



HAL
open science

Partial backwards routing protocol for VANETs

Tawfiq Nebbou, Mohamed Lehsaini, Hacene Fouchal

► **To cite this version:**

Tawfiq Nebbou, Mohamed Lehsaini, Hacene Fouchal. Partial backwards routing protocol for VANETs. Vehicular Communications, 2019, 18, pp.100162. 10.1016/j.vehcom.2019.100162 . hal-02969805

HAL Id: hal-02969805

<https://hal.science/hal-02969805>

Submitted on 25 Oct 2021

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.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Partial Backwards Routing Protocol for VANETs

Tawfiq Nebbou^a, Mohamed Lehsaini^a, Hacène Fouchal^{b,*}

^aSTIC Laboratory, Tlemcen University, Algeria

^bCRéSTIC, Université de Reims Champagne-Ardenne, France

Abstract

The design and implementation of efficient routing protocols for highly dynamic networks such as Vehicular Ad hoc NETWORKs (VANETs) is a challenging task, due to the specific features of these networks such as rapid movements of nodes, frequent link disconnections and large scale networks. Efficient routing mechanisms need to adopt to the VANETs characteristics to achieve high routing performances. One alternative is the use of the traffic information in the routing process. In this paper, we propose a novel routing protocol called Partial Backwards Routing Protocol (PBRP), which consists of three integrated strategies: dissemination of road traffic information, partial forwarding algorithm and backwards recovery strategy working together to provide permanent and advanced information about vehicular traffic which widely helps the routing algorithm to stand up to dynamic changes of VANETs topology, and to bypass the local vision which is the drawback of GyTAR/EGyTAR protocols. Our proposed protocol has been developed over OMNET++ simulator, evaluated and compared with some other protocols. The simulation results show that the packet delivery ratio of our proposal is upper about 54% compared to GyTAR/EGyTAR and the end-to-end delay of our protocol is reduced by 75% for most of scenarios.

Keywords: VANETs, Road Connectivity, Traffic information, Geographic Routing Protocols, Backward strategy.

1. Introduction

A Vehicular Ad hoc Network (VANET) is a highly mobile wireless ad hoc network designed to achieve many objectives related to driving safety, traffic management. VANET is a self-organized network composed of the interconnection of vehicles in which the vehicles can communicate with each other directly via Vehicle-to-Vehicle (V2V) communications, or with infrastructure namely the Road-Side-Unit (RSU) through Vehicle-to-Infrastructure (V2I) communication. A mixed communication V2V and V2I is also possible in such networks.

VANETs have their own features not common with the other classes of wireless ad hoc networks, even with its nearest class which is the Mobile Ad-hoc Networks (MANETs). The high mobility of vehicular nodes implies frequent topology changes causing difficulties to use topology-based routing protocols. Many researches have been proposed to address the routing problems in VANETs, such as geographic routing protocols.

In geographic routing protocols, when an intermediate vehicular node needs to forward data packets to a destination vehicle, the source vehicle transmits the data directly to destination in the case of destination within its communication range i.e. destination is a neighboring node. In the other cases, the source vehicle requests its position through location services such as GLS (Grid Location Service) [1, 2], HLS (Hierarchical Location Service) [3], RLS (Reactive Location Service) [4] and SFLS (Semi-Flooding Location Service) [5]. Once the position of the destination node is available, the source node inserts the location information about the destination in the header of packet and selects a neighboring vehicle as a next-hop by using its own forwarding strategy (e.g. nearest neighbor to the destination).

When the next-hop receives properly the data packets it becomes the new forwarding node and should repeat the same process until the destination is reached using a greedy approach, note that the forwarding node repeats the same process of source node except the destination position request process, because of the destination position is already in the packet header. The request process is not updated in every hop because when a vehicle sends position request to LIS, in this case, the vehicle

*Corresponding author

Email address: Hacene.Fouchal@univ-reims.fr (Hacène Fouchal)

cannot forward data until it receives destination position replay from its LIS. So, it is not appropriate to repeat this process because it takes significant time and generates more delay to deliver packets. We suggest that only the last vehicle can repeat the location request process. In the scenario where no suitable neighbour could be found (local optimum), the forwarding node can either switch to recovery mode, or drop data if there is no recovery mode strategy supported by the routing protocol. In the case of the well known recovery mode, the Store Carry and Forward (SCF) and the Perimeter Forwarding are proposed in GPSR protocol [6].

This study presents a novel routing protocol for VANETs called Partial Backwards Routing Protocol (PBRP), which consists of three integrated strategies: dissemination of road traffic information, partial forwarding algorithm and backwards recovery strategy working together to provide permanent and advanced information about vehicular traffic which widely helps the routing algorithm to stand up to dynamic changes of VANETs topology, and to bypass the local vision which is the drawback of GyTAR/EGyTAR protocols. We have proposed primary ideas of this work in [7]. This kind of protocol is very useful for some practical applications as emergency message exchange, entertainment stream exchange between vehicles.

The remainder of the paper is organized as follows. Section 2 is dedicated to related works. Section 3 details our solution. Section 4 details the results of our simulations. Section 5 concludes the study and gives some ideas on future extensions.

2. Related works

The well known greedy routing protocol for VANETs called Greedy Perimeter Stateless Routing (GPSR) is proposed in [6]. In GPSR protocol, the packet transmission is performed through two routing strategies: Greedy forwarding and Perimeter forwarding. Greedy forwarding strategy consists in selecting the closest node to the destination as the next-hop, in the case where the distance between the forwarding node and the destination is closer than the distance to the destination of its neighbors. In such cases, GPSR switches to the perimeter forwarding strategy which is based on the right-hand rule. GPSR is more efficient on highway environments than in city one due the presence of obstacles, which leads to increase link failures and causes more greedy forwarding fails.

In [8], the authors present a recent survey on geographic routing protocols. [9] is dedicated to a review on various routing protocols, it provides a very useful

study on routing over vehicular networks. A detailed distinction of various routing techniques is presented with discussion according to its advantages, and disadvantages, along with its constraints. [10] discusses a comprehensive survey of position-based routing protocols for FANETs (Flying Ad hoc Networks) with their various categories. FANETs are a special case of VANETs, they cover a 3-dimension space. Another intelligent routing protocol for vehicular ad hoc networks proposed in [11] able to interact with the environment to learn the transmission parameters by taking into account various metrics such as data-rate and route length. This protocol is based on two algorithms: Rate estimation algorithm and Route selection algorithm. The Rate estimation algorithm adopts Q-Learning algorithm to estimate the reception ratio by using Hello messages as an indicator of link quality. In addition, the Route selection algorithm uses the Q-Learning and fuzzy logic-based algorithms to find the best route in term of end-to-end delay and to evaluate the direct link respectively.

In [12], using the three proposed mechanisms, the authors propose an optimized geographic routing protocol called O-CLWPR based on CLWPR which uses information transport via Stigmergy, Social Behavior and Adaptive Caching Time of the Carry-and-Forward mechanism. The simulations have been done using NS-3 simulator and the obtained results show a real improvement of the end-to-end delay.

In [13], a new clustering-based reliable low-latency multipath routing (CRLLR) scheme is proposed by using Ant Colony Optimization (ACO) technique. The link reliability is used as criteria for Cluster Head (CH) selection i.e a vehicle will be selected as CH if it has maximum link reliability. The ACO technique is employed to compute the optimal routes among the communicating vehicles for VANETs in terms of reliability, end-to-end latency, throughput and energy consumption. Simulation results demonstrate that the proposed protocol outperforms other protocols in term of overall latency and reliability.

Another geographic routing protocol called Directional Greedy Routing (DGR) [14] solves the previous drawback of GPSR protocol. DGR uses the information about vehicle directions to improve the inter-vehicle communication, it combines two forwarding strategies: 1) Position first forwarding which consists to forward the packets to the closest node towards destination, 2) Direction first forwarding based on a simple geographic greedy forwarding in which it tries to find the closest neighbor node moving towards the destination as next-hop. Predictive Directional Greedy Routing (PDGR) [14] is an extended version of DGR, it uses two-hop

neighbor beacons to predict the future neighbors, reducing links failure and improve the next-hop selection mechanism.

Geographic Source Routing (GSR) [15] is an another alternative of geographic routing protocol that takes the advantage of city environment by using map street information in the routing process. In GSR, the geographic position of destination node is maintained via the reactive-based location service RLS [4]. In addition, GSR uses Dijkstra algorithm on a graph containing the geographic information and the urban topology to calculate the shortest path, this path is composed of a sequence of junctions from source to destination. Each packet should follow the sorted list of junctions within the path to reach the destination. However, the major weakness of GSR is that uses a static information to compute the shortest path, this mechanism is almost not suitable in such dynamic networks as VANETs which leads to decrease the performance of routing.

A-STAR (Anchor-based Street and Traffic Aware Routing) [16] uses traffic vehicular information in the routing process, it assigns a connectivity value to the roads depending of its capacity and number of bus line. The source vehicle computes the shortest path toward the destination by applying Dijkstra algorithm on a graph which is weighed by the road connectivity values.

VPGR (Vertex-Based Predictive Greedy Routing) [17] is a vertex-based routing protocol using the hybrid communication V2V and V2I. VPGR predicts a sequence of vertices from the source node to a fixed infrastructure within the location area of the final destination, then uses PDGR forwarding technique to forward packet between vertices. VPGR uses a simple metric RT (Remaining Time) to select the shortest path, this metric aims to select a vertex, which has a large remaining time of vehicles. This mechanism does not add new information for routing process because there are vehicles in the crossroads. Moreover, the shortest path is not optimal because it does not take in consideration the vehicular traffic information between vertices.

GyTAR (Improved Greedy Traffic Aware Routing) [18] is a geographical routing protocol conceived for VANETs in urban environment. The main idea of this protocol is the dynamic junction selection mechanism based on road density (number of vehicles on the road) and curvometric distance to the destination. GyTAR uses Cell Density Packet (CDP) to provide the information about candidate junctions in which the packets must pass, however, the initiation of CDPs is not regular with time because the initiation is based on which vehicle is about to leave its road. EGyTAR (Enhanced Greedy Traffic Aware Routing protocol) [19] is an extended ver-

sion of GyTAR. In EGyTAR the dynamic junction selection mechanism is based on the number of vehicles moving toward the candidate junctions (directional density).

AQRV (Adaptive Quality of service (QoS) based Routing for VANETs) [20] adopts ant colony algorithm to solve NP-complet problem which is constrained optimization problem of routing selection issues such as connectivity probability, packet delivery ratio and end-to-end delay. Moreover, in order to achieve best QoS the authors propose Ant Colony Optimization (ACO) to address this goal.

In B-MFR (Border node-based most forward within radius) [21] routing protocol, the authors define tree types of nodes based on the location of the neighboring nodes relative to the transmission range, in which the neighboring nodes within the transmission range are called "interior nodes", the nodes which are located on the border of transmission range are called "border nodes", and those outside are referred to as outer nodes. The main idea of this protocol is to forward the packet through the border nodes in order to minimize the number of hops between source and destination.

GPGR (Grid-based Predictive Geographic) routing protocol proposed in [22] combines between road topology information and vehicles mobility information in order to improve inter-vehicle communications and reduce local maximum forwarding problem. GPGR can avoid links failure by selecting a suitable next-hop based on the information of topology and vehicles mobility.

In LSGO (Link State Aware Geographic Opportunistic) routing protocol [23], the authors propose that each node calculates the link Expected Transmission Count (ETX) by the help of Periodic Hello packets. LSGO uses ETX value and destination distance as priority metrics. The higher priority node can forward the packet to its destination directly, in the contrary, the low priority node must wait a predefined timer, if the timer expires, and the high priority node does not transmit the packet, then, this low priority node can send the packet. For this reason LSGO uses multicast communications to perform the transmission, however, this mechanism increases the routing overhead and congests the network throughput.

LOR (Localized Opportunistic Routing) protocol [24] is an opportunistic routing protocol in which the topology are partitioned into nested Close-Node-Sets (CNSs) based on local information by using Distributed Minimum Transmission Selection algorithm (MTS-B). As result, the network are transformed from network of nodes to network of CNSs, in other words, each CNS acts as single node, then, for routing data. LOR proto-

col adopts AODV [25] to find CNSs path which contains CNS sequences.

HLAR (Hybrid Location-based Ad hoc Routing) protocol [26] combines between reactive routing protocols and location-based routing protocols. The main idea of HLAR is to use reactive routing protocol as an alternative for routing data when the location information degrades. HLAR combines greedy forwarding algorithm and AODV-ETX which is a modified version of AODV in which the Expected Transmission Count (ETX) [27] is used as metric instead of minimum hop-count.

3. Contribution

We aim to propose a novel routing protocol which is robust and efficient for *Non Delay Tolerant Networks* conceived for VANETs in order to improve the routing performance in city environment. The proposed protocol is composed of three essentials parts: 1) Dissemination of Road Traffic Information, 2) Partial Routing Algorithm, 3) Backwards Recovery Strategy.

Furthermore, in our contribution, we used Manhattan Grid mobility model which describes the vehicles movement in a grid area of 1500mx1500m composed of 16 straight roads in urban area. Moreover, the nature of the roads whether curved or straight does not have an impact on the results since the radio links propagate directly.

3.1. Dissemination of Road Traffic Information

The goal of Dissemination of Road Traffic Information (DRTI) is to deal with dynamic nature of VANETs. The main idea of DRTI is to maintain Road Traffic Information (RTI) of each road and share it with its successor roads as shown in Figure 1.

The closest vehicle to a junction (cross roads) is selected as RTIP Initiator Node(RIN) (yellow vehicle). The RINs are responsible to initiate periodically the Road Traffic Information Packet(RTIP), and forward it in the direction of all neighbouring junctions. The RIN node initiates periodically (at each 0.8 second) the RTIP packet in order to compute the connectivity, taking into consideration the vehicle speed in urban environment. The connectivity of a road does not have significant changes after 0.8 seconds. In other hand, the RTIP lifetime equals to 0.8 seconds which ensures the validity of the connectivity values. The first interesting property of our dissemination mechanism is that the RTIP packet can hold a number of roads information depending on the RTIP lifetime, this is mainly proposed to bypass the local vision which is the case of many routing protocols

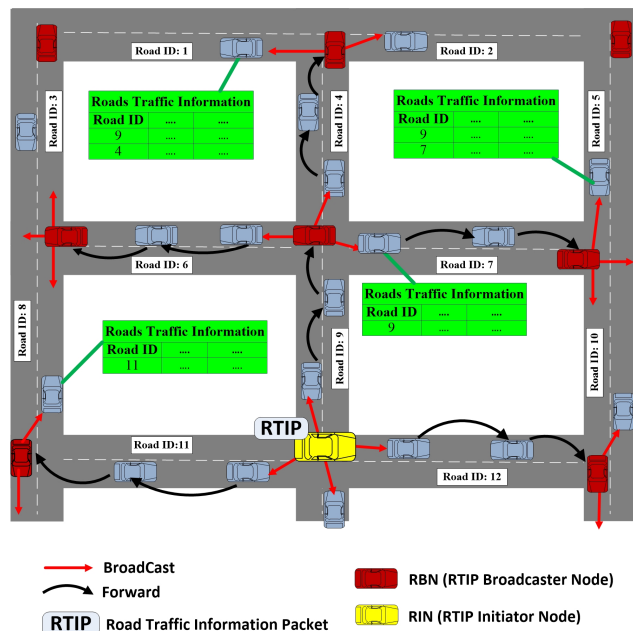


Figure 1: Dissemination of Road Traffic Information

such as GyTAR [28] and EGYTAR [19]. Figure 2 shown an example for local vision and advanced vision.

Additionally, DRTI provides a fresh information about roads such as packet crossing delay, the density of roads (number of vehicles) and road connectivity. For achieving this objective, in the following sub-sections we describe Road Dividing scheme, Road Traffic Information Packet (RTIP), the calculation of road connectivity and finally the dissemination of RTIP algorithm.

3.1.1. Road Dividing scheme

In our proposed routing protocol, each road is divided into a fixed number of physical cells in such manner that the area of a given road is fully covered by a certain number of physical cells with the same size depending on the transmission range (R_{tr}), the road width and the road length. In order to ensure these properties, let us start with an example of two consecutive cells like it shown in Figure 3:

The blue circles represent cells that come into contact in point 'c', the points 'a' and 'e' indicate centres of cells, 'f' and 'g' designate the intersection of cells with the road side as described in the figure, 'b' and 'd' are the projection of the points 'f' and 'g' respectively on segment [ae]. The yellow area represents the remaining area of the confluence of cells with the road, $[ae] = 2R_{tr}$ and $[ce] = [eg] = R_{tr}$.

It should be noted that the physical cells are used to defining a group of vehicles. Moreover, each group have

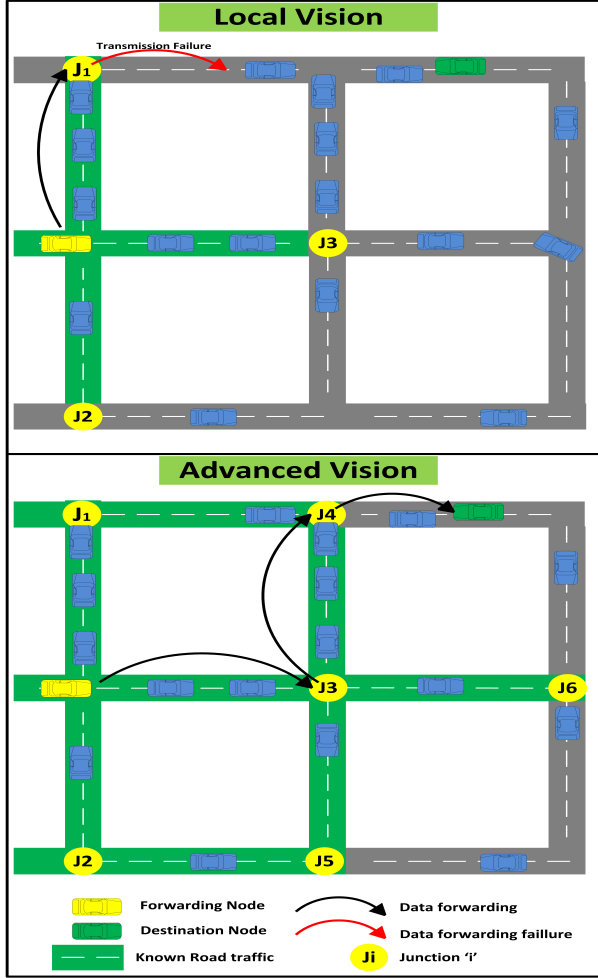


Figure 2: Local Vision vs. Advanced Vision

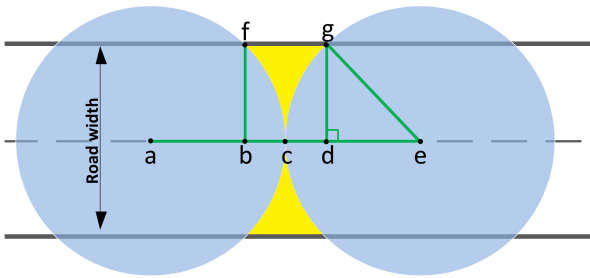


Figure 3: Cell Size: case1

to handle the vehicles information and must be involved in the dissemination of road information process.

The problem posed here is when the vehicles located in the yellow area can not be involved in the dissemination of road information process, this is because they are not belong to any group. We therefore propose to

determine the distance between cells in order to avoid the blind area (yellow area) as shown in Figure 4. This new cell distance is computed by pulling the right cell towards the left cell by the length of the segment $[bd]$ as follow :

$$new_cell_distance = [ae] - [bd] \quad (1)$$

Based on the figure 3, we have:

$$[bd] = 2[cd] = 2([ce] - [de]) \quad (2)$$

$$[de] = \sqrt{[ge]^2 - [gd]^2} = \sqrt{R_{tr}^2 - (Rw/2)^2} \quad (3)$$

Where Rw represents the road width.

According to (3):

$$(2) \Rightarrow [bd] = 2R_{tr} - 2\sqrt{R_{tr}^2 - (Rw/2)^2} \quad (4)$$

According to (4):

$$(1) \Rightarrow new_cell_distance = 2\sqrt{R_{tr}^2 - (Rw/2)^2} \quad (5)$$

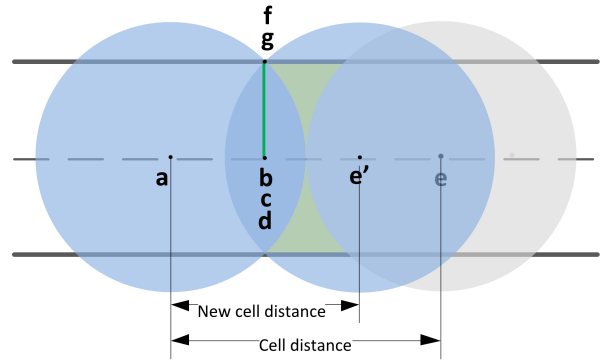


Figure 4: Cell Size: case2

Thus, according to (5), the maximum spacing cell in order to avoid blind area is defined as:

$$Max_cell_spacing = 2\sqrt{R_{tr}^2 - \frac{Rw^2}{4}} \quad (6)$$

Then, the number of cells for a given road is computed as follows:

$$Number_of_cell = ceiling\left(\frac{road_length}{Max_cell_spacing}\right) \quad (7)$$

Where, $road_{length}$ represents the road length, the function $ceiling(x)$ give an integer greater than or equal to x as expressed by equation 8.

$$ceiling(x) = \begin{cases} x & \text{if } x \in \mathbf{N} \\ INT(x) + 1 & \text{if } x \in \mathbf{R} \end{cases} \quad (8)$$

Finally, the cell size is given by:

$$Cell_{size} = \frac{road_{length}}{Number_of_cell} \quad (9)$$

If we do not assume that the roads are straight, the connectivity will be better and we will have more cells (see Figure 5). In addition, since the radio links are straight, the nature of the road does not have an impact on the connection between vehicles. Except that the modelling of the obstacles cannot be generalised according to an analytical model, however in the part of simulations and following the model of mobility Manhattan Grid, the obstacles are taken into account, this can be reflected on the reality.

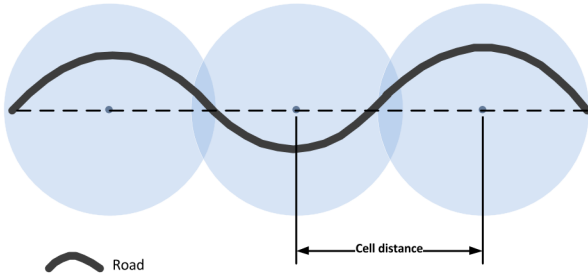


Figure 5: example of cell distance of curve road

3.1.2. Road Traffic Information Packet

The RTIP is designed to store and transfer the roads traffic information. The format of RTIP is pictured in Figure 6, it consists of three parts:

1. **Static Header:** Initialized just once by the RTIP Initiator Node (RIN), it consists of two fields:
 - (a) *Time-stamp*: represents the initiation time of RTIP,
 - (b) *Source Junction ID*: comprises the current junction ID of the RIN.

In addition, these previous fields serve as an identifier of RTIP.

2. **Dynamic Header:** used by RIN and RTIP Broadcaster Node (RBN), unlike of static header, the dynamic header can be updated at each broadcasting process (described in Algorithm). The latter

contains the information relevant to the next road which will be processed. It is composed of:

- (a) *From Junction ID*: represents the current junction ID of the forwarding node (RIN or RBN),
- (b) *To Junction ID*: denotes the destination junction ID,
- (c) *Time-stamp*: designates the start time of the broadcasting process.

3. **Roads Information:** The most important part, it comprises the roads information entry, where each entry is created by a RBN (or by RIN for the initial RTIP). In addition, all entries contain the following fields:

- (a) *Road ID*: the current road identifier,
- (b) *Road Connectivity*: indicates the value of the road connectivity, it must be updated (following the Algorithm) at all Group Leaders (GLs) located in the current road,
- (c) *Road Density*: denotes the road density (the number of vehicles),
- (d) *Delay*: designates the time taken for a to through the current roads,
- (e) *Time-stamp*: represents the creation time of the road information table entry.

| Road Traffic Information Packet | | | | |
|---------------------------------|------------------|-------------------|--------------------|------------|
| Static Header | Time-stamp | | Source Junction ID | |
| | From Junction ID | | To Junction ID | Time-stamp |
| Dynamic Header | Road ID | Road Connectivity | Road Density | Delay |
| | ... | ... | ... | ... |
| Road Information Table | ... | ... | ... | ... |
| | ... | ... | ... | ... |

Figure 6: Road Traffic Information Packet (RTIP)

3.1.3. The Connectivity: two-vehicles, Cell and Road

In this subsection, we aim to define the Link Connectivity (LC) of: two-consecutive vehicles, cell and road. Note that the lifetime of all these information is equals to 0.8 seconds and it is enough to ensure the validity of these information.

Firstly, the LC between two consecutive vehicles measures the remaining communication distance between these vehicles. Consequently, two vehicles are connected as long as the remaining communication distance value is greater than zero, otherwise; the vehicles are

not connected ($LC = 0$). The LC of two-vehicles can be expressed as:

$$LC_{nodes}(v_i, v_j) = \begin{cases} R_{tr} - dist(v_i, v_j), & \text{if } dist(v_i, v_j) \leq R_{tr} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Where v_i and v_j depict the vehicle id i and j respectively, $dist$ stands for *distance* and R_{tr} is an abbreviation for range of transmission.

Indeed, the physical cell defines a group of vehicles. In this context, the LC of cell is related to the LC of its group members (vehicles), we therefore propose that the LC of a cell is the mean LC_{nodes} of all consecutive vehicles within the cell. LC_{cell} is defined as:

$$LC_{cell} = \frac{\sum_{i=1}^{N-1} LC_{nodes}(v_i, v_{i+1})}{N-1} \quad (11)$$

Where N represents the number of vehicles within in the cell (cell density).

In the similar manner of Link Connectivity of cells, the road is composed of a set of consecutive cells, and by taking into consideration the connectivity inter-cells and the cell density which differs one from another. The completed road connectivity can be expressed as follow:

$$LC_{road} = \frac{\sum_{i=1}^{N_{cell}} (N_i * LC_{cell_i}) + \sum_{i=1}^{N_{cell}-1} LC_{inter}(cell_i, cell_{i+1})}{\sum_{i=1}^{N_{cell}} N_i} \quad (12)$$

$$LC_{inter}(cell_i, cell_{i+1}) = LC_{nodes}(last_v(cell_i), first_v(cell_{i+1})) \quad (13)$$

Where $LC_{inter}(cell_i, cell_{i+1})$ is the link connectivity between the last vehicle of $cell_i$ and the first vehicle of $cell_{i+1}$ (the next cell).

Moreover, in order to compute the connectivity of a road with formula 12, we must have the connectivity and density values of all cells located in the road. For the reason of the good exploitation of spaces (e.g. packet size), we aim to compute the road connectivity in a progressive manner by using only two fields: previous cell connectivity and previous cell density, with this context, the road connectivity will be computed from cell to cell progressively as follow:

$$PLC_{i(>1)} = \frac{D_{i-1}PLC_{(i-1)} + N_iLC_{cell_i} + LC_{inter}(cell_{i-1}, cell_i)}{D_i} \quad (14)$$

Where:

$$\begin{cases} PLC_1 = LC_{cell_1} \\ D_{i(>1)} = D_{i-1} + N_i = \sum_{j=1}^i N_j \\ D_1 = N_1 \end{cases}$$

PLC_i indicates Progressive Link Connectivity at cell number i , (PLC_{i-1} for previous cells connectivity), D_i indicates the previous cells density (the sum of density of all previous cells of the cell number i (the current cell density is included)).

Finally, the road connectivity can be also expressed as:

$$LC_{road} = PLC_{N_{cell}} \quad (15)$$

We can demonstrate formula 15 by mathematical induction as follow:

- **Base case:** Show that the formula.15 true for $N_{cell} = 1$.

$$\begin{aligned} -PLC_1 &= LC_{cell_1} \quad \text{by definition (formula 14);} \\ -LC_{road} &= \frac{\sum_{i=1}^1 (N_i * LC_{cell_i}) + \sum_{i=1}^0 LC_{inter}(cell_i, cell_{i+1})}{\sum_{i=1}^1 N_i} \\ &= \frac{N_1 * LC_{cell_1}}{N_1} = LC_{cell_1} \end{aligned}$$

We see that base case is true for $N_{cell} = 1$.

- **Inductive step:** Show that if formula.15 is true for $N_{cell} = n(n > 1)$, then also true for $N_{cell} = n + 1$.

Using formula 14 for $N_{cell} = n + 1$:

$$PLC_{n+1} = \frac{D_n PLC_n + N_n LC_{cell_n}}{D_{n+1}} + \frac{LC_{inter}(cell_n, cell_{n+1})}{D_{n+1}} \quad (a)$$

by assumption, $LC_{road} = PLC_{N_{cell}}$ is true for $N_{cell} = n$, we have:

$$\begin{aligned} PLC_n &= LC_{road} \\ &= \frac{\sum_{i=1}^n (N_i * LC_{cell_i}) + \sum_{i=1}^{n-1} LC_{inter}(cell_i, cell_{i+1})}{\sum_{i=1}^n N_i} \end{aligned}$$

by multiplying by D_n , we have:

$$D_n PLC_n = D_n \frac{\sum_{i=1}^n (N_i * LC_{cell_i})}{\sum_{i=1}^n N_i} + D_n \frac{\sum_{i=1}^{n-1} LC_{inter}(cell_i, cell_{i+1})}{\sum_{i=1}^n N_i} \quad (b)$$

We have $D_n = \sum_{i=1}^n N_i$ by using definition of formula.14, by substitution D_n in (b) we have:

$$D_n PLC_n = \sum_{i=1}^n (N_i LC_{cell_i}) + \sum_{i=1}^{n-1} LC_{inter}(cell_i, cell_{i+1}) \quad (c)$$

by substitution $D_n PLC_n$ in the right-hand side of (a) we have:

$$PLC_{n+1} = \frac{\sum_{i=1}^n (N_i LC_{cell_i}) + \sum_{i=1}^{n-1} LC_{inter}(cell_i, cell_{i+1})}{D_{n+1}} + \frac{N_n LC_{cell_n} + LC_{inter}(cell_n, cell_{n+1})}{D_{n+1}} \quad (d)$$

As a consequence:

$$\sum_{i=1}^n N_i LC_{cell_i} + N_n LC_{cell_n} = \sum_{i=1}^{n+1} N_i LC_{cell_i} \quad (e)$$

$$\sum_{i=1}^n LC_{inter}(cell_i, cell_{i+1}) = \sum_{i=1}^{n-1} LC_{inter}(i, i+1) + LC_{inter}(cell_n, cell_{n+1}) \quad (f)$$

by substitution (e) and (f) in the right-hand side of (d) we have:

$$PLC_{n+1} = \frac{\sum_{i=1}^{n+1} N_i LC_{cell_i} + \sum_{i=1}^n LCI_{(i, i+1)}}{D_{n+1}} \quad (g)$$

We have $D_{n+1} = \sum_{i=1}^{n+1} N_i$,

by substitution D_{n+1} , (g) become:

$$PLC_{n+1} = \frac{\sum_{i=1}^{n+1} N_i LC_{cell_i} + \sum_{i=1}^n LC_{inter}(cell_i, cell_{i+1})}{\sum_{i=1}^{n+1} N_i} = LC_{road_{(n+1)}}$$

The above calculation proves that formula.15 is true for $N_{cell} = n + 1$,

Finally, the base case and the inductive step have been performed by induction proofs, this allows formula.15 to be true.

3.1.4. Maintain and Share Road Traffic Information

Due the high dynamic nature of VANET networks, the topology changes frequently. Therefore, in order to enable the routing decision robust in such dynamic networks, it is very important to maintain a fresh and permanent road traffic information. For this reason we propose a distributed group information able to provide permanent road traffic information included within the RTIP packets. To ensure this property, each road is divided into a number of physical cells defining a group of vehicles (see Section 3.1.1). Each group elects a Group Leader (GL) (i.e. the closest vehicle to the cell center), we also propose that each RTIP packet must pass through all GLs belonging on the road as shown in Figure 7. Moreover, the GLs are mandated to update the RTIPs by including the information of its group (density, connectivity,...) at a fixed frequencies. In addition, since the vehicles are moving in both directions, transmission of route traffic information forwarding mechanism involves vehicles in both directions.

3.2. Partial Routing Algorithm

The proposed Partial routing protocol adopts anchor-based routing approach by using road traffic information, thanks to DRTI. Given that the RTIP packet can hold a number of roads information entry depending on the RTIP lifetime. Therefore, the forwarding node will have traffic information about a limited area of roads, in other words, the forwarding node has traffic information such as connectivity, density and road crossing delay about a set of roads which is stored in its Road Traffic Information Table (RTIT).

The proposed protocol is composed of two modules: Partial Path and Partial Path Forwarding Algorithm.

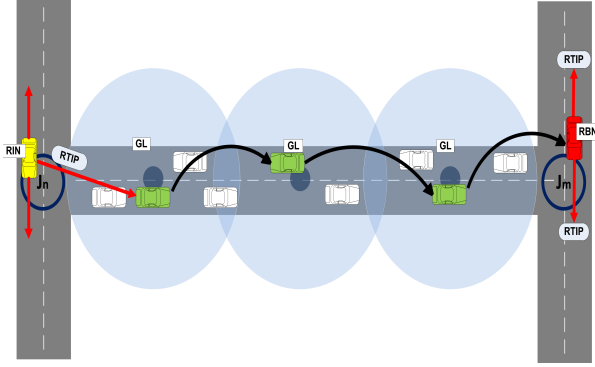


Figure 7: Road Traffic Information Forwarding Mechanism

3.2.1. Partial Path

Based on the given traffic information, the forwarding node computes a partial path towards the destination according to algorithm 2.

Firstly, the forwarding node generate a graph $G = (V, E)$ to represent the network, in which V and E represents a set of vertices (network junctions) and a set of edges (network roads) respectively. Moreover, each edge is weighed by the score function described as follows:

$$Score(road_i) = \begin{cases} \alpha(1 - \frac{LC_{road_i}}{R_{ir}}) + \beta(\frac{delay(road_i)}{RTIP_{lifetime}}), & if road_i \in RTI_{table} \\ 1 & otherwise \end{cases} \quad (16)$$

Where α and β denote the weight score ($\alpha + \beta = 1$).

In equation 16, we tends to compute a score of a given edge in such way that the road score is lower as long as it has higher connectivity and lower delay. In the scenario where the given road does not belong to RTIT, the score is 1 (unknown road traffic information).

Secondly, we compute the shortest path from source position to destination position by applying shortest path algorithm (such as Dijkstra algorithm) on a subgraph of the network where we consider only the non visited edges (we remove all visited edges by this current data packet), this is mainly to avoiding the routing loops.

Finally, the partial path is a subpath of the computed shortest path, whereby we keep only edges with known

traffic information, in other words, we remove from the shortest path all edges with unknown traffic information (edges with score=1). Figure 8 shows an example of partial path.

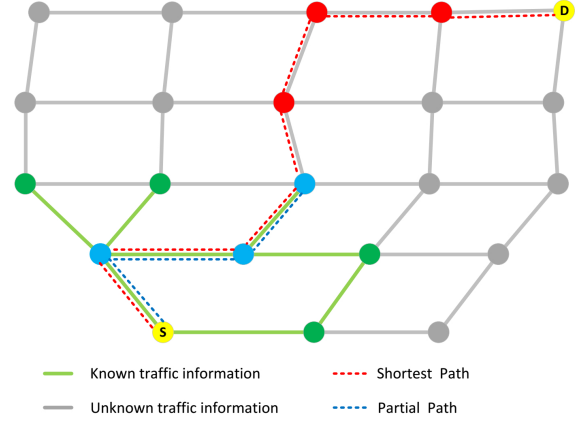


Figure 8: Partial path vs Shortest path

3.2.2. Partial Path Forwarding Algorithm

In our proposed routing protocol, the packet-header composed of three essential fields: partial path stack, recovery path stack and visited roads stack. The routing with partial path is very simple as described in algorithm 1, following the scenario depicted in Figures 9 and 10, where the source node S want to send data to the destination node D. Firstly the source node computes partial path and inserts it in the partial path stack. Then, by applying greedy forwarding approach the packets are forwarded by following all the junctions within the partial path stack in a sorted way, and when the data packet reaches an intermediate junction, we remove it from partial path stack and add it to recovery path stack. Moreover, we update the visited road stack for each crossed. When the data packet reaches the last junction of partial path (partial path stack becomes empty) (see node 4 in Figure 9), the current forwarding node computes a new partial path, and repeats the same process until the final destination is reached. In the other cases when we occur maximum local problem (the partial path forwarding fails) (see node 5 in Figure 9), we switch to Backward Forwarding Algorithm (described in Section 3.2.3). Figure 10 shows an example of backward recovery mode (see node 5 in Figure 10), when the packet reach maximum local problem, the current forwarding node uses recovery stack to go back to the previous junction, when the packet reaches a junction in which we can compute a new path from this junction to the destination, more then, this path should not contain

any visited road. If that is the case, the routing switches to partial routing mode to forward the packet to the final destination. Figures 12a, 12b and 12c show an example of forwarding data by using partial path forwarding algorithm, and we summarize the fundamental process of the proposed algorithm in the flowchart depicted in Figure 11.

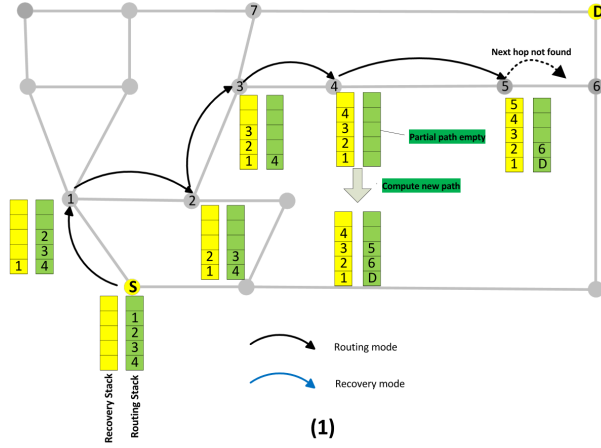


Figure 9: Partial path routing (1)

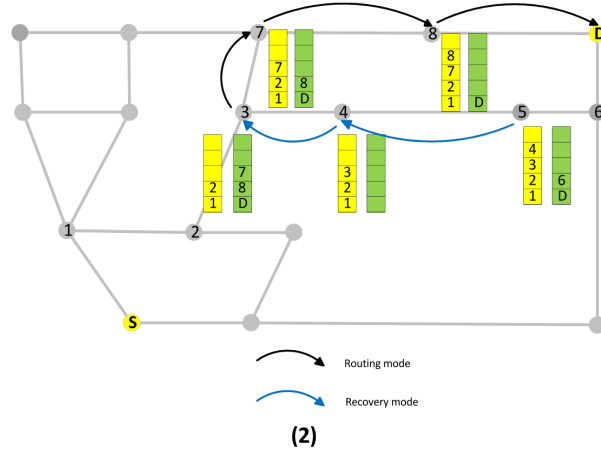


Figure 10: Partial path routing (2)

3.2.3. Backwards Forwarding Algorithm

The Backwards Forwarding Algorithm serves as a recovery mode when the partial path forwarding algorithm occurs maximum local problem. Furthermore, for making our proposed routing protocol more suitable for Non-DTN, in our proposed recovery mode, we don't use any delayed process like Carry and Forward approaches. So, when we occur local maximum problem,

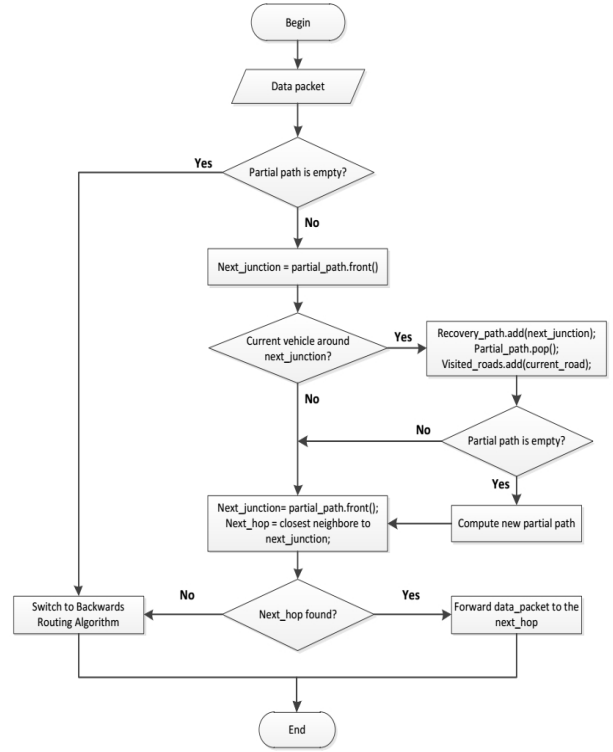


Figure 11: Flowchart of Partial Path Routing Algorithm

the forwarding node adds the current road into visited roads stack and returns the data packet back to the previous junction by using recovery path stack and when the data packet has reached the previous junction, the current node computes new partial path. Partial path algorithm does not consider the visited roads in the path calculation, the previous road (where we occur local maximum problem) will not belong to the new partial path. In the case of the new partial path is empty, the forwarding node removes the current junction from recovery path stack and forwards the packet to the previous junction (front of recovery path stack). This process is repeated until we find non empty partial path, or in the other cases, if the previous junction not found (recovery path stack is empty) we will drop the data packet instead of carry forward in order to keep our protocol suitable for Non-DTN. We summarize the fundamental process of the Backwards algorithm in the flowchart depicted in Figure 13 and we give more details in algorithm 6.

4. Evaluation

In this section, we evaluate the routing performance of our proposed routing protocol PBRP compared to

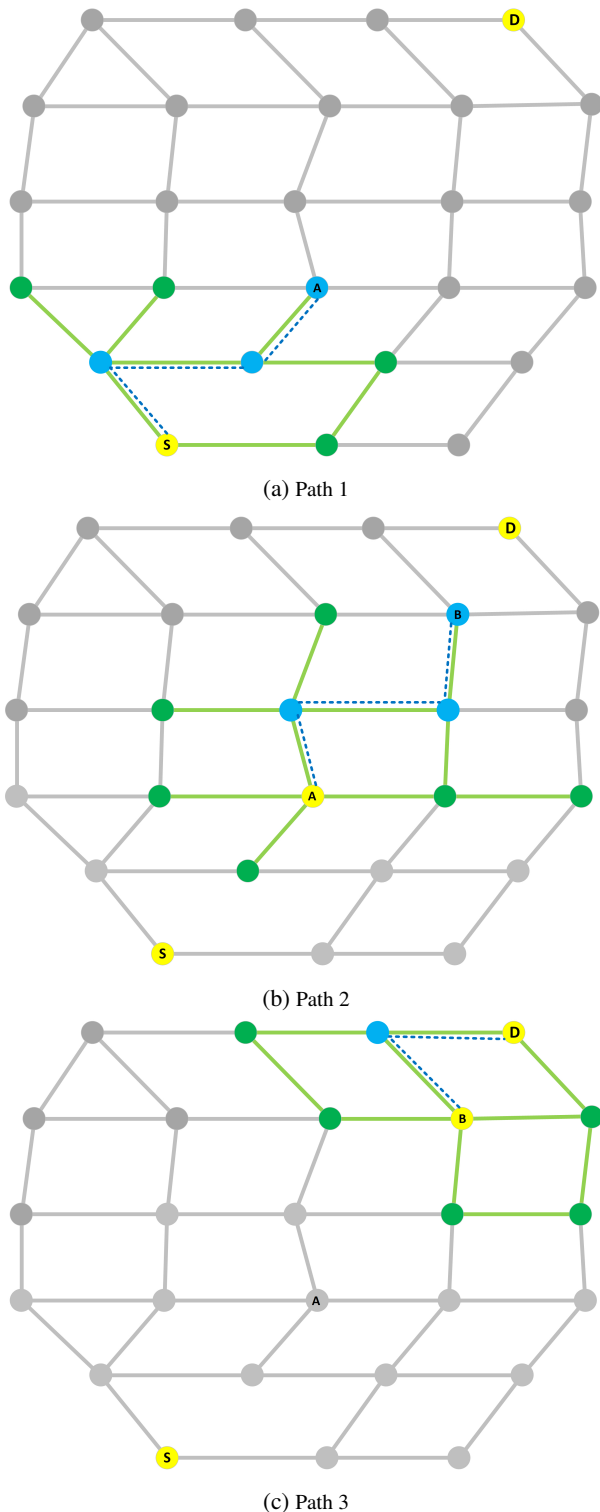


Figure 12: Example of partial path forwarding

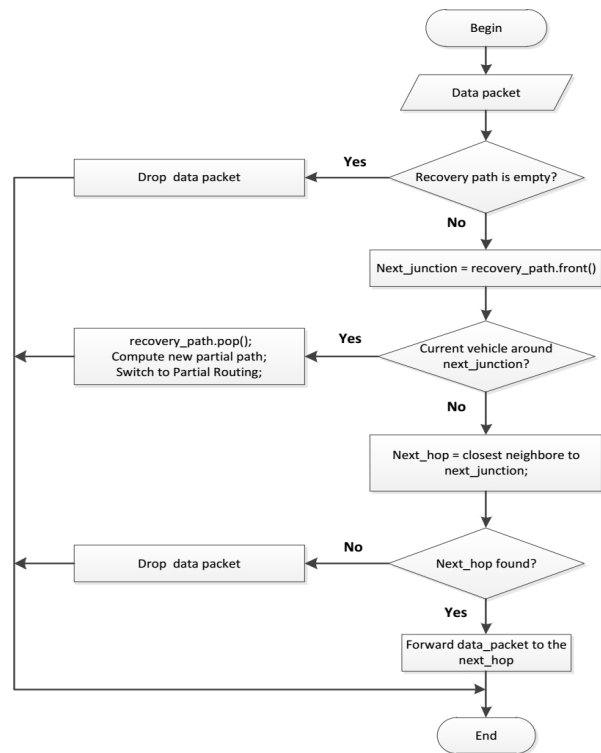


Figure 13: Flowchart of Backwards Forwarding Algorithm

GyTAR and EGyTAR in terms of packet delivery ratio, routing overhead and end-to-end delay. The Packet Delivery Ratio (PDR) metric measures the ratio of the total received packets over the total packets generated from the source, End-to-end delay represents traveling duration for a packet from the source to destination, and the Routing Overhead is measured in terms of control packets size. We consider the size of CDP packets for both protocols GyTAR and EGyTAR and for our proposal protocol, the routing overhead takes into account the RTIP packet size.

In our simulation we use Manhathan Grid mobility model which describes the vehicles movement in a grid area of 1500x1500m composed of 16 straight roads and all these protocols are implemented on OMNET++ simulator.

4.1. Working environment

The routing performances of GyTAR, EGyTAR and our proposed protocol PBRP are evaluated using OMNET-5.0 simulator and INET-3.4.0 internet stack framework. We carry out the simulation results depending on the simulation parameters which are summarized in table 1, by running each scenario 10 times, so that, the results are averaged. We includes some obstacles

Algorithm 1 : Partial Path Algorithm

```
1: Begin
   - path : The partial path (list of junctions)
   - data_packet : The data packet
   - G : Network graph
   -  $N_i$  : Node i
   - dst_p : The Destination position
   - self_p : The self position
   - neighbort : The neighboring table
   - next_hop : The next hop address
   - RTIt : Road Traffic Information Table
   - closest_j : The closest junction to  $N_i$ 
   - next_j : The next junction
   -  $R_{tr}$  : The range transmission
2:  $N_i$  receives data_packet;
3: if (data_packet.destination  $\in$  neighbort) then
4:   next_hop = data_packet.getDestinationAddress();
5: else
6:   dst_p = data_packet.getDestinationPosition();
7:   path = data_packet.getPath();
8:   if (path = null) then
9:     path = PartialPath(self_p, dst_p, G, RTIt);
10:    data_packet.setPath(path);
11:    if (path is empty) then
12:      next_hop = getClosestNeighboreTo(closest_j);
13:    end if
14:  else
15:    next_j = path.front();
16:    if (distance(self_p, next_j) <  $R_{tr}$ ) then
17:      path.pop(); /* delete front junction */
18:      if (path not empty) then
19:        next_j = path.front();
20:      else
21:        path = PartialPath(next_j, dst_p, G, RTIt);
22:        next_j = path.front();
23:      end if
24:      data_packet.setPath(path);
25:    end if
26:    next_hop = getClosestNeighboreTo(next_j);
27:  end if
28: end if
29: if (next_hop  $\neq$  null) then
30:   Forward data_packet to the next_hop;
31: else
32:   Store data_packet and carry it;
33: end if
34: End
```

defined by its shape, location and material in a grid area

Algorithm 2 : Partial Path Algorithm

```
1: Begin
   - pathpartial : The partial path
   - pathfull : The full path
   - G(V, E) : Network graph
   -  $G_w(V, E_w)$  : Weighted network graph
   -  $G_{Sub}(V_{Sub}, E_{Sub})$  : Subgraph of  $G_w$ 
   - RTIt : Road Traffic Information Table
   - visited_roads : Array of visited roads
   - dst_p : The Destination position
   - from_p : from position
2: for (all edges "e"  $\in$  E) do
3:   if (e  $\in$  RTIt) then
4:     e.weight = Score(e, RTIt);
5:   else
6:     e.weight = 1;
7:   end if
8:    $E_w = E_w \cup \{e\}$ ;
9: end for
10:  $E_{Sub} = \{e \in E_w \wedge e \notin \text{visited\_roads}\}$ 
11: pathfull = getShortestPath(from_p, dst_p, Gw);
12: for (all edges "e"  $\in$  pathfull) do
13:   if (e.weight  $\neq$  1) then
14:     pathpartial = pathpartial  $\cup$  {e};
15:   end if
16: end for
17: Return pathpartial;
18: End
```

of 1500x1500m which are described in an XML file. In addition, each vehicle has a digital map file to be aware about simulation environment. We also use Manhattan Grid mobility model to describe the movement of vehicular nodes in the simulation environment, Delictric Obstacle Loss Model for radio obstacle and 802.11p [29] protocol for MAC layer. The remaining simulation parameters are summarized in the Table 1.

4.2. Simulation Results

In this subsection, we present simulation results obtained for each protocol and compare them.

4.2.1. Packet Delivery Ratio

Figure 14 shows the packet delivery ratio depending on the number of vehicular nodes. As shown in this figure, the packet delivery ratio of all protocols increases as the network density increases, it is due to when the number of nodes increases, the forwarding node will have more chance to find the next hop. Furthermore, we clearly observe that our proposed protocol achieves

Algorithm 3 : Periodic Road Traffic Information Initiation

```
1: Begin
   -  $V$  : Vehicle
   -  $rti_p$  : Road traffic information packet
   -  $cell_n$  : Neighbor cell
   -  $current_j$  : The current junction of  $V$ 
   -  $first_j$  : The first junction of  $cell_n$ 
   -  $last_j$  : The last junction of  $cell_n$ 
   -  $SH$  : The static header of  $rti_p$ 
   -  $DH$  : The dynamic header of  $rti_p$ 
   -  $RIN$  : Road Traffic Information Initiator Node
2: if ( $V$  around cross roads) then
3:   if ( $V$  receive  $RIN\_initiation\_msg$ ) then
4:      $timer = 0$ ;
5:      $Start_{timer}(V)$ ;
6:   end if
7:   if ( $V$  is  $RIN \wedge timer > 1sec$ ) then
8:      $rti_p = new RTIPacket()$ ;
9:      $rti_p.SH.time\_stamp = currentTime()$ ;
10:     $rti_p.DH.time\_stamp = currentTime()$ ;
11:     $rti_p.SH.source\_junction = current_j$ ;
12:    for (all neighboring cells " $cell_n$ ") do
13:       $rti_p.DH.from\_junction = cell_n.first_j$ ;
14:       $rti_p.DH.to\_junction = cell_n.last_j$ ;
15:      Forward  $rti_p$  to  $GL$  of  $cell_n$ ;
16:    end for
17:    Broadcast  $RIN\_initiation\_msg$  around  $current_j$ ;
18:  end if
19: end if
20: End
```

higher packet delivery ratio continuously compared to that of GyTAR/EGyTAR. This is mainly because our proposed protocol uses an advanced vision and a permanent road traffic information thanks to DRTI, which makes our protocol able to avoid empty area in which the network is disconnected. Moreover, unlike the full path, the partial path has positive effects for data routing, it is also dynamic, flexible and practical for the dynamic networks such as VANETs, since it can be changed during the forwarding process as the network change by exploiting the variation of road traffic information. In other hand, our Backward recovery mode increases also the packet reception ratio, it serves to give another chance to forward data packet when the forwarding process occurs maximum local problem by forwarding-back the data packet and re-forwarding it through another path. However, GyTAR and EGYTAR show lower packet delivery ratio for the simple reason

Algorithm 4 : Road Traffic Information Forwarding Algorithm

```
1: Begin
   -  $V$  : Vehicle
   -  $RTI_{lf}$  : Road Traffic Information life time
   -  $GL$  : Group Leader
2:  $V$  receive  $rti_p$ ;
3: if ( $(currentTime - rti_p.SH.time\_stamp) > RTI_{lf}$ ) then
4:   delete  $rti_p$ ;
5:   Return;
6: end if
7: if ( $rti_p$  is broadcast) then
8:   Update RTI table following algorithm.5;
9:   Return;
10: end if
11:  $to_j = rti_p.DH.to\_junction$ ;
12:  $from_j = rti_p.DH.to\_junction$ ;
13: if ( $V$  is  $GL$  of  $current\_cell$ ) then
14:   Update  $rti_p$  following algorithm.6;
15:   if ( $current\_cell$  is last cell) then
16:     if ( $V$  is around  $to_j$ ) then
17:       Forward  $rti_p$  to  $RBN$ ;
18:     else
19:       Forward  $rti_p$  to closest neighbor to  $RBN$ ;
20:     end if
21:   else
22:     Forward  $rti_p$  to  $GL$  of next cell;
23:   end if
24: else
25:   Forward  $rti_p$  to  $GL$  of  $current\_cell$ ;
26: end if
27: if ( $V$  is  $RBN$  of  $to_j$ ) then
28:    $road\_id = getRoadID(from_j, to_j)$ ;
29:    $delay = currentTime - rti_p.DH.time\_stamp$ ;
30:    $rti_p.roadTable[road\_id].delay = delay$ ;
31:   Broadcast  $rti_p$  around  $to_j$ ;
32:   for (all neighboring cells " $cell_n$ ") do
33:      $rti_p.DH.from\_junction = cell_n.first_j$ ;
34:      $rti_p.DH.to\_junction = cell_n.last_j$ ;
35:     Forward  $rti_p$  to  $GL$  of  $cell_n$ ;
36:   end for
37: end if
38: End
```

that the Junction Selection Mechanism used in GyTAR and EGYTAR has a local vision of the network traffic and takes into consideration the traffic information for the candidate roads only and not for the next roads. This leads, in some cases, to forward the data packet in or-

Algorithm 5 : PRP Algorithm

```
1: Begin
   - data_packet : The data packet
   - RH : The routing header of data_packet
   - partial_path : The partial path of RH
   - recovery_path : The recovery path of RH
   - visited_roads : The visited roads of RH
   - V : The forwarding vehicle
2: if (partial_path is not empty) then
3:   next_junction = partial_path.front();
4:   if (V around next_junction) then
5:     recovery_path.add(next_junction);
6:     partial_path.pop();
7:     visited_roads.add(current_road);
8:     if (partial_path is empty) then
9:       partial_path = compute partial path following
        algorithm.2
10:    end if
11:  end if
12:  next_junction = partial_path.front();
13:  if (next_junction ≠ NULL) then
14:    next_hop = getClosestNeighboreTo(next_junction);
15:    if (next_hop found) then
16:      forward data_packet to next_hop;
17:      return;
18:    end if
19:  end if
20: end if
21: /* at this point, the path or the next hop was not
   found */
22: RH.routing_mode = Backwards_Mode;
23: Switching to BackwardsRoutingAlgorithm;
24: End
```

der to avoid the empty areas. Moreover, the initiation of CDP packets in the both protocols GyTAR/EGyTAR depends on the movement of the vehicles. This later is not regular all the time which means that GyTAR/EGyTAR can't harness the traffic information permanently.

4.2.2. End-to-end Delay

Figure 15 illustrates the average end-to-end delay. Generally, our proposed protocol shows lower delay than GyTAR/EGyTAR, the reason behind this is that when the maximum problem occurs, GyTAR/EGyTAR use Carry And Forward strategy which consists in holding the packet and carry it thereby increasing the delay. In the contrary, in our proposed recovery mode (Backwards Forwarding Algorithm) we don't use any delayed process as is described in 3.2.3, which would

Algorithm 6 : Backwards Algorithm

```
1: Begin
   - data_packet : The data packet
   - RH : The routing header of data_packet
   - partial_path : The partial path of RH
   - recovery_path : The recovery path of RH
   - V : The forwarding vehicle
2: if (recovery_path is empty) then
3:   Drop data_packet;
4:   return;
5: else
6:   if (V around recovery_path.front()) then
7:     recovery_path.pop();
8:     partial_path = compute partial path following
        algorithm.2
9:     RH.routing_mode = PRP_Mode;
10:    Switch to PRPRoutingAlgorithm;
11:   else
12:     next_hop = getClosestNeighboreTo(recovery_path.front());
13:     if (next_hop found) then
14:       forward data_packet to next_hop;
15:       return;
16:     else
17:       Drop data_packet;
18:       return;
19:     end if
20:   end if
21: end if
22: End
```

make lower forwarding delay. As shown in Figure 15, The end to end delay is not optimized before 200 nodes because of overhead generated by the dissemination of RTIP, when the number of nodes increases the number of RTIP transmissions increases which causes more delay.

4.2.3. Routing Overhead

Figure 16 illustrates the routing overhead measured in terms of total size of control messages (we consider CDP packets size for GyTAR/EGyTAR and RTIP packets size for PBRP). As shown in Figure 16, the overhead of all protocols increases as the network density increases, this is because as the number of nodes increases, the number of control packets increases. We observe also, when the number of vehicles is less than 170 vehicles, our proposal has lower routing overhead than GyTAR/EGyTAR. This can be explained by the fact that the lower density of vehicles make more blank spaces allowing the vehicles to move quickly. Since the

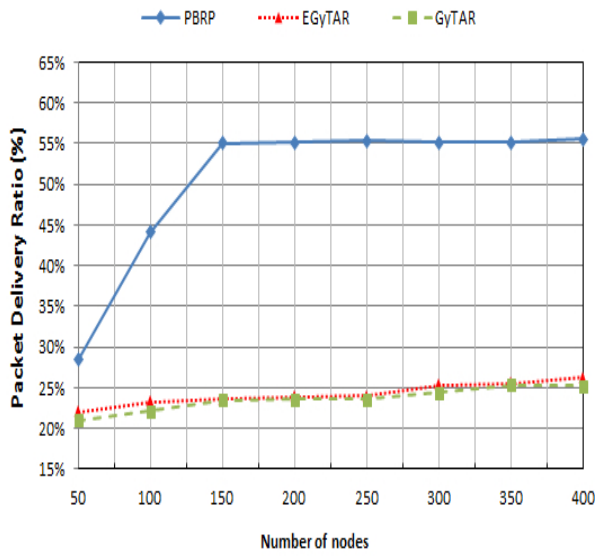


Figure 14: Packet delivery ratio

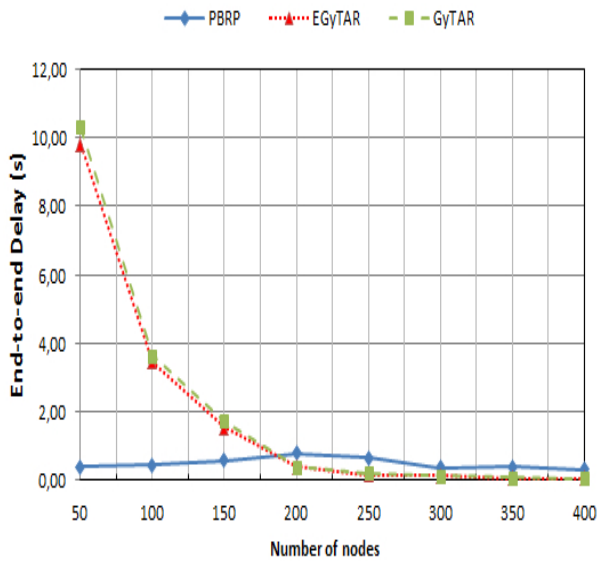


Figure 15: End-to-end delay

initiation of CDP packet in GyTAR/EGyTAR depends on mobility of vehicles, the generated CDP increases as the number of vehicles which are leaving the roads increases. In the contrary, our proposal uses a regular dissemination of traffic information. However, when the network density becomes upper than 170 nodes, the routing overhead of PBRP shows higher values due to the effects of RTIP packets that tend to travel through

multiple roads by providing an enhanced traffic information which helps the routing process to transmit more packets.

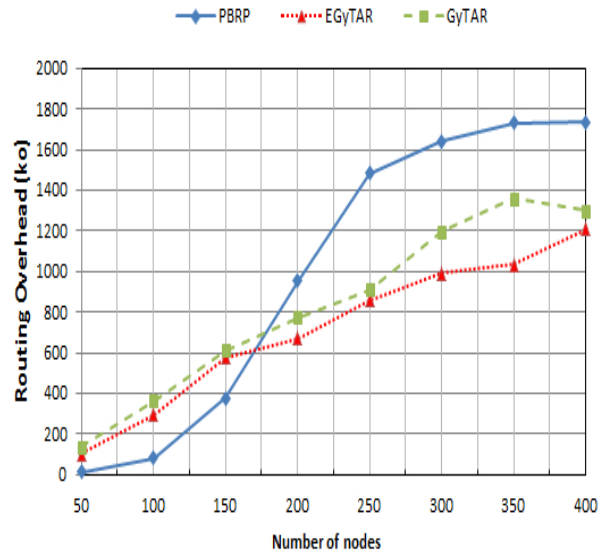


Figure 16: Routing Overhead

5. Conclusion and Future Works

In this paper we have proposed a new geographic routing protocol based on a distributed dissemination of road traffic information. This protocol provides a fresh permanent and advanced information about vehicular traffic included within the RTIP packets. We also have proposed a partial routing protocol that exploits the traffic information which is provided by DRTI to forward the data packets through partial connected paths.

We have implemented and evaluated our proposal through simulations using OMNET++ simulator. The results obtained showed that our proposal exploits the traffic information in an efficient manner and provides significant performance in terms of packet delivery ratio, end-to-end delay and routing overhead compared to GyTAR/EGyTAR protocols due to the use of an advanced vision and a permanent road traffic information thanks to DRTI and recovery mode.

As future works, we tend to optimize RTIP lifetime and RTIP initiation algorithm by using historical traffic statistics.

Table 1: Simulation Parameters

| Parameters | Values |
|---------------------------|--------------------------|
| Simulation time | 100 Sec |
| Beacon interval | 0.5 Sec |
| RTIP lifetime | 0.8 Sec |
| MAC protocol | 802.11p |
| Network area | 1500x1500 m ² |
| Transmission Range | 250 meters |
| Mobility model | Manhattan Mobility |
| Vehicle speed mean | 50 Km/h |
| Channel capacity | 6Mbps |
| Number of intersections | 16 |
| Number of roads | 24 |
| Traffic model | UDPBasicApp |
| Packet sending interval | 0.2 sec |
| Packet size | 128 bytes |
| Number of vehicles | 50-400 |
| Radio obstacle loss model | Dielectric |

References

- [1] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, 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, ACM, New York, NY, USA, 2000, pp. 120–130. doi:10.1145/345910.345931.
- [2] M. Ayaida, H. Fouchal, L. Afilal, Y. Ghamri-Doudane, A Comparison of Reactive, Grid and Hierarchical Location-based Services for VANETs, in: submitted to Vehicular Technology Conference (VTC Fall), 2012 IEEE, 2012, pp. 1–5.
- [3] M. Ayaida, M. Barhoumi, H. Fouchal, Y. Ghamri-Doudane, L. Afilal, PHRHLS: A Movement-prediction-based joint routing and Hierarchical Location Service for Vanets, in: Proceedings of IEEE International Conference on Communications (ICC'2013), Budapest, Hungary, IEEE Computer Society, 2013, pp. 1424–1428.
- [4] M. Kasemann, H. Hartenstein, M. Mauve, A Reactive Location Service for Mobile Ad Hoc Networks, Department of Computer Science University of Mannheim Tech Rep TR02014 (2002) 121–133.
- [5] H. C. E. Renault, E. Amar, S. Boumerdassi, Semi-Flooding Location Service, in: Proceedings of IEEE International Conference on Vehicular Technology, Ottawa, Canada, 2010, pp. 1–5.
- [6] B. Karp, 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), MobiCom'00, ACM, New York, NY, USA, 2000, pp. 243–254. doi:10.1145/345910.345953.
- [7] T. Nebbou, M. Lehsaini, H. Fouchal, Advanced Measurement of Road Traffic Information in City Environments, in: In Proceedings of 14th IEEE International Wireless Communications and Mobile Computing Conference (IWCMC'2018), 2018, pp. 1255–1260. doi:10.1109/IWCMC.2018.1212901.
- [8] S. Boussofouf-Lahlah, F. Semchedine, L. Bouallouche-Medjkoune, Geographic routing protocols for Vehicular Ad hoc NETWORKS (VANETS): A survey, Vehicular Communications 11 (2018) 20 – 31. doi:10.1016/j.vehcom.2018.01.006.
- [9] A. Dua, N. Kumar, S. Bawa, A systematic review on routing protocols for Vehicular Ad Hoc Networks, Vehicular Communications 1 (1) (2014) 33–52. doi:10.1016/j.vehcom.2014.01.001.
- [10] O. S. Oubbati, A. Lakas, F. Zhou, M. GneÅ, M. B. Yagoubi, A survey on position-based routing protocols for Flying Ad hoc NETWORKS (FANETS), Vehicular Communications 10 (2017) 29 – 56. doi:10.1016/j.vehcom.2017.10.003.
- [11] C. Wu, Y. Ji, F. Liu, S. Ohzahata, T. Kato, Toward Practical and Intelligent Routing in Vehicular Ad Hoc Networks, IEEE Transactions on Vehicular Technology 64 (12) (2015) 5503–5519. doi:10.1109/TVT.2015.2481464.
- [12] M. S. Daas, S. Chikhi, Optimizing geographic routing protocols for urban vanets using stigmergy, social behavior and adaptive c-n-f mechanisms: An optimized clwpr, Vehicular Communications 14 (2018) 97 – 108. doi:https://doi.org/10.1016/j.vehcom.2018.10.001.
- [13] F. Abbas, P. Fan, Clustering-based reliable low-latency routing scheme using aco method for vehicular networks, Vehicular Communications 12 (2018) 66 – 74. doi:https://doi.org/10.1016/j.vehcom.2018.02.004.
- [14] J. Gong, C. Z. Xu, J. Holle, Predictive Directional Greedy Routing in Vehicular Ad hoc Networks, in: Distributed Computing Systems Workshops, 2007. ICDCSW '07. 27th International Conference on, Toronto, Canada, 2007, pp. 2–2.
- [15] C. Lochert, H. Hartenstein, J. Tian, H. Fuessler, D. Hermann, 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.
- [16] B.-C. Seet, G. Liu, B.-S. Lee, C.-H. Foh, K.-J. Wong, K.-K. Lee, A-STAR: A Mobile Ad Hoc Routing Strategy for Metropolitan Vehicular Communications, in: N. Mitrou, K. Kontovasilis, G. N. Rouskas, I. Iliadis, L. Merakos (Eds.), Networking 2004, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 989–999.
- [17] R. K. Shrestha, S. Moh, I. Chung, D. Choi, Vertex-Based Multihop Vehicle-to-Infrastructure Routing for Vehicular Ad Hoc Networks, in: 2010 43rd Hawaii International Conference on System Sciences, 2010, pp. 1–7. doi:10.1109/HICSS.2010.435.
- [18] M. Jerbi, S. M. Senouci, T. Rasheed, Y. Ghamri-Doudane, Towards Efficient Geographic Routing in Urban Vehicular Networks, IEEE Transactions on Vehicular Technology 58 (9) (2009) 5048–5059. doi:10.1109/TVT.2009.2024341.
- [19] S. Bilal, S. Madani, I. Khan, Enhanced junction selection mechanism for routing protocol in VANETS, Int. Arab J. Inf. Technol. 8 (4) (2011) 422–429.
- [20] G. Li, L. Boukhatem, J. Wu, Adaptive Quality-of-Service-Based Routing for Vehicular Ad Hoc Networks With Ant Colony Optimization, IEEE Transactions on Vehicular Technology 66 (4) (2017) 3249–3264. doi:10.1109/TVT.2016.2586382.
- [21] R. S. Raw, D. K. Lobiyal, B-MFR routing protocol for vehicular ad hoc networks, in: 2010 International Conference on Networking and Information Technology, 2010, pp. 420–423. doi:10.1109/ICNIT.2010.5508482.
- [22] S.-H. Cha, K.-W. Lee, H.-S. Cho, Grid-Based Predictive Geographical Routing for Inter-Vehicle Communication in Urban Areas, International Journal of Distributed Sensor Networks 8 (3) (2012) 819497. doi:10.1155/2012/819497.
- [23] X. Cai, Y. He, C. Zhao, L. Zhu, C. Li, LSGO: Link State aware Geographic Opportunistic routing protocol for VANETS, EURASIP Journal on Wireless Communications and Networking 2014 (1) (2014) 96. doi:10.1186/1687-1499-2014-96.
- [24] Y. Li, A. Mohaisen, Z. Zhang, Trading Optimality for Scalability in Large-Scale Opportunistic Routing, IEEE Trans-

- actions on Vehicular Technology 62 (5) (2013) 2253–2263. doi:10.1109/TVT.2012.2237045.
- [25] I. D. Chakeres, E. M. Belding-Royer, AODV routing protocol implementation design, in: 24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings., 2004, pp. 698–703. doi:10.1109/ICDCSW.2004.1284108.
 - [26] M. Al-Rabayah, R. Malaney, A New Scalable Hybrid Routing Protocol for VANETs, IEEE Transactions on Vehicular Technology 61 (6) (2012) 2625–2635. doi:10.1109/TVT.2012.2198837.
 - [27] D. S. J. De Couto, D. Aguayo, J. Bicket, R. Morris, A high-throughput path metric for multi-hop wireless routing, Wireless Networks 11 (4) (2005) 419–434. doi:10.1007/s11276-005-1766-z.
 - [28] M. Jerbi, R. Meraihi, S.-M. Senouci, Y. Ghamri-Doudane, Gy-TAR: Improved Greedy Traffic Aware Routing Protocol for Vehicular Ad Hoc Networks in City Environments, in: Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks, VANET '06, ACM, New York, NY, USA, 2006, pp. 88–89. doi:10.1145/1161064.1161080.
 - [29] D. Jiang, L. DeGrossi, IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments, in: VTC Spring 2008 - IEEE Vehicular Technology Conference, 2008, pp. 2036–2040. doi:10.1109/VETECS.2008.458.