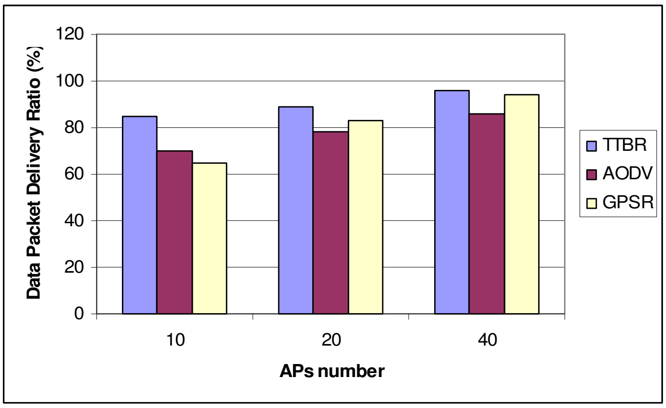


Fig.17: Data Packet Delivery Ratio v Number of Nodes in Manhattan scenario.

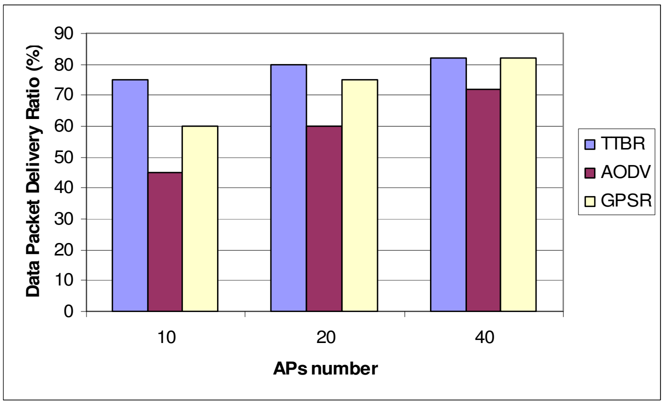


Fig.18: Data Packet Delivery Ratio v Number of Nodes in Manhattan scenario.

B. *Freeway Environment*

Normalized Control Overhead is presented in Fig.19 for the Freeway case. An increase of the control packets sent within the network is observed for a high number of nodes. Also in this case TTBR performs better than AODV. The higher speed of mobile nodes increases the control overhead too. However, the TTBR maintains its control packet overhead under the AODV control packets. The Freeway environment is different from the Manhattan scenario, because the direction of the nodes changes less frequently. This determines a reduction of TTBR control overhead because it makes use of the map info to perform the high level and local trajectory forwarding.

The Normalized Control Overhead is also evaluated for a different number of lanes (2,3,4) such as depicted in Fig.20. The higher is the lane number, the lower is the control overhead for both AODV and TTBR, because more paths can be found to reach the destination and a better load distribution is performed after the route selection.

In a similar way, the Data Packet Delivery Ratio is observed in the Freeway mobility scenario. TTBR outperforms AODV as expected because the map-based trajectory forwarding is more effective in the path finding. This behavior is more accentuated for higher speed and a higher number of nodes.

It is possible to conclude that an IP-based forwarding such as AODV is low performing in VANET especially for high mobility and high density because of the high link breakage frequency and of the greater state info storage in the routing table. On the other hand, TTBR based on the geo-forwarding paradigm and on the trajectory-based routing permits the network state info storage to be reduced and to take advantage of the map info to build a route able to avoid obstacles or voids and to reach the destination vehicle in the best way. TTBR, differently from AODV, does not perform frequent Route Discovery reducing the control overhead, but it limits its discovery to the Peer Servers. Furthermore, the Trajectory usage permits a stable path found on the map info basis to be maintained and mobile nodes nearest to the curve to be selected dynamically avoiding the building of a route. The grid space subdivision permits a specific short-distance network state knowledge to be separated by a coarse long-distance network view. This approach followed by other protocols such as [8,14] offers high scalability in terms of network size, traffic load and node mobility grade.

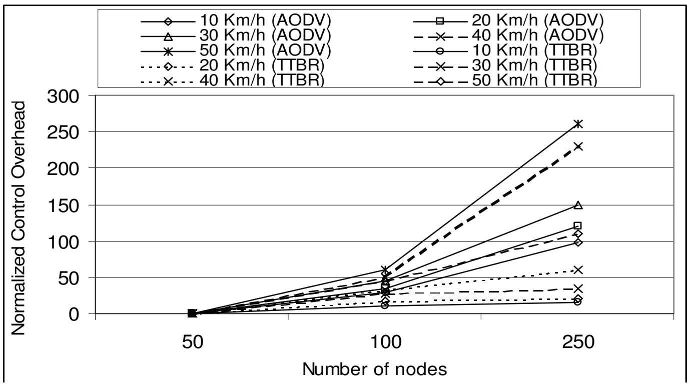


Fig.19: Normalized Control Overhead v Number of Nodes in Freeway scenario.

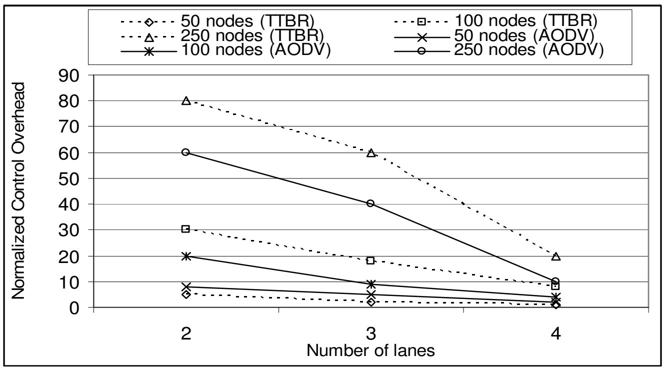


Fig.20: Normalized Control Overhead v Number of Nodes in Freeway scenario.

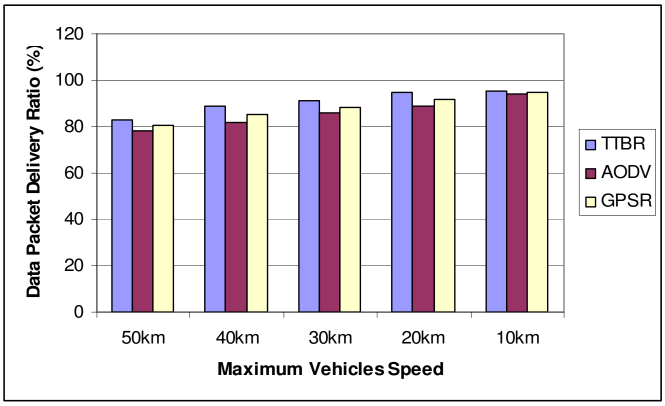


Fig.21: Data Packet Delivery Ratio v Number of Nodes in Freeway scenario.

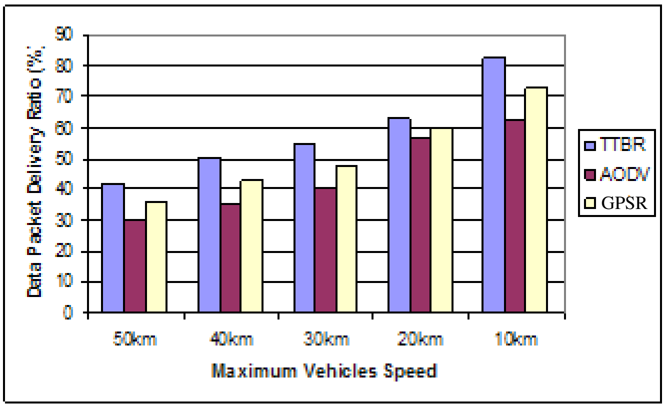


Fig.22: Data Packet Delivery Ratio v Number of Nodes in Freeway scenario.

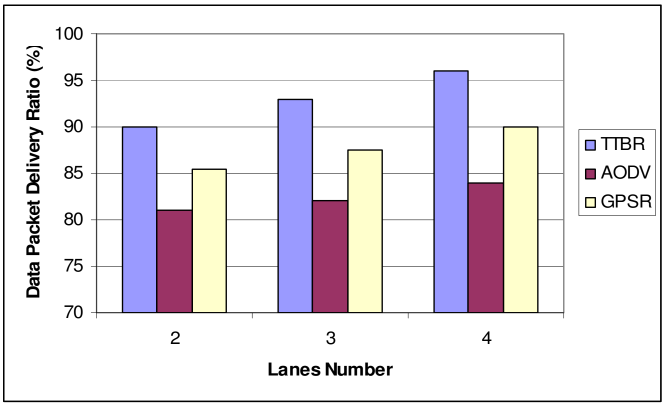


Fig.23: Data Packet Delivery Ratio v Number of Nodes in Freeway scenario.

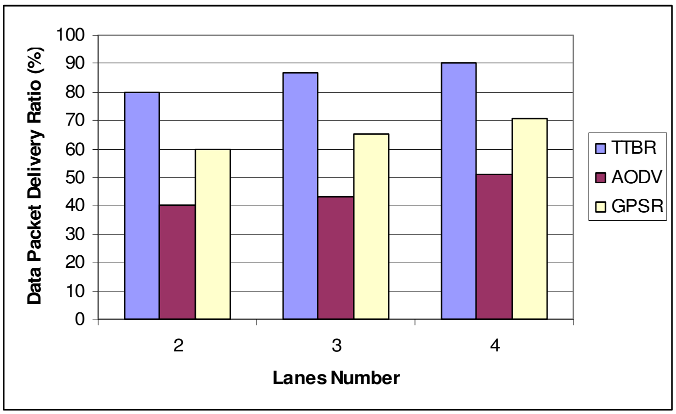


Fig.24: Data Packet Delivery Ratio v Number of Nodes in Freeway scenario.

XIII. CONCLUSIONS

A new routing protocol for vehicular ad hoc networks is proposed and developed. This protocol presents good scalability properties, which is a critical issue for VANET, where a high nodal density needs to be supported and high number of nodes can be involved (>100). This scalability target was obtained through the trajectory concept applied in the forwarding scheme. Specifically, a hierarchical trajectory-based routing scheme is proposed where a specific local and map-based trajectory is applied to forward the data packet in a local environment and global and coarse cell-based trajectories are determined to send the data packet to a larger distance. The local trajectory can take advantage of the specific map knowledge to find an optimized path on the street and corner. On the other hand, the cell-based trajectory permits the avoidance of specific info storage, such as anchor points, streets or nodes that increase the complexity of the trajectory-based forwarding. The cell-based trajectory needs just to know the cell-id and other coarse parameters such as data traffic or vehicular load and so on, that permits the best high-level choice to be made following some optimization criteria.

REFERENCES

[1] F.Borgonovo, A.Capone, M.Cesana, L.Fratta, "ADHOC MAC: new MAC architecture for ad hoc networks providing efficient and reliable point-to-point and broadcast services," *Wireless Networks Journal*, Vol.10, issue 4, pp.359-366, July 2004.

[2] J.LeBrun et al. "Knowledge-Based Opportunistic Forwarding in Vehicular Wireless Ad Hoc Networks," *in 62nd Vehicular Technology Conference*, Vol.4, 25-28 Sept., 2005, pp 2289–2293.

[3] W.Chen, S.Cai, "Ad Hoc Peer-to-Peer Network Architecture for Vehicle Safety Communications," *IEEE Comm. Magazine*, vol.43, Issue 4, Apr.2005 pp.100- 107.

[4] H.J.Reumerman, M.Roggero, M.Ruffini, "The Application-BasedClustering Concept and Requirement for Intervehicle Networks," *IEEE Comm. Magazine*, vol.43, Issue 4, Apr.2005 pp.108- 113.

[5] R.A.Santos, O.Alvarez, A.Edwards, "Performance Evaluation of two Location-Based Routing Protocols in Vehicular Ad-Hoc Networks," *in 62nd Vehicular Technology Conference (VTC-2005-Fall),* Vol.4, 25-28 Sept., 2005, pp 2287 – 2291.

[6] H.Wu et al. "MDDV: A Mobility-Centric Data Dissemination Algorithm for Vehicular Networks," *in Proc. of the 1st ACM Int. Workshop on Vehicular Ad Hoc Networks (VANET'04)*, pp. 47-56, 1 Oct., Philadelphia, PA, USA, 2004.

[7] C.Lochert et al. "A Routing Strategy for Vehicular Ad Hoc Networks in City Environments," in IEEE Proceedings of Intelligent Vehicles Symposium, 9-11 June 2003, pp.156–161.

[8] F. De Rango, M.Gerla, K.Biao Zhou, S.Marano, "GeO-LANMAR Routing Protocol: Asymptotic Analysis in Large and Dense Ad Hoc Networks," *2nd Int. Conf. On Broadband Networks (Broadnet 2005)*, 3-7 Oct., Boston, Massachusetts, USA, 2005.

[9] I. Stojmenovic, Position based routing in ad hoc networks, *IEEE Communications Magazine*, Vol. 40, No. 7, July 2002, 128-134.

[10] B.Karp, H.T.Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *in ACM/IEEE Proc. of of Int. Conf. on Mobile Computing and Networking (MobiCom'00)*, Boston, Massachusetts, United States, pp.243-254, 2000.

[11] Christian Lochert, Martin Mauve, Holger Füßler, Hannes Hartenstein, "Geographic Routing in City Scenarios", *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, pp. 69--72, January 2005.

[12] Christian Maihöfer, Reinhold Eberhardt, Elmar Schoch, "CGGC: Cached Greedy Geocast", *WWIC 2004*, pp. 13-25, 2004.

[13] D.Niculescu, B.Nath, "Trajectory Based Forwarding and Its Applications," *in Proc. of 9th Int. Conf. on Mobile Computing and Networking (Mobicom'03)*, San Diego, CA, USA.

[14] L.Blazevic, J.-Y.Le Boudec, S.Giordano, "A Location-Based Routing Method for Mobile ad Hoc Networks," *IEEE Transaction on Mobile Computing*, vol.4, n.2, Apr.2005, pp.97-110.

[15] J. Gong, Cheng-Zhong Xu and J. Holle, "Predictive Directional Greedy Routing in Vehicular Ad Hoc Networks", *27-th International Conference on Distrubeted Computing System Workshops, 2007*, 22-29 June 2007.

[16] M. Yuksel, R. Pradhan, S. Kalyanaraman, "An Implementation framework for trajectory-based routing in ad hoc networks", Ad Hoc Networks, Vol. 4, issue 1, Jan. 2006, pp. 125-137.

[17] Y. Ding, C. Wang, L. Xiao, "Static-node assisted adaptive routing protocol in vehicular networks", *4-th ACM International Workshop on Vehicular Ad Hoc Networks*, Sept. 10-20 2007, Montreal Canada.

[18] F.De Rango et al., "Hierarchical Trajectory-based Routing Protocol for Vehicular Ad Hoc Networks," *SPECTS 2008*, Edhinburg, Scotland, 2008.

[19] C. Santivanez et al., "Asymptotic Behaviour of Mobile Ad Hoc Routing Protocols with respect to Traffic, Mobility and Size", *TR-CDSP-00-52*, Department of Electrical & Computer Enginnering Northeastern University, Boston, MA, Oct.2000.

[20] W.H.Liao, Y.-C. Tseng, J.-P. Sheu, "GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks," *Telecommunication Systems 2001,volume 18, number 1-3, pages 37-60.*

[21] Y.Yu, G.-H. Lu, Z.-L.Zhang, "Enhancing Location Service Scalability with HIGH-GRADE," in IEEE Int. Conf. on Mobile Ad-hoc and Sensor Systems, 25-27 Oct., 2004 pp.164-173.

[22] S.J.Phikip, C.Qiao, "Hierarchical Grid Location Management for Large Wireless Ad Hoc Networks," *in 4th ACM Symp. On Mobile Ad Hoc Networking and Computing (MobiHoc'03)*, Annapolis, MD, USA,. pp.33-34, June 1-3, 2003.

[23] F.Bai, N.Sadagopan, A.Helmy, "The IMPORTANT framework for analysing the Impact of Mobility on Performance od Routing Protocols for Ad Hoc Networks," *Ad Hoc Networks Journal – Elsevier*, Vol.1, Issue 4, pp.383-403, Nov.2003.

[24] Network Simulator v.2 (ns-2); http://www.isi.edu/nsnam/ns/.

[25] http://ceng.usc.edu/~helmy.

[26] JengFarn Lee and MengChang Chen, "Sniffing Out Correct Error Frame Model of NS-2 Simulator", *11-th Communications and Networking Simulation Symposium (CNS 2008),* April 14-17 2008, Ottawa, Canada.

Floriano De Rango received the degree in computer science engineering in October 2000, and a Ph.D. in electronics and communications engineering in January 2005, both at the University of Calabria, Italy. From January 2000 to October 2000 he worked in the Telecom Research LAB C.S.E.L.T. in Turin as visiting scholar student. From March 2004 to November 2004 he was visiting researcher at the University of California at Los Angeles (UCLA). From November 2004 until September 2007 he has been a Research Fellow in the D.E.I.S. Department, University of Calabria where he is now Assistant Professor. He was recipient of Young Researcher Award in 2007. He served as reviewer and TPC member for many International Conferences such as IEEE VTC, ICC, WCNC, Globecom, Med Hoc Net, SPECTS, WirelessCOM, WinSys and reviewer for many journals such as IEEE Communication Letters, JSAC, IEEE Trans.on Vehicular Technology, Computer Communication, Eurasip JWCN, WINET etc. His interests include Satellite networks, IP QoS architectures, Adaptive Wireless Networks, Ad Hoc Networks and Pervasive Computing. He has co-authored more than 130 papers among International Journal and Conferences Proceedings.

Fiore Veltri graduated in computer engineering at the University of Calabria, Italy, in 2005. Since 2005 he has been with the telecommunications research group of D.E.I.S. in the University of Calabria. Since 2006 he has been a Ph.D student in Electronics and Communications Engineering at the University of Calabria. His research interests include: wireless network physical layer, UWB network, channel and mobility model, routing protocol in ad-hoc network.

Peppino Fazio received the degree in computer science engineering in May 2004. Since November 2004 he has been a Ph.D student in Electronics and Communications Engineering at the University of Calabria and he has got the Ph.D. in January 2008; at the moment he is a research fellow at DEIS Department of University of Calabria and he is collaborating with DISCA Department of “Universidad Politecnica de Valencia”. His research interests include mobile communication networks, QoS architectures and interworking wireless and wired networks, mobility modelling for WLAN environments and mobility analysis for prediction purposes.

Salvatore Marano, graduated in electronics engineering at the University of Rome in 1973. In 1974 he joined the Fondazione Ugo Bordoni. Between 1976 and 1977 he worked at the ITT Laboratory in Leeds, United Kingdom. Since 1979 he has been an Associate Professor at the University of Calabria, Italy. He was reviewer for many journals such as IEEE Communication Letters, JSAC, IEEE Trans.on Vehicular Technology, IEEE Transaction on Wireless Comm., European Transaction on Telecommunication Journal. His research interests include performance evaluation in mobile communication systems, satellite systems and 3g/4G networks. He published more than 130 papers among international conferences and journals.