IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 16, 2019, accepted January 30, 2019, date of publication February 11, 2019, date of current version February 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2898776

Improvement and Performance Evaluation of GPSR-Based Routing Techniques for Vehicular Ad Hoc Networks

ANDREY SILVA[1], NIAZ REZA[2], AND AURENICE OLIVEIRA[2], (Senior Member, IEEE)
[1]Institute of Technology, Federal University of Para, Belem 66075-110, Brazil
[2]Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI 49931-1295, USA

Corresponding author: Aurenice Oliveira (oliveira@mtu.edu)

ABSTRACT Geographic routing has been widely studied over the years as an effective solution for Vehicular Ad Hoc Networks (VANETs), especially because of the availability of wireless devices and global positioning system services. Given the unpredictable behavior of VANETs, selecting the next relay node has been proved a very challenging task. Therefore, in order to maintain acceptable network performance, the routing algorithm needs to be carefully designed to adapt to the fast network changes. The Geographic Perimeter Stateless Routing (GPSR) protocol is a widely adopted position-based routing protocol for VANETs, which makes it a good benchmark candidate. In this paper, we analyze the shortcomings of GPSR and propose a new strategy named Path Aware GPSR (PA-GPSR), which includes additional extension tables in the Neighbors' Table to select the best path and bypass the nodes that have delivered such previous packets in recovery mode. Moreover, our proposed algorithm can eliminate packet routing loops avoiding the delivery of the same packet to the same neighbor node. These PA-GPSR features can, for instance, help to overcome link-breakage due to the unavoidable reasons, such as road accidents or dead-end roads. We used the Simulation of Urban MObility (SUMO) and Network Simulator-version 3 (NS-3) platform to compare our proposed algorithm to the traditional GPSR and Maxduration-Minangle GPSR (MM-GPSR) in scenarios varying the number of nodes as well as the number of source-destination pairs. Our results show that the proposed PA-GPSR strategy performed better than the traditional GPSR and MM-GPSR when packet loss rate, end-to-end delay, and network yield are considered as performance metrics.

INDEX TERMS GPSR, routing protocol, VANETs, SUMO, NS-3.

I. INTRODUCTION

Due to the increasing number of on-road vehicles, it has become extremely important to ensure driver, passenger, and pedestrian safety through research and development of efficient and reliable protocols for vehicular networks. The technological boom of wireless technologies in the last couple of decades extended the deployment of wireless communication devices on VANETs. A variety of applications such as safety or driver assistance are expected to be enabled and incorporated into vehicles thanks to the capacity of information sharing among the vehicles and infrastructure [1].

Protocol development in VANETs is a challenging task propelling researchers to try diverse strategies to solve it. Unfortunately, existing Mobile Ad-hoc Networks (MANETs) routing protocols need to have a stable route connection between the source and destination. Although, VANETs have high dynamic topology changes resulting in frequently disconnected network, and because of it, the algorithms designed to MANETs cannot be directly used in VANETs [2]. The fast changes in network topology may lead the packet to be routed to a long path and the complex environment of wireless channels may lead to a situation of channel congestion, forcing the packet to be dropped. This situation can vary according to the location, speed and direction of vehicles [3]. Therefore, maintaining a global topology for VANETs for every vehicle is very challenging [4]. Additionally, the communication links between vehicles are very short, which may degrade the performance of VANETs applications [5]. Consequently, even the most promising routing strategies for MANETs fail

The associate editor coordinating the review of this manuscript and approving it for publication was Zhaolong Ning.

to show acceptable performance when applied to VANETs. Therefore, routing algorithms in VANETs need to be developed differently than for traditional MANETs [6].

VANETs basically incorporate two major types of applications: safety applications and convenience applications (services for passengers inside the vehicles). Example of safety applications are collision warning services which give the drivers information related to a potential collision situation immediately ahead to try to reduce the number of road accidents. Convenience services, on the other hand, include entertainment and information services, such as parking lot information and file sharing [1]. Despite not as urgent as the safety applications, convenience services (usually served by unicast routing approaches) are still significant. In general, safety application messages originating from a vehicle are useful to all its neighboring vehicles or in a specific zone of interest. Therefore, the communication needs to be broadcast instead of unicast [7]. However, due to its fast topology changes and unpredictable wireless behavior, VANETs suffer from some challenges in implementing broadcast mechanism such as broadcast storm problem [8].

To address the issues of broadcast protocols, Costa *et al.* [9] proposed a distributed protocol called DDRX. This algorithm uses nodes betweenness and degree centrality to decide which neighbor will continue the dissemination process. This approach aims to reduce the broadcast storm and achieves a lower end-to-end delay. Akabane *et al.* [10] presented a protocol called Context-Aware Routing pROtocol (CARRO) for the dissemination of data in urban and highway scenarios exploring the knowledge of the geographic context. The algorithm selects vehicles located in high-priority geographic areas within its range of communication to continue the dissemination process, and when the geographic area is sparse, it makes use of the store-carry-forward mechanism.

Combining broadcast suppression and store-carry-forward mechanism to address the broadcast storm and the sparse network problem, distributed vehicular broadcast (DV-CAST) [11] was designed to be able to operate in extreme traffic regimes (sparse and dense) utilizing a list of one-hop neighbors to decide which one will re-transmit the data. However, the DV-CAST leads to an increase in the overhead since the nodes need to exchange hello messages periodically. To address this overhead problem, Khan *et al.* [12] proposed the Beacon-Less Broadcast (BL-CAST) protocol that, differently from DV-CAST, does not use hello messages. Alternatively, the BL-CAST includes the position information of the node in its broadcast message.

In this manuscript, we considered interactive entertainment applications as the type of service for our investigation. Hence, as described in [1], the data does not need to be disseminated among all the vehicles in the network. Therefore, the desirable properties of the routing protocol should be unicast instead of broadcast routing, which motivates us to use unicast routing to forward the data. Hence, we opted to use as a baseline algorithm the Geographic Perimeter Stateless Routing (GPSR), which is a widely adopted position-based unicast routing strategy [13] which exploits the positions of neighbor nodes to make decisions to forward the packets.

Different aspects of GPSR have been studied and improved in the past. Houssaini *et al.* [14] presented GPSR+PRedict which estimates vehicle movement in the near future to select the best neighbor based on its future position. Another version of GPSR, namely GPCR [15], uses nodes in the vicinity of a junction to avoid the loss of packets due to creation of local optimum. GPSRJ+ [16] further improves GPCR by taking advantage of the inherent planar nature of urban maps and focuses on improving the recovery mechanism. The authors of [17] proposed to use the speed, density and direction of the nodes to improve the greedy forwarding strategy, because the GPSR protocol tends to make inaccurate packet forwarding decisions due to its inability to consider these parameters. Greedy Curvemetric Routing Protocol (GCRP) [18] uses local geographic information about neighbors and destination (imported from a city digital map) to forward the packet to the closest neighbor in relation to the destination using curvemetric distance as parameter instead of Euclidian distance.

P-GPSR [19] aims to select the most adequate relay node in order to disseminate messages between vehicles in a probabilistic way, exploiting the beacon messages that are exchanged among vehicles and combining parameters like neighbor's speed, direction and link stability. Yang *et al.* [20] have proposed the Maxduration-Minangle GPSR (MM-GPSR), introducing a new strategy to find the next hop node in greedy forwarding through the stability of neighbor nodes T and communication area Q, using a predefined λ parameter to control the distance of the communication area. In this case, only nodes in the communication area will be able to receive packets and the node with highest stability will be selected as next hop. Yang *et al.* [20] also introduced a new concept of minimum angle to find the next hop in case of recovery forwarding. The recovery mode of MM-GPSR divides the plane in two parts based on the position of actual node and destination and uses the minimal angle value to decide which node will be selected to receive data.

In this manuscript, we are proposing an extension of our previous work [21]. Here, we introduce additional features to improve the performance of our proposed algorithm including: 1) use of information from extended Neighbors' Table to make the best possible path selection, and trying to avoid neighbors that find local maximum on its path for each destination and avoid packet loops. 2) replacement of the continuous greedy forwarding in recovery mode to right-hand and left-hand rule combined at the same time. In this work, we compared the performances of our proposed PA-GPSR with MM-GPSR and traditional GPSR, analyzing performance metrics such as packet loss rate, end-to-end delay and network yield. Our results show that our proposed algorithm can reduce the packet loss rate, end-to-end delay as well as improve network yield.

The rest of this paper is organized as follows. Section II presents the background about traditional GPSR. Section III describes our improved forwarding mechanisms. Section IV presents the simulation results and this paper is concluded in Section V.

II. BACKGROUND

In this section, we provide a short description of the Geographic Perimeter Stateless Routing (GPSR) algorithm.

A. TRADITIONAL GPSR ROUTING STRATEGY

GPSR is a typical position-based routing strategy that uses the information of the surrounding vehicles to decide which neighbor node will be selected to receive the data [22]. It makes use of two forwarding schemes for delivering the packets from the source to the destination: greedy forwarding and perimeter forwarding (recovery mode). It is assumed that every node has its own position coordinates information available via GPS and/or Short-Range Localization. Periodically, the nodes exchange this information with its one-hop neighbor through beacon messages. Therefore at any given time, every node has the position information of all of its neighbors within the communication range as well as the position of the destination through beacon messages and location service.

Based on the response of beacon messages, the actual node chooses the best neighbor which is closer to the destination according to greedy forwarding. However, if the actual node does not receive any response from a neighbor within a time-out interval, it considers the communication link as broken and deletes these entries from neighbors table. There may be some situations where there is no best neighbor than the actual node itself, which is known as local maximum condition. In this condition, GPSR can no longer maintain the greedy forwarding strategy, but rather it turns into recovery mode to forward the packet to the next node. In recovery mode strategy, all nodes follow the right-hand rule to transmit the packet to the next node. Upon receiving the packets, every node checks the packet header field, whether it is in greedy mode or recovery mode. If it is in recovery mode, the actual node checks if its distance to destination is lower than the node that entered in recovery mode (the position of the node that entered in recovery mode is available in the header of the data packet), and in this case forwards the data using greedy forwarding; if not, it keeps using recovery mode.

Consider Fig. 1 where node S wants to deliver packets to its desired destination node D. It is assumed that all nodes are equipped with GPS device which provides their own position coordinates. All nodes add their own IP with their position information in beacon messages and periodically broadcast it. The solid circle around node S indicates its communication range. Out of the two neighbors that lie within the communication range of node S, node C is the closest to the destination D and is the best option for receiving the packets. Therefore, node S sends the packets to node C according to the greedy forwarding algorithm. After receiving the packets, node C wants to forward them to its best neighbor. However, there is no node closer to the destination than node C itself

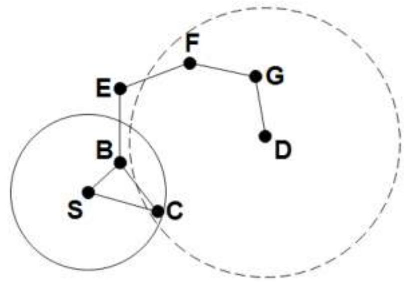

FIGURE 1. GPSR forwarding example.

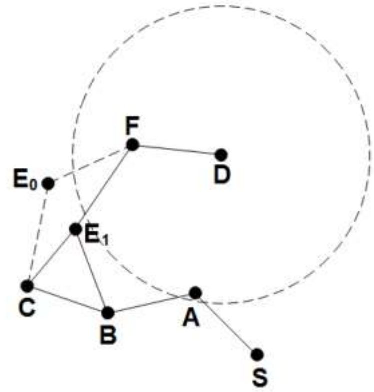

FIGURE 2. GPSR routing loop in recovery mode.

(the dotted circle shows that C is the closest node to D), causing the problem of local maximum. Recovery mode helps node C to recover from local maximum following the right-hand rule to forward packets to node B. However, node C still is closer to destination than node B. Then, it further continues forwarding packets via recovery mode and forwards them to node E. Node E sends the packets to node F (that is closer to the destination than node C). Therefore, it turns back to greedy mode and forwards the packets to node G that forwards the packets to destination D.

B. DRAWBACKS OF GPSR

Since GPSR is a geographical strategy (i.e., it defines the route based on the position of the destination and the neighbor to forward data) it can lead packets toward dead ends, increasing the end-to-end delay and the number of hops necessary to reach the destination. Besides, due to high node mobility and obstacles, the GPSR strategy may suffer with lower performance since it doesn't take into account these features.

The traditional GPSR also has drawbacks in the recovery strategy, which is depicted in Fig. 2. For instance, assume that source S wants to send packets to destination D. The node A is closer to node D, hence receives packets from node S in greedy mode. Upon receiving the packets, there is no node closer to D than A itself. The forwarding node now switches into recovery mode and the packet travels towards the destination through right-hand rule. Due to recovery mode, the packets travel through [B, C, E_0, F]. Then, node F forwards the packets to D. However, due to movement of the node, if node

E_0 comes to position E_1, it will be within the communication range of node B. Hence, in the right-hand rule, node E_1 will receive packets from node C and it will not forward the packet to node F. Rather, it will transmit the packets towards node B, and node B will forward it back towards node C. Therefore, a routing loop will be generated around these three nodes $[B, C, E_1]$ for those packets. However, for future packets, node B will send to node E_1 directly, and node E_1 will forward to F, since the position of node E_1 is now updated.

III. PATH AWARE GPSR ROUTING STRATEGY

The Path Aware GPSR strategy (PA-GPSR) that we are proposing is a Vehicle-to-vehicle (V2V) position-based routing protocol scheme (currently designed only to urban scenarios) that aims to reduce the drawbacks of GPSR discussed in Section II using a particular form of greedy and recovery forwarding. Our goal is to improve the greedy and recovery forwarding strategies of the GPSR by introducing two extensions of the Neighbors' Table (NT) called Deny Table (DT) and Recently Sent Table (RST). The DT and RST will be used by the packet forwarding decision policy.

Another contribution of this work is the replacement of the *right-hand* rule in recovery mode by a new recovery algorithm that duplicates the packet and sends it using the *right-hand* rule and the *left-hand* rule.

A. NEIGHBORS' TABLE

All vehicles periodically transmit a *hello packet* to their closest neighbors (one hop). With this *hello packet* information, the nodes create a new entry in the NT or update it. The default GPSR NT has one entry for each neighbor. Each entry has the neighbor identification (IP address), its x and y coordinates, and the time-stamp of the last received *hello packet*.

B. RECENTLY SENT TABLE AND DENY TABLE

In our approach, the NT has extensions called DT and RST. The DT is composed of two fields: the IP address of the neighbor and a vector of IP address of destinations. The main purpose of DT is to avoid a route that is not appropriate to a specific destination.

In order to handle packet loops and control the packet forwarding, we create the RST that is composed of two fields: neighbor identification (IP address) and a vector of three elements tuple: (F, I, D). The **F** element is the type of forwarding used for that packet, it can be **G** (for greedy forwarding), **L** (for left-hand forwarding) and **R** (for right-hand forwarding). The information about the type of forwarding is provided using a new field in the packet header. The **I** element is the packet identification, and the **D** element is the destination IP address. For packet identification, we use the Identification field of the IPv4 packet header, since it is a unique number (for each source-destination pair) for every packet, and the destination IP address is available in the data packet.

The entries of DT and RST are controlled by the main NT. If the entry in the main NT expires, the entries at DT and RST are also deleted. Moreover, the DT entry is refreshed at every new *hello packet* for a specific neighbor, making our algorithm self-adjustable. Therefore, our algorithm can adapt to the network changes by itself.

C. FORWARDING STRATEGY SCHEMES

1) GREEDY FORWARDING

In our *new greedy forwarding* scheme, the source node (or intermediate node) forwards the data packet to the next hop neighbor that is closer to the destination. However, this node will only be chosen if the destination for that packet is not present on its entry at DT or if that packet has not been sent to that node yet.

When a node receives a packet from its neighbor in recovery mode, the actual node will add the IP address of the destination node for that packet in the DT for this neighbor. Then, no data packet for that destination will be sent to this node (local maximum occurred) until the DT is refreshed and the destination address is removed from there. Additionally, if a neighbor node is closer to the destination and the destination for that packet is not present on DT, the node will check in RST if this packet has already been sent to this neighbor node. If it has not been sent, the neighbor node is able to receive the packet. Otherwise, the next neighbor node closer to the destination will be verified. If this condition is not satisfied even after checking all the entries in the NT, then the algorithm enters into recovery mode. Moreover, similar to GPSR, the proposed PA-GPSR algorithm also enters into recovery mode if the current node is closer to the destination than all of its neighbors and the destination is not reachable by one hop.

The PA-GPSR routing protocol is shown in detail in Algorithm 1, where: R is the node receiving a packet, N is the set of one-hop neighbors of R, n is a node of the set N, D is the destination node, d is a vector containing the distance of nodes n to D, p is a packet for D, I is the packet identification and F is the forwarding method used by the previous node.

2) RECOVERY FORWARDING

The recovery mode strategy used by PA-GPSR is based on both the right-hand rule and the left-hand rule. When a node enters into recovery mode, it will duplicate the packet and send one of them using the right-hand rule and another using the left-hand rule. The main reason behind this approach is to avoid the problem of routing path redundancy, as depending on the situation, the right-hand rule can lead to a long path to reach the destination. In another scenario, the left-hand rule can also lead to the same behavior. Since there is no way to know which one will be better for a specific situation, we create our algorithm to use both strategies. However, it will increase the network overhead, since we are duplicating packets at this point. Hence, it is necessary to create a mechanism to reduce this overhead. Therefore, our algorithm does not forward packets that are already sent using the information from the packet header and RST. This way,

Algorithm 1 Proposed Path Aware GPSR Algorithm

```
At_Receiving_Packet
if is Hello_Packet then
    n_addr = from_hello_packet_get_node_addr();
    DT_Refresh(n_addr);
else if Data Packet is in Recovery Mode then
    D_addr = from_data_packet_get_destination_addr();
    P_addr = from_data_packet_get_previous_node
    _addr();
    DT_Add(P_addr,D_addr);
end if
At_Forwarding_Data_Packet
D_addr = from_data_packet_get_destination_addr();
I = from_data_packet_get_id();
if in Greedy Mode then
    if n ∈ N && Distance (n, D) ≤ Distance (R, D) then
        d(n) = Distance (n, D);
        n_addr = from_NT_get_neighbor_node_addr();
        if DT_check(n_addr, D_addr) == false &&
        RST_check(n_addr, (F, I, D_addr)) == false &&
        d(n) == is_min_distance_to_D then
            RST_Add("G", I, D_addr); {for neighbor n}
            Forward_Packet(p, n);
        end if
    else
        Go to Recovery Mode;
    end if
else
    if n ∈ N && Distance (R, D) ≤ Distance (n, D) then
        n_right_addr = from_NT_get_right_neighbor_node
        _addr();
        n_left_addr = from_NT_get_left_neighbor_node
        _addr();
        Forward_Packet(p, n_right); {Recovery mode (right-
        hand rule)}
        Forward_Packet(p, n_left); {Recovery mode (left-
        hand rule)}
        RST_Add(n_right_addr, ("R", I, D_addr));
        RST_Add(n_left_addr, ("L", I, D_addr));
    else
        F = from_data_packet_get_forwarding_method();
        n_addr = from_NT_get_neighbor_node_addr();
        if RST_Check(n_addr, (F, I, D_addr)) == false
        then
            Forward_Packet(p, n_F); {Recovery mode using
            left-hand or right-hand rule based on F}
        else
            Discard_packet(p);
        end if
    end if
end if
```

a node that forwards a packet in right-hand mode will not forward an incoming packet that is following the left-hand rule mode, which will help to reduce the network overhead.

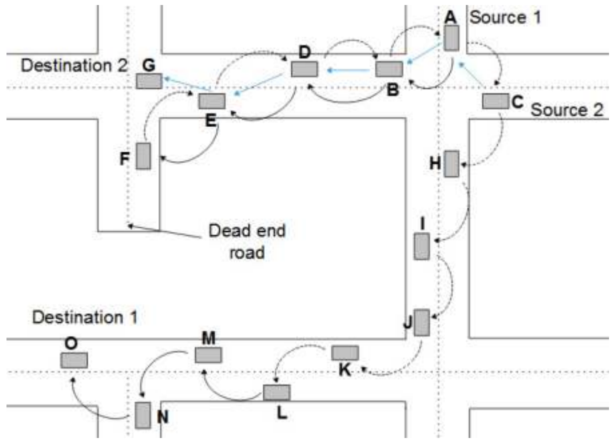

FIGURE 3. GPSR greedy forwarding leading to local maximum.

D. PA-GPSR PACKET FORWARDING EXAMPLE

1) GREEDY FORWARDING

As discussed in Section II, GPSR greedy forwarding can lead to a local maximum. In Fig. 3, the source 1 (node A) intends to forward packets to destination 1 (node O) and source 2 (node C) intends to forward packets to destination 2 (node G). Each node performs the distance calculation to select the closest node to the destination. Therefore, the packet routing path for the source-destination pair 2 involving the nodes A and O using greedy forwarding (curved solid arrows) is $P1_{greedy} = [B, D, E, F]$. This routing path will lead the packet to a dead end road. Since node F is closest to the destination than all its neighbors, the node F should use the recovery mode strategy to deliver the packet. Using the right-hand rule (curved dotted arrows), the path to reach the destination will be $P1_{recovery} = [E, D, B, A, C, H, I, J, K, L]$. Since node L is closest to the destination than node F, it returns to greedy forwarding. Therefore, node L sends the packet to node M, node M sends to node N and node N sends it to destination O. At the end, GPSR uses 17 hops to reach the destination node O. The source-destination pair involving the nodes C and G will not face any problems related to greedy forwarding (this pair will be useful to explain the DT behavior when dealing with multiple packet requests). Therefore, the routing path for this pair is always $P2_{greedy} = [A, B, D, E, G]$.

The GPSR strategy will follow this routing path for every single packet, which may contribute to increasing the end-to-end delay and lead to an unnecessary traffic overhead caused by the recovery mode (in case of the source-destination pair 1). Based on that, PA-GPSR will use the DT (one of its NT extended tables) to avoid sending packets to nodes that are delivering packets in recovery mode. Similarly to traditional GPSR, PA-GPSR will select the node with the lowest distance to O. The first packet will have the same route as in the traditional GPSR. Then, the routing path selected will be $P1_{greedy} = [B, D, E, F]$ and the recovery mode routing path is $P1_{recovery} = [E, D, B, A, C, H, I, J, K, L]$ (PA-GPSR recovery mode will also send a packet in left-hand rule from node F to node G, and node G will send the packet to node E.

Nevertheless, by the time that this packet arrives at node E, it will be discarded, since the same packet was already sent in right-hand rule to node D). However, unlike GPSR, the next packets sent by A (to the destination O) will not be forwarded to node B anymore. The main reason is A receives the first packet in recovery mode from node B. Thus, A will skip the entry of B from its NT, because the destination for this packet is now in the entry of DT for this neighbor. Therefore, the path selected for the next packets will be $P1_{greedy} = [C, H, I, J, K, L, M, N]$, for all next packets sent by A after B has been inserted into DT of node A. At the end, PA-GPSR uses 17 hops to reach the destination in the first packet. After that, however, PA-GPSR will use only 9 hops to reach the destination.

In case of incoming packets caused by multiple source requests, it will not disturb the DT and refresh it in the mean time (i.e. without properly concluding at a final efficient route for any packet) because the DT entries for each destination are independent. Therefore, node A will not skip the entry of node B for incoming packets to different destinations, showing that the system will still work normally for multiples requests. For example, the source 2 (node C) intends to send packets in greedy forwarding (straight blue arrows) to destination 2 (node G), node B now is at DT list of node A, however, since the packets are for a different destination, node A will successfully deliver all the packets to node B.

After a while, node B will send the hello message again and the DT entry of node B in node A will be refreshed removing all old information there. In this case, if the route through node F is still unreachable, node B will again be avoided. Thus, our algorithm can easily adapt to avoid local maximum.

The MM-GPSR algorithm presented in [20], on the other hand, has an unpredictable behavior for greedy forwarding strategy because it uses allowed communication area Q and the cumulative communication duration T between the nodes, which depending on the values of λ (parameter that affects the size of Q) and T will lead packets to a different route [20]. Therefore, we opted not to explore MM-GPSR greedy forwarding example.

2) RECOVERY FORWARDING

When greedy forwarding fails, GPSR will turn into recovery mode. However, depending on the adopted strategy to forward the data, it can lead to a path redundancy. The GPSR recovery mode uses *right-hand* rule to forward data as explained in Section II-A. The MM-GPSR algorithm proposed in [20] divides the plane in two parts based on the position of actual node and destination and uses the minimal angle value to decide which node will receive the data (depending on the angle value it will lead to *right-hand* or *left-hand* rule) while PA-GPSR uses packet duplication, one following the *right-hand* rule and the other following the *left-hand* rule. Fig. 4 shows two examples where depending on the strategy adopted, the routing strategy will lead to a long path to reach the destination, which contributes to the increase of end-to-end delay. Fig. 4a is the same example adopted in [20].

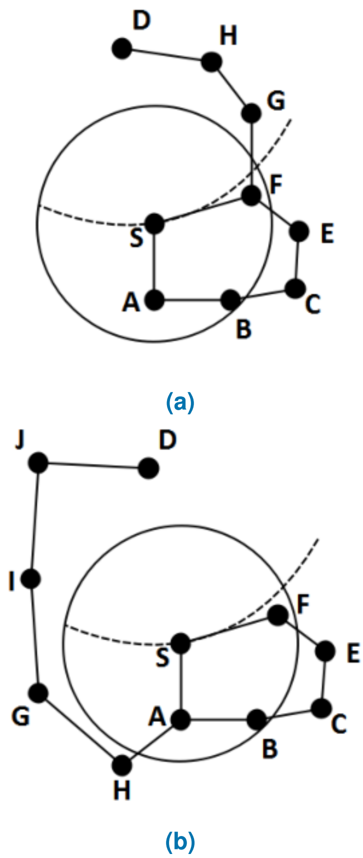

FIGURE 4. Recovery mode forwarding examples. (a) Left-hand rule as best route. (b) Right-hand rule as best route.

In both examples of Fig. 4, node S wants to send data to node D. The dotted line in Fig. 4a and Fig. 4b shows that S is closer to destination D. The circle corresponds to communication range of node S and the solid lines are the possible paths to forward packets for each node. Table 1 shows the built path, the number of hops and the total end-to-end delay for both examples illustrated in Fig. 4. For simplicity, we considered the end-to-end delay for each link transition to be increased by 10 ms. The long red arrows in Table 1 indicate that the node is delivering packets in greedy forwarding.

Analyzing Table 1, the packet always reaches the destination with the best path selected by PA-GPSR using right-hand (RH) and left-hand (LH) packets. The traditional GPSR, depending of the situation, can lead to path redundancy, since it always uses the right-hand rule to deliver the packets in recovery mode. The MM-GPSR can lead to packet loop, since it does not avoid sending data to the node that previously sent the packet.

IV. RESULTS

In this section, we evaluated the performance of our proposed PA-GPSR protocol against the traditional GPSR and the MM-GPSR using simulation-based experiments. The simulations of all routing protocols were conducted for

TABLE 1. Routing path parameters.

Algorithm	Path	Total Hops	Total Delay
	Example from Fig. 4a		
PA-GPSR (LH packet)	S $\Rightarrow$ F $\Rightarrow$ G $\Longrightarrow$ H $\Longrightarrow$ D	4	40 ms
PA-GPSR (RH packet)	S $\Rightarrow$ A $\Rightarrow$ B $\Rightarrow$ C $\Rightarrow$ E $\Rightarrow$ F $\Rightarrow$ *discarded*	-	-
GPSR	S $\Rightarrow$ A $\Rightarrow$ B $\Rightarrow$ C $\Rightarrow$ E $\Rightarrow$ F $\Rightarrow$ G $\Longrightarrow$ H $\Longrightarrow$ D	9	90 ms
MM-GPSR	S $\Rightarrow$ F $\Rightarrow$ G $\Longrightarrow$ H $\Longrightarrow$ D	4	40 ms
	Example from Fig. 4b		
PA-GPSR (LH packet)	S $\Rightarrow$ F $\Rightarrow$ E $\Rightarrow$ C $\Rightarrow$ B $\Rightarrow$ A $\Rightarrow$ *discarded*	-	-
PA-GPSR (RH packet)	S $\Rightarrow$ A $\Rightarrow$ H $\Rightarrow$ G $\Rightarrow$ I $\Longrightarrow$ J $\Longrightarrow$ D	6	60 ms
GPSR	S $\Rightarrow$ A $\Rightarrow$ H $\Rightarrow$ G $\Rightarrow$ I $\Longrightarrow$ J $\Longrightarrow$ D	6	60 ms
MM-GPSR	S $\Rightarrow$ F $\Rightarrow$ S $\Rightarrow$ F $\Rightarrow$ *loop*	-	-

different scenarios, with each scenario having different number of nodes and different pairs of source-destination. Using the available GPSR code as basis, we implemented PA-GPSR and MM-GPSR in network simulator *NS-3* (v3.23) [23]. A previous performance comparison between the GPSR and MM-GPSR was made in [20]. Thus, for consistency, we tried to use the same network configuration. However, some key parameters (for instance, number of CBR connections, beacon interval, packet rate and street structure) were not detailed in the work presented in [20]. Consequently, the results presented here may have some differences as compared to results shown in [20]. Moreover, we used Simulation of Urban MObility (SUMO) [24] to obtain the trace files corresponding to vehicle mobility while Yang *et al.* [20] used VanetMobiSim.

To facilitate the reproduction of our work, we provided an open source implementation of PA-GPSR together with the implementations of MM-GPSR and GPSR (the GPSR code is available for this version of the simulator, but some adjustments are necessary). We also provided all files used during our experiment. The simulation scenarios, scripts to read and plot the results, and the trace files obtained from SUMO are available at *github.com/CSVNetLab/PA-GPSR*.

A. SIMULATION SETUP

The simulation of vehicles was conducted in an area of 1100m^2 with 9 intersections and 12 two-way streets, as shown in Fig. 5. The initial position of vehicles was randomly distributed and the movement of vehicles on the roads was based on the Car-following model (Krauss model) restricted along the street. The vehicles moved with speed not exceeding 15 m/s.

To simulate a sparse network, we used 50, 70, 90 and 110 nodes. The *hello packet* interval was set to 1 second. Each vehicle was equipped with an omnidirectional antenna,

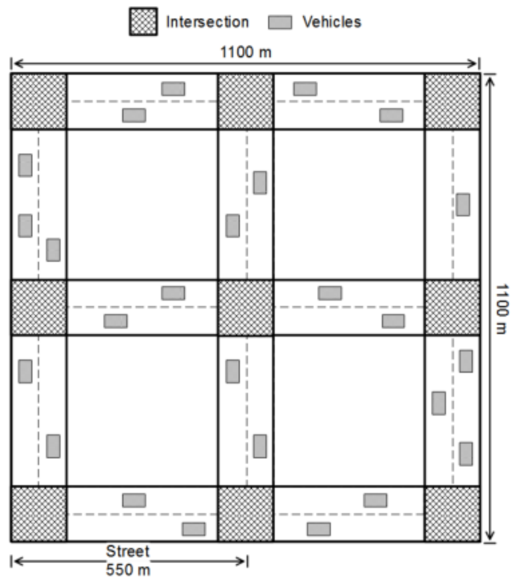

FIGURE 5. Simulation scenario with 9 intersections and 12 streets.

TABLE 2. Simulation parameters.

Parameter	Value	Unit
Simulator	NS-3/SUMO	-
Packet Size	512	byte
Simulation Time	200	s
Simulation Area	1100 x 1100	m^2
Number of CBR Connections	5, 10, 15, 20	-
Number of Nodes	30, 50, 70, 90, 110	-
Max. Speed	15	m/s
Data Type	CBR	-
Hello Interval	1	s
NT Entry Lifetime	2	s
Transport Protocol	UDP	-
Packet Interval	0.2	s
Mac Protocol	802.11p	-
Channel Data Rate	3	Mbps
Transmission Range	250	m
Propagation Model	Two-ray ground	-
Total of Simulation Runs	30	-
MM-GPSR λ Factor	0.3	-
Routing Protocol	GPSR, MM-GPSR, PA-GPSR	-

the communication range of vehicles was set to 250 meters (approximately), and the channel data rate was set to 3 Mbps. The IEEE 802.11p standard was used to model MAC layer and *Two-ray ground* radio propagation model was used to compute the wireless channel fading characteristics. We considered the data traffic to be Constant Bit Rate (CBR) for each node pairs (source-destination) to generate packets of fixed size (512 bytes). To evaluate the impact of the existing traffic in network, we varied the number of CBR connections from 5 to 20 for each scenario with different numbers of nodes. Random source-destination pairs were selected for each group of simulations. In this way, to perform the results for 5 CBR connections, we randomly selected 5 pairs and used the same pairs for all the sets of simulation runs. Moreover, the position of the nodes was available through a precise location service. Therefore, there was no error in the

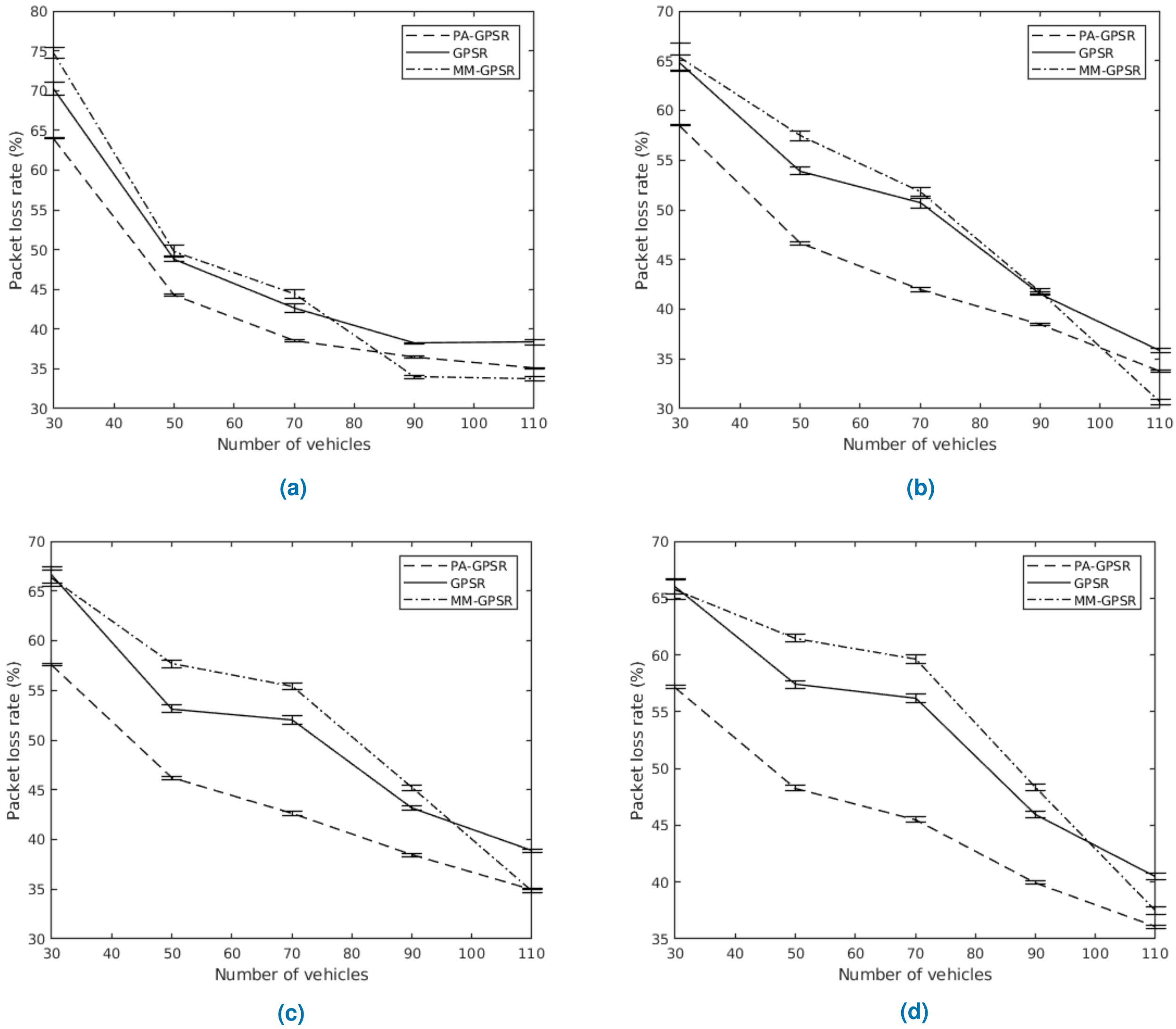

FIGURE 6. **Average packet loss rate for varying the number of vehicles and CBR connections. (a) 5 CBR connections. (b) 10 CBR connections. (c) 15 CBR connections. (d) 20 CBR connections.**

location information. We also assumed UDP as the transport layer protocol for our study.

The total time for each simulation was configured to 200 seconds. All the results shown in this paper represent an average of 30 simulation runs and a 95% confidence interval. Additional simulation parameters are summarized in Table 2. These parameters were selected based on the previous studies [3], [4], [6], [20].

The performance metrics used in our simulations are defined as follows:

- **Packet loss rate**: Represents the ratio of the total lost packets L to the total number of packets sent from the source nodes T_{source}.

$$\text{Loss rate (\%)} = \frac{L}{T_{source}} \times 100 \tag{1}$$

- **End-to-end delay**: The average value of all successfully received packets delay D_n.

$$\text{Delay} = \frac{\sum_{n=1}^{N} D_n}{N} \tag{2}$$

- **Network yield**: Ratio of the total packets received R at the destination over to the total number of packets sent by all the nodes of the network T_{all}. It measures both transmission cost as well as achieved throughput in the network.

$$\text{Net. yield} = \frac{R}{T_{all}} \tag{3}$$

B. PACKET LOSS RATE

Fig. 6 shows the packet loss rate with different numbers of nodes and CBR connections. In general, the packet loss rate

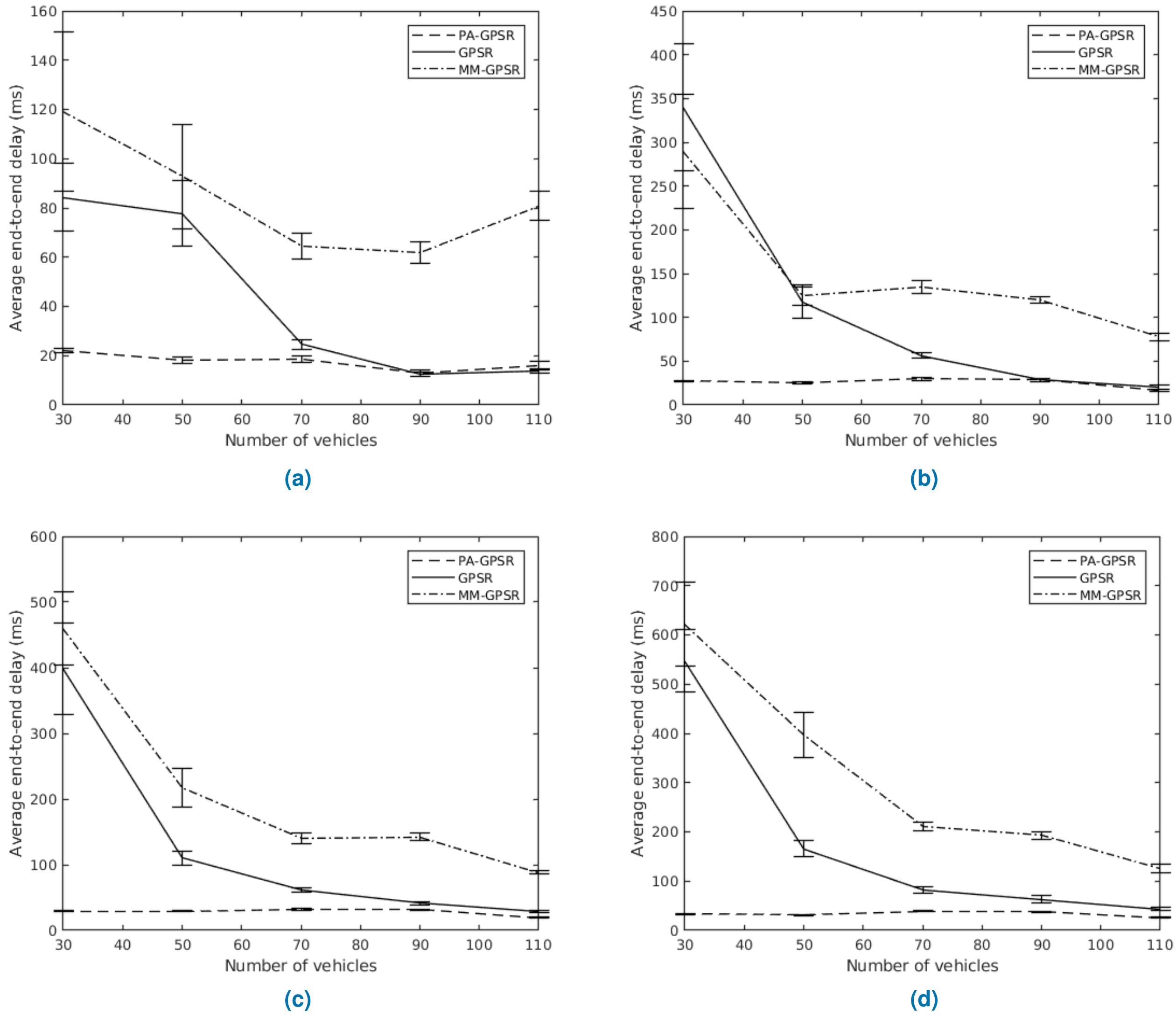

FIGURE 7. **Average end-to-end delay for varying the number of vehicles and CBR connections. (a) 5 CBR connections. (b) 10 CBR connections. (c) 15 CBR connections. (d) 20 CBR connections.**

decreases for all the three routing protocols in all scenarios. This occurs due to an increase in the number of vehicles which improve the connectivity of network and reduce the probability of encountering a network partition. In the scenario with five CBR connections, compared with GPSR and MM-GPSR, PA-GPSR has better performance of avoiding communication interruption, having smaller packet loss rate when the number of nodes are below 90, followed by GPSR. MM-GPSR performs better than PA-GPSR and GPSR when the number of nodes are 90 and 110 respectively, as shown in Fig. 6a. One reason for this lies in the fact that MM-GPSR leads to different routes to reach the destination caused by the stability parameter. For a high number of nodes, MM-GPSR may use different routes to reach destination, and it will increase the number of hops. However, it can help to reduce the channel congestion, balancing the load through the network and reducing the packet loss. Nevertheless, even in this case, the differences between PA-GPSR and MM-GPSR results are very small ($< 3\%$).

In the scenario with 10 CBR connections, PA-GPSR performs better than MM-GPSR and GPSR for almost all number of nodes as shown in Fig. 6b. Compared to the results in Fig. 6a, GPSR and MM-GPSR have an increase in the packet loss rate, while PA-GPSR achieves almost the same performance. Due to the availability of more source-destination pairs, the chances to have more packets following a long path (due to a wrong decision of the recovery mode) increase. Since MM-GPSR and GPSR do not have a robust recovery mode, they are more susceptible to these problems than PA-GPSR when the number of pairs increase. In this scenario, MM-GPSR only performs better than PA-GPSR in scenario with 110 nodes with a difference of around 5%. Fig. 6c and Fig. 6d illustrate the scenarios with 15 and 20 CBR connections, respectively. In general, the results

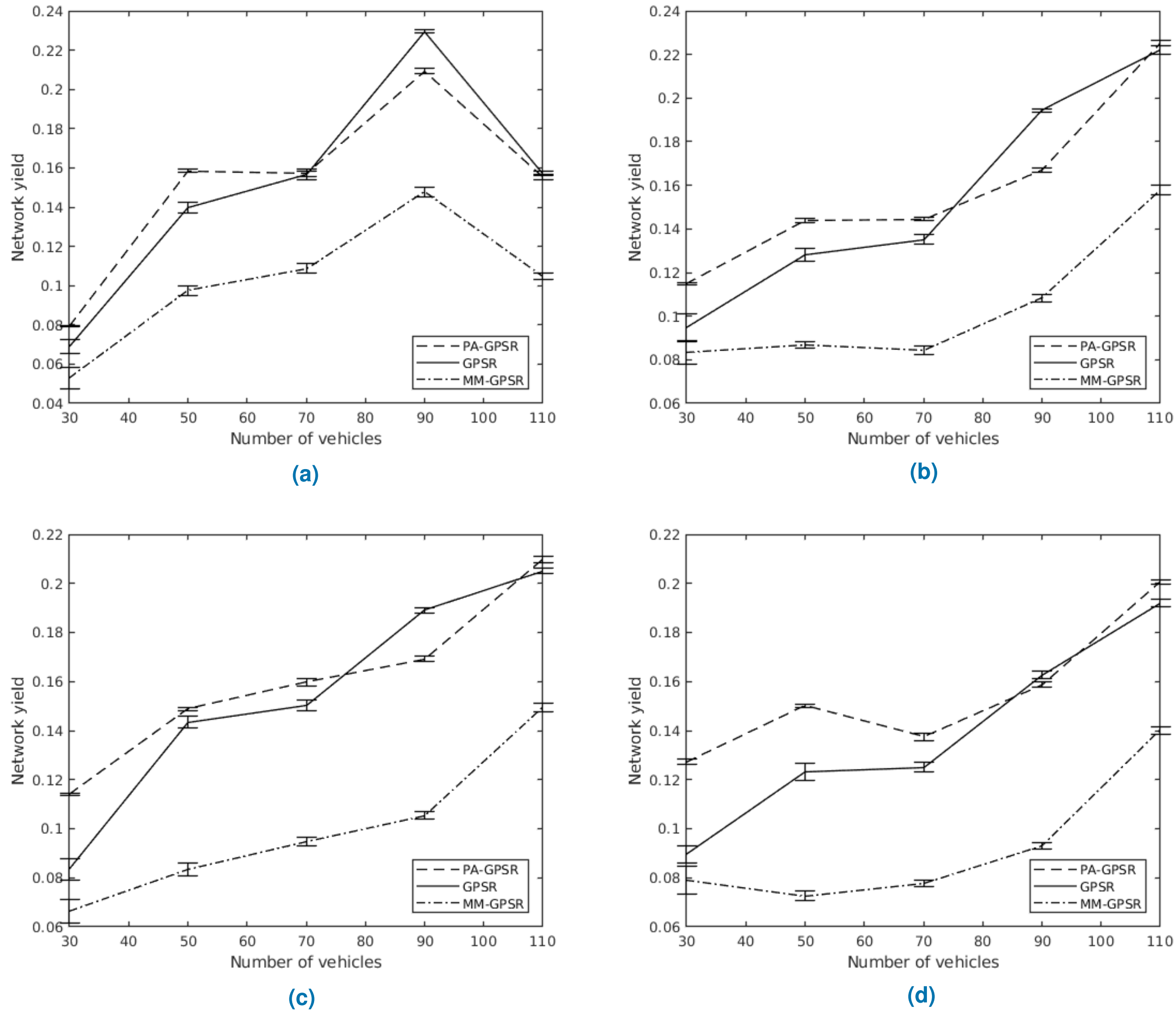

FIGURE 8. **Average network yield for varying the number of vehicles and CBR connections. (a) 5 CBR connections. (b) 10 CBR connections. (c) 15 CBR connections. (d) 20 CBR connections.**

are very similar. For these scenarios, PA-GPSR outperforms MM-GPSR and GPSR in all results involving the number of nodes (with exception of the results with 110 nodes and fifteen CBR connections, where the results are very similar).

A reasonable justification for the fact that the PA-GPSR is performing better than GPSR and MM-GPSR is that PA-GPSR has packet loop control. Packet loops can occur in a situation where a small number of moving nodes exist, without this control, any algorithm will face an increase in the packet loss rate, because every packet has a time limit to stay in the network.

C. END-TO-END DELAY

Fig. 7 shows the comparison of end-to-end delay with different number of nodes and CBR connections. In general, when the number of nodes increase, the end-to-end delay decreases for MM-GPSR and GPSR (with exception of Fig. 7a and Fig. 7b, where it first decreases and then rises again for MM-GPSR). The PA-GPSR algorithm keeps the same result for all the scenarios. PA-GPSR and GPSR clearly have smaller end-to-end delay in comparison to MM-GPSR. As shown in Fig. 7a, compared to GPSR, PA-GPSR has smaller end-to-end delay when the number of nodes are smaller than 90 for five CBR connections.

Analyzing Fig. 7b, Fig. 7c and Fig. 7d, the end-to-end delay for MM-GPSR and GPSR increases when the number of source-destination pairs increase. Analyzing only the values of MM-GPSR maximum end-to-end delay, we observed 120, 290, 450 and 600 ms, approximately; while the values for GPSR are 85, 340, 400 and 550 ms approximately. However, the PA-GPSR values are 22, 30, 31 and 39 ms approximately, clearly demonstrating the PA-GPSR stability when compared with GPSR and MM-GPSR. This can be explained based on the loop control method (detailed in the previous results section) and the recovery mode of PA-GPSR. Since the PA-GPSR recovery mode uses both right-hand and left-hand rule, it will lead the packet to find the best route to the destination regardless of which side the destination is (left or right), while the minimal angle in MM-GPSR and the right-hand rule in GPSR can lead to path redundancy (caused by wrong choice of next hop), increasing the end-to-end delay. MM-GPSR also uses a λ factor equal to 0.3 to determine the communication distance d_{max} to forward packets. Therefore, using a fixed value of λ may not be suitable to lead to a good path to destination, increasing the end-to-end delay.

Another important aspect that must be taken into account is the fact that end-to-end delay can only be calculated if the packet is received at the destination node. Therefore, an algorithm can have a small end-to-end delay when packets with higher delays are dropped from the network. Hence, it is important to make a cross reference between the end-to-end delay and the number of packets lost. If an algorithm has a low value of end-to-end delay and the destination node receives more packets (or the lowest packet loss rate) in comparison to other algorithms, it means that this algorithm indeed has the minimal latency.

D. NETWORK YIELD

Fig. 8 illustrates the network yield for different number of nodes and CBR connections. The network yield is equivalent to the goodput of the network. It takes into consideration the routing performance of the packet delivery ratio and the throughput of the whole network. With exception of results shown in Fig. 8a for 110 nodes (the small number of CBR connections can lead to an unstable result), when the number of nodes increase (and also the number the CBR connections), the network yield of all the three routing protocols increases. With the increase in the number of vehicles, the network connectivity also increases since all three algorithms have greedy forwarding (or a variation of it) as their main forwarding strategy. In fact, the greedy forwarding is very suitable for this situation (well connected network) as it will reduce the number of hops to reach the destination, increasing the packet delivery ratio and increasing the network yield as well.

Our proposed PA-GPSR algorithm has higher network yield in comparison with MM-GPSR and GPSR for all set of pairs of source-destination as shown in Fig. 8. The only exception occurs when the GPSR has bigger network yield in the scenarios with numbers of vehicles equal to 90. Two possible reasons for this behavior are: 1) the right-hand rule is the best path option to reach the destination. 2) there is no situation where the DT is useful to reduce the number of hops. In this case, the greedy forwarding of the PA-GPSR acts exactly as GPSR greedy forwarding. Therefore, since PA-GPSR uses packet duplication in recovery mode (even reaching the destination with the same number of hops and same packet delivery ratio), the network yield of PA-GPSR tends to be higher than GPSR.

MM-GPSR has a smaller network yield than PA-GPSR and GPSR. Analyzing Fig. 6 and Fig. 7 we can observe that MM-GPSR has a higher packet loss rate and higher delay values, respectively. We can infer that MM-GPSR is reaching the destination with a higher number of hops than PA-GPSR and GPSR. These two aspects (high packet loss error and high number of hops) contribute to reducing the network yield of MM-GPSR.

V. CONCLUSION

The unpredictable behavior of the vehicular networks with fast topology changes, high speed nodes, limited transmission range and limited wireless channel make the designing of routing protocols in VANETs very challenging. In this paper, we described how the selection of the next hop in greedy forwarding can be improved by exploiting the information of the neighbors. We also improved the recovery mode using packet duplication. Our proposed algorithm has two new tables as an expansion of NT to avoid packet loop and nodes that are delivering packets in recovery mode for a specific destination. The proposed algorithm was simulated successfully in an urban sparse scenario in NS-3. The experiments demonstrate that the proposed protocol shows better performance over traditional GPSR and MM-GPSR protocols as far as packet loss rate, end-to-end delay and network yield are concerned. To facilitate the reproduction of our results, we provided an open source implementation of PA-GPSR and its competitors (GPSR and MM-GPSR) at *github.com/CSVNetLab/PA-GPSR*. For future implementation, we plan to modify our algorithm considering different parameters such as speed, direction and node density. We also plan to compare our algorithm in a dense urban scenario and in a highway scenario to have more realistic simulations. Additionally, we intend to use a GPS error model instead of considering perfect location information for the nodes.

REFERENCES

[1] F. Cunha *et al.*, "Data communication in VANETs: Protocols, applications and challenges," *Ad Hoc Netw.*, vol. 44, pp. 90–103, Jul. 2016.

[2] P. Ranjan and K. K. Ahirwar, "Comparative study of vanet and manet routing protocols," in *Proc. Int. Conf. Adv. Comput. Commun. Technol. (ACCT)*, 2011, pp. 517–523.

[3] X. M. Zhang, K. H. Chen, X. L. Cao, and D. K. Sung, "A street-centric routing protocol based on microtopology in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 7, pp. 5680–5694, Jul. 2016.

[4] X. Zhang, X. Cao, L. Yan, and D. K. Sung, "A street-centric opportunistic routing protocol based on link correlation for urban VANETs," *IEEE Trans. Mobile Comput.*, vol. 15, no. 7, pp. 1586–1599, Jul. 2016.

[5] A. K. Ali, I. Phillips, and H. Yang, "Evaluating VANET routing in urban environments," in *Proc. 39th Int. Conf. Telecommun. Signal Process. (TSP)*, Jun. 2016, pp. 60–63.

[6] N. Li, J.-F. Martínez-Ortega, V. H. Díaz, and J. A. S. Fernandez, "Probability prediction-based reliable and efficient opportunistic routing algorithm for VANETs," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1933–1947, Aug. 2018.

[7] R. Fracchia, M. Meo, and D. Rossi, "VANETS: To beacon or not to beacon?" in *Proc. Autonet IEEE Globecom*, Nov. 2006, pp. 1–9.

[8] C. Suthaputchakun and Z. Sun, "Multihop broadcast protocol in intermittently connected vehicular networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 2, pp. 616–628, Apr. 2018.

[9] J. Costa, D. Rosario, A. M. de Souza, L. A. Villas, and E. Cerqueira, "Protocolo para disseminação de dados em VANETs baseado em métricas de redes complexas: Um estudo de caso com sistema de gerenciamento de trânsito," *Simpósio Brasileiro Redes Comput.*, vol. 36, pp. 1065–1078, May 2018.

[10] A. T. Akabane, R. W. Pazzi, E. R. Madeira, and L. A. Villas, "CARRO: A context-awareness protocol for data dissemination in urban and highway scenarios," in *Proc. 8th IEEE Latin-Amer. Conf. Commun. (LATINCOM)*, Nov. 2016, pp. 1–6.
[11] O. K. Tonguz, N. Wisitpongphan, and F. Bai, "DV-CAST: A distributed vehicular broadcast protocol for vehicular ad hoc networks," *IEEE Wireless Commun.*, vol. 17, no. 2, pp. 47–57, Apr. 2010.
[12] A. Khan, J.-C. Nam, and Y.-Z. Cho, "Beacon-less broadcast protocol for vehicular ad hoc networks," in *Proc. 19th Asia–Pacific Conf. Commun. (APCC)*, Aug. 2013, pp. 153–154.
[13] A. N. Vigilia and J. S. Suseela, "Survey on unicast, multicast and broadcast routing techniques in vehicular ad-hoc networks–present and future," *Brit. J. Math. Comput. Sci.*, vol. 13, no. 4, pp. 1–26, 2016.
[14] Z. S. Houssaini, I. Zaimi, M. Oumsis, and S. E. A. Ouatik, "GPSR+Predict: An enhancement for GPSR to make smart routing decision by anticipating movement of vehicles in VANETs," *Adv. Sci. Technol. Eng. Syst. J.*, vol. 2, no. 3, pp. 137–146, 2017.
[15] C. Lochert, M. Mauve, H. Füßler, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 9, no. 1, pp. 69–72, 2005.
[16] K. C. Lee J. Härri, U. Lee, and M. Gerla, "Enhanced perimeter routing for geographic forwarding protocols in urban vehicular scenarios," in *Proc. IEEE Globecom Workshops*, Nov. 2007, pp. 1–10.
[17] D. Xiao, L. Peng, C. O. Asogwa, and L. Huang, "An improved GPSR routing protocol," *Int. J. Adv. Comput. Technol*, vol. 3, no. 5, pp. 132–139, 2011.
[18] T. Nebbou and M. Lehsaini, "Greedy curvemetric-based routing protocol for VANETs," in *Proc. Int. Conf. Sel. Topics Mobile Wireless Netw. (MoWNeT)*, Jun. 2018, pp. 1–6.
[19] S. Dahmane and P. Lorenz, "Weighted probabilistic next-hop forwarder decision-making in VANET environments," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.
[20] X. Yang, M. Li, Z. Qian, and T. Di, "Improvement of GPSR protocol in vehicular ad hoc network," *IEEE Access*, vol. 6, pp. 39515–39524, 2018.
[21] A. Silva, K. M. N. Reza, and A. Oliveira, "An adaptive GPSR routing protocol for VANETs," in *Proc. 15th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2018, pp. 1–6.
[22] P. Zhou, X. Xiao, W. Zhang, and W. Ning, "An improved GPSR routing algorithm based on vehicle trajectory mining," in *Proc. Int. Conf. Geo-Spatial Knowl. Intell.* Springer, 2017, pp. 343–349.
[23] *The NS-3 Network Simulator*. Accessed: Aug. 15, 2018. [Online]. Available: http://www.nsnam.org/
[24] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO—Simulation of urban mobility," *Int. J. Adv. Syst. Meas.*, vol. 5, nos. 3&4, pp. 128–138, Dec. 2012.

ANDREY SILVA received the B.Sc. degree in computer engineering, and the M.Sc. degree in electrical engineering from the Federal University of Pará, Brazil, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering. He is in a visiting Scholar Program with CSVNet Laboratory, Michigan Technological University, Houghton, MI, USA. His current research interests include vehicular communications, machine learning, and digital signal processing.

NIAZ REZA received the B.Sc. degree in electrical, electronic and communication engineering from the Military Institute of Science and Technology of Bangladesh, in 2015. He is currently pursuing the Ph.D. degree in electrical engineering from Michigan Technological University, Houghton, MI, USA. In 2015, he joined the EECE Department of Military Institute of Science and Technology, as a lecturer, and has served for two years. His research interests are mainly focused on vehicular ad-hoc networks, wireless communications, and connected vehicles.

AURENICE OLIVEIRA (SM'15) received the B.Sc. degree in electrical engineering from the Federal University of Bahia, Brazil, in 1995, the M.Sc. degree in electrical engineering from the State University of Campinas, Brazil, in 1998, and the Ph.D. degree in electrical engineering from the University of Maryland, College Park, MD, USA, in 2005. She is currently an Associate Professor with the Department of Electrical and Computer Engineering, Michigan Technological University. Her current research interests include vehicular communications, intelligent transportation systems, communication networks and systems, network security, signal processing, and engineering education. She is a Senior Member of the Society of Automotive Engineers and the American Society for Engineering Education. She serves on the Editorial Board of the *International Journal of Engineering Research and Innovations*, and as the ABET-EAC Program Evaluator.