2018 15th International Symposium on Wireless Communication Systems (ISWCS)

An Adaptive GPSR Routing Protocol for VANETs

Andrey Silva†‡, K. M. Niaz Reza† and Aurenice Oliveira†
Department of Electrical and Computer Engineering
†Michigan Technological University, Houghton-MI, USA
‡Federal University of Para, Belem, Brazil
Email: {atorresd, kreza, oliveira}@mtu.edu

***Abstract*—The fast topology change and high-speed mobility of vehicles, as well as the limited radio range, usually lead to wrong packet forwarding decisions in highly dynamic Vehicular Ad Hoc Network (VANET). This type of environment makes data routing very challenging. Position-based routing protocols are popular for VANET due to the availability of GPS devices. Geographic Perimeter Stateless Routing (GPSR) has been widely adopted to cope up with VANET challenges. Nevertheless, there are still improvements that could be incorporated into GPSR to make it more reliable and efficient. In this paper, we describe an Adaptive GPSR (AGPSR), including additional information in the Neighbors Table to select the best path and bypass the nodes that delivered the previous packets in recovery mode. This approach can avoid possible link-breakage due to for instance a road accident. We compared our results with the traditional GPSR using the Simulation of Urban MObility (SUMO) and Network Simulator-version 3 (NS-3) for both static and mobility scenarios. Our results show that the proposed AGPSR strategy has better performance than traditional GPSR when packet delivery ratio, lost packets and hop count are used as performance metrics.**

***Index Terms*—GPSR, Adaptive, VANET, SUMO, NS3**

I. INTRODUCTION

Today, the number of vehicles on the roads is vast, especially in developed countries. Vehicular Ad Hoc Networks (VANETs) provide a critical structure to improve road safety, traffic efficient and eco-friendly transportation, and infotainment. Unlike stand-alone and autonomous systems, VANETs provide the supporting structure for communication among vehicles (nodes). They enable data sharing of vehicle driving conditions in timely and accurate fashion. It is predicted that inter-vehicle communication will become a significant component in autonomous vehicles [1].

VANET includes both vehicle-to-vehicle (V2V) communication and vehicle-to-infrastructure (V2I) communication based on IEEE 802.11p wireless radio interface [2]. Large-scale urban VANETs have several challenges such as rapid topology change due to fast movement of vehicles, which are often constrained by the road structure. In urban areas, there are also large tall buildings alongside the roads as well as junctions acting as transmission obstacles. As a result, communication data is transmitted through multipath channels, reducing the system performance. One also needs to consider large variations in the communication channel characteristics when considering different environments such as highways and urban scenarios. Urban VANET topology is complex and challenging to work with due to the presence of buildings, trees, variable node density, the distance between nodes, among others. If vehicle density is low in a given area, the implication is a frequent break in connectivity between nodes [3]. Hence, it is critical to design robust and efficient routing protocols able to overcome these challenges.

Position based routing (PBR) protocols also known as Geographic routing do not establish routing tables or store routes for the entire network. Most PBR assumes that vehicles are equipped with a GPS device [4]. These protocols select the next-hop neighbor based on position information of their neighboring nodes, destination nodes, and their own position [5]. Our work is based on Greedy Perimeter Stateless Routing (GPSR) [6], which is a typical example of PBR protocol. It forwards the packets to the neighboring node which is geographically closer to the destination. Greedy Perimeter Coordinator Routing (GPCR) [7] considers planarized graphs at street intersections to select the next-hop neighbor. Geographic Perimeter Stateless Routing Junction+ (GPSRJ+) [8] is an improvement of GPCR that predicts the route of the junction node to minimize the speed factor effects on GPSR protocol.

In this paper, we propose to use additional information from neighbor nodes to make the best possible path selection, trying to avoid neighbors that find local maximum on its path. Our proposed algorithm is referred to as Adaptive GPSR or AGPSR. It is a scheme that uses a new field on Neighbors Table (NT) to help to select the next hop to forward packets. By bypassing nodes that are delivering packets in recovery mode, AGPSR can increase the packet delivery ratio and reduce the number of hops, end-to-end delay and packet loss.

II. BACKGROUND

In this section, we provide a short description of the GPSR algorithm.

A. Traditional GPSR Routing Strategy

GPSR Routing Protocol is stateless and quickly adapts to the change in network topology. GPSR can work on two modes: *greedy forwarding* and *recovery mode (perimeter forwarding)*. It assumes that every node has the information about its position coordinates through GPS and Short-Range Localization. The source node attaches the destinations' locations to the packets. Every node periodically transmits *hello packets* containing its IP and position and collects information about its one-hop neighbors.

978-1-5386-5005-9/18/$31.00 ©2018 IEEE

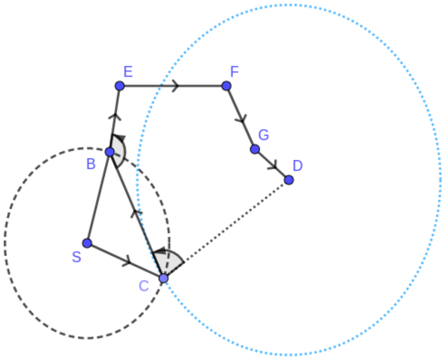


Fig. 1: GPSR Forwarding Example.

In Fig. 1, node S wants to transmit packets to destination D. It first enters into greedy mode and finds nodes C and B as its neighbor through the response of the *hello packets* (the black dotted circle shows the range of S). However, among the two neighbor nodes, node C is closer to node D and hence node S forwards the packets to node C through greedy forwarding. At node C, there is no neighbor node closer to D than C itself (node C is in the perimeter of the blue dotted circle around node D). This refers to as *local maximum*. Hence it turns into recovery mode and follows the right-hand rule to forward the packets to node B. Similarly, at node B, there is no node closer to D than B itself. Thus, it stays in recovery mode and forwards the packets to node E. Upon receiving the packets, E finds out that node F is closer to D. Hence, it returns into greedy mode and forwards the packets to F. Similarly, G receives the packets and finally transmits to destination D.

B. Drawbacks of GPSR

GPSR has a few drawbacks which at times makes it unsuitable for highly mobile VANET scenario. It forwards the packets in the greedy mode based on the position of the neighboring nodes. The original GPSR does not take into account the vehicle speed and direction. However, the high speed of vehicles changes the position of the nodes which eventually turns the greedy forwarding inaccurate. This strategy causes a significant amount of packet drops. Therefore, choosing the best possible neighbor is critical to increasing the Packet Delivery Ratio (PDR).

Another drawback is graphically shown in Fig. 2. In this example, node A wants to transmit the packets to node J through GPSR protocol. However, by the time node D receives the packets, it faces transmission obstacles and can not deliver to node J through greedy forwarding (blue arrow). Then it turns into recovery mode (red arrow) and follows the right-hand rule. Thus, the packets follow the path $P = [D, C, B, A, E, F, G]$. Node G receives the packets and finds that node H is closer to destination node J than node G itself. Hence it returns into greedy mode and transfers to node H. Similarly, node H transfers to node J. However, in this strategy, the number of hop counts is increased because every new packet will follow this same path. In this paper, we propose a forwarding decision based on bypassing the nodes that delivered the packets in recovery mode. Our proposed scheme adapts with the scenario and consequently achieves an overall better performance than the traditional GPSR.

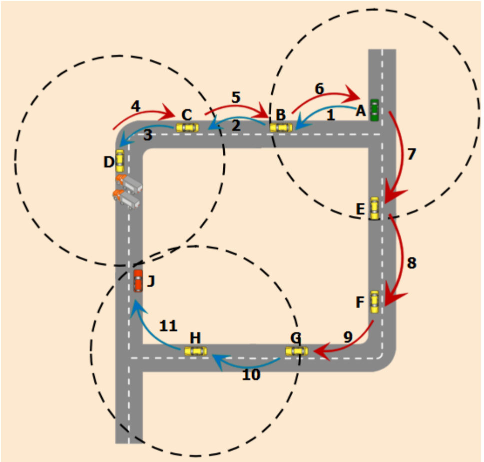


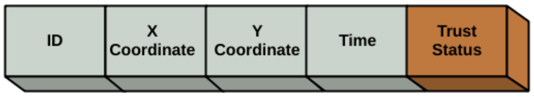
Fig. 2: GPSR Drawback in a Jammed Traffic.

III. ADAPTIVE GPSR ROUTING STRATEGY

The adaptive GPSR strategy (AGPSR) that we are proposing is a position-based routing scheme that aims to reduce the drawbacks of GPSR illustrated in Section II and in Fig. 2 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 a new parameter in the neighbors' list: *neighbors trust status*. The trust status will be used by the packet forwarding decision policy for greedy mode. Another contribution of this work is the replacement of the right-hand rule in perimeter mode by a new recovery algorithm called *continuous greedy mode*.

A. Neighbors Table (NT)

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 their table. 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*. In our approach, the NT has a new field called *trust status*. In Fig. 3, we show the new NT with the new field in brown. The default GPSR NT fields are in gray.



ID	X Coordinate	Y Coordinate	Time	Trust Status

Fig. 3: Neighbors Table with Default and New Fields.

B. Forwarding Strategy Schemes

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 trust status field is equal to zero. When a node receives a packet from its neighbor in recovery mode, the trust status field for this neighbor is set to one. Then, no data packet will be sent to this node (local maximum occurred) until the trust status field returns to zero. In this case, the second node closer to the destination will be chosen. However, only if the trust status field is equal to zero. This process will continue until a node that satisfies this condition is reached. If this condition is not satisfied even after checking all the entries in the NT, then the algorithm enters in recovery mode. Moreover, similar to GPSR, the proposed AGPSR algorithm also enters in 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. We are extending our investigation to use a probabilistic trust status field instead of a deterministic one. In this way, we believe we will be able to solve problems related to the role of different destination zones. When using only the deterministic trust status for all the nodes, the skipping of one neighbor node may affect the route selection for other destinations.

The trust status field returns to zero when the node sends a *hello packet* because every new *hello packet* resets the trust status of this node in the neighbors NT, making our algorithm self-adjustable. Therefore, our algorithm can adapt to the network changes by itself. The details of AGPSR routing protocol is shown 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 distance vector of nodes n to D, p is a packet for D and h is a *hello packet*.

Algorithm 1 Proposed Adaptive GPSR algorithm.

```
At_Receiving_Packet
if is Hello_Packet && $n \in N$ then
    trust_status = 0;
else if Data Packet is in Recovery Mode && $n \in N$ then
    trust_status = 1;
end if
At_Forwarding_Data_Packet
if $n \in N$ && Distance ($n$, $D$) $\leq$ Distance ($R$, $D$) then
    $d(n)$ = Distance_to_D;
    if trust_status = 0 && $d(n)$ = is_min_distance_to_D then
        Forward_Packet($p$, $n$);
    else if $n \in N$ && $n$ is not the previous sender node && $d(n)$ = is_min_distance_to_D || size($N$) = 1 then
        Forward_Packet($p$, $n$); {Recovery mode}
    end if
else if $n \in N$ && $n$ is not the previous sender node && $d(n)$ = is_min_distance_to_D || size($N$) = 1 then
    Forward_Packet($p$, $n$); {Recovery mode}
end if
```

The *recovery mode strategy* (or *continuous greedy*) used by AGPSR is very similar to greedy mode. However, the difference is that the node skips the entry at NT of the neighbor who sent the packets and doesn't take into account the *trust status* field when it will perform the next hop selection. As per this rule, if node A has two neighbors B and C and receives a packet from node B, and there is no route to destination, it enters in recovery mode and sends the packet through its next neighbor C, even if B is the closest node to reach the destination. In this case, if node A receives a packet from node B and has no other nodes to send the packet than node B, it will send back to B.

A calculation based on the GPSR recovery mode using the NT is performed to discover the node that sent the packet in recovery mode. In AGPSR, instead of using the destination position to calculate the angle, our algorithm uses the previous node's position. Therefore, the node that sent the packet is the node that has the minimal angle value. In static nodes, this value should be zero. However, for non-static nodes, this value should be close to zero (because of node mobility). We use Fig. 4 to illustrate an example of how AGPSR discovers the node that sent the packet.

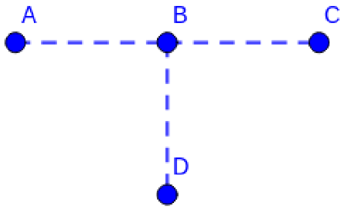


Fig. 4: Neighbor Discovery Based on Angle.

Assuming that node B receives a packet from node C. Node B needs to know among the three neighbors (A, C and D) which one sent the packet. To find that out, node B will perform an angle calculation based on the *Previous Position* information of the packet header. The previous position is the (x,y) position of node C. Based on these calculations node B will conclude that for nodes A, D and C, the angles are 180, 270 and 0, respectively. Hence, the node that sent the packet was node C.

C. AGPSR Packet Forwarding Example

In Fig. 2, the source A intends to forward packets to node J. Like in GPSR, the source and neighbor nodes exchange hello messages providing mobility information. The source node performs the distance calculation to choose the next hop. The node with lower distance to J will be selected as next hop, in this case, node B. The first packet will follow exactly as in the traditional GPSR. Then, the path selected will be $P(1) = [A, B, C, D, C, B, A, E, F, G, H, J]$, where $P(1)$ indicates the path made by the first packet to reach the destination. However, unlike GPSR, the next packets sent by A 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 *trust status* field for this node is non-zero. Therefore, the path selected for the next packets will be

$P(n) = [A, E, F, G, H, J]$, where n is the next packets sent by A after B had the *trust status* entry in the NT of node A marked as one. After a while, node B will send the hello message again and the *trust status* entry in the NT of node A will be reset to zero. In this case, if the route to node J still is unreachable, node B again will be avoided, and this will occur until the route to node J passing by node B works again. Thus, our algorithm can easily adapt to traffic jamming.

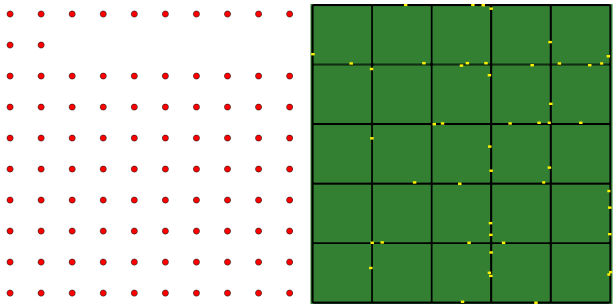
IV. RESULTS

In this section, we conduct simulation-based experiments to evaluate the performance of the proposed protocol against GPSR with the use of Simulation of Urban MObility (SUMO) [9] and Network Simulator-version 3 (NS-3) [10]. We obtained the trace files corresponding to vehicle mobility from SUMO, converted these files to NS3-compatible files, and used them for network simulation.

A. Simulation Setup

The simulation of vehicles is conducted in two urban scenarios using static and moving nodes in a Manhattan-grid of 1000x1000m, as shown in Fig. 5. In urban static nodes scenario (Fig. 5a), equally spaced 100 nodes are used. We would like to point out that we chose this relatively simplified scenario as proof of concept for our proposed protocol. For more realistic results, the scenarios should be extended using randomly placed static nodes and real urban topologies. We are currently adding new scenarios to our study. At the beginning of the simulation, the nodes that are in row 2 and columns 3 to 10 disappear from the range of all the other nodes to simulate a traffic jam. Therefore, the real number of nodes for this simulation scenario is 92. In this scenario, node 1 (row 1, column 1) sends packet to node 30 (row 3, column 10). The *hello packet* interval is set to 1 second. The communication range of vehicles is set to 100 meters, so the nodes at diagonal are not neighbors. The IEEE 802.11p standard is used to model MAC layer and *Two-ray ground* radio propagation model is used to compute the wireless channel fading characteristics. We consider the data traffic to be Constant Bit Rate (CBR) that is attached to each source node to generate packets of fixed size (200 bytes). A single pair source-destination generates packets every 0.02 seconds. However, for safety applications, the interval between packets can be less than 0.02 seconds. In addition, the position of the nodes was available through precise location service. Therefore, there is no error in the location information (an error model will be generated for further investigations). We also assumed UDP as the transport layer protocol for our study.

Moving nodes scenario (Fig. 5b) has the same parameters as the static nodes scenario, except that it has 50 nodes (yellow points) randomly distributed over the multi-lane roads and can move in all directions. The movements of the vehicles on the roads are based on the Car-following model (Krauss model), and the vehicles' speed is set up to not more than 20 m/s. In our simulation results, we have not considered the direction of movement and speed. Hence, we considered a fixed speed for all the vehicles. We are currently running additional simulations considering both the speed and direction, which will also be used as other parameters to evaluate the system performance. The communication range of vehicles is set to 250 meters. Lastly, the pair source-destination are randomly selected in this scenario.



(a) Static Nodes (b) Moving Nodes

Fig. 5: Simulation Scenarios.

The total time of each simulation run is configured to 300 seconds. All the results shown in the paper represent the average of 30 simulation runs and a 95 % confidence interval. The configuration of simulation parameters is summarized in Table I. These parameters are selected based on the previous studies as their simulated vehicular scenario [4], [5]. The parameters evaluated in our simulations are defined as follows:

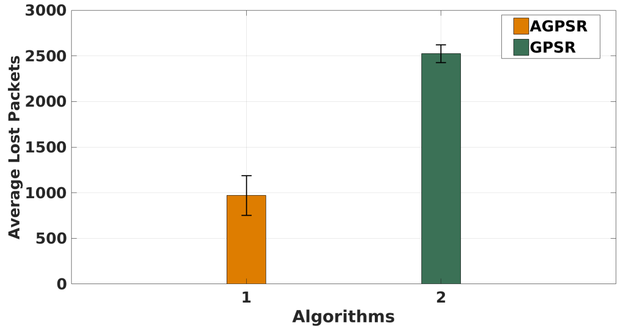
- **Packet Delivery Ratio (PDR):** The percentage of packets received by the destination for the total number of transmitted packets by the source.
- **Hop Count:** Average number of hops for all the packets received by the destination.
- **Lost Packets:** Difference between the number of packets transmitted and received.

TABLE I: Simulation Parameters.

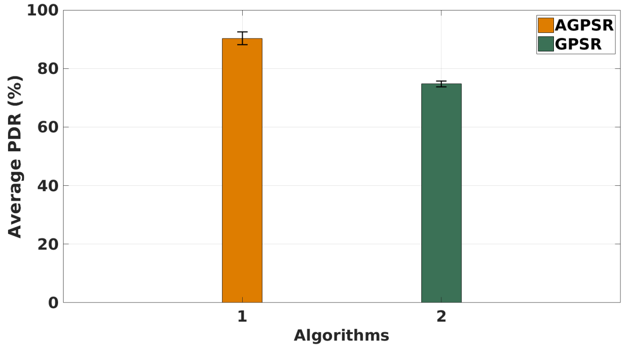
Parameter	Static Node Scenario	Moving Nodes Scenario
Simulator	NS-3/SUMO	NS-3/SUMO
Packet Size	200 bytes	200 bytes
Simulation Time	300s	300s
Simulation Area	1000x1000m	1000x1000m
Simulation Scenario	Manhattan grid	Manhattan grid
Pair Source-Destination	1 (Deterministic)	1 (Random)
Number of Nodes	100 (92)	50
Max. Speed	0	20 m/s
Data Type	CBR	CBR
Hello Interval	1s	1s
NT Entry Lifetime	2s	2s
Transport Protocol	UDP	UDP
Packet Interval	0.02s	0.02s
Mac Protocol	802.11p	802.11p
Transmission Range	100m	250m
Propagation Model	Two-ray ground	Two-ray ground
Routing Protocol	GPSR, AGPSR	GPSR, AGPSR

B. Scenarios Results

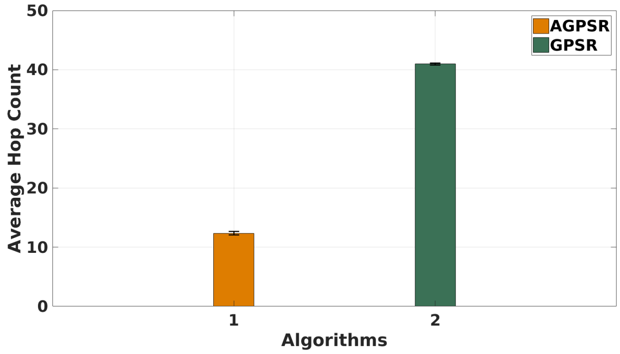
In Fig. 6, we present average results for static nodes scenario, with respective confidence intervals. The AGPSR outperforms the GPSR for all used metrics. The reductions of AGPSR concerning GPSR in relation to lost packet and hop count (Figs. 6a and 6c) were on average 60% and 70%, respectively. For packet delivery ratio, AGPSR had a gain of about 17% (Fig. 6b). The better performance of AGPSR for this scenario can be explained by its ability to avoid nodes that are sending packets in recovery mode. Thus, data packets are not sent through the path that is jammed, which will reduce the hop count and lost packets and consequently increases the packet delivery ratio.



(a) Average lost packets.

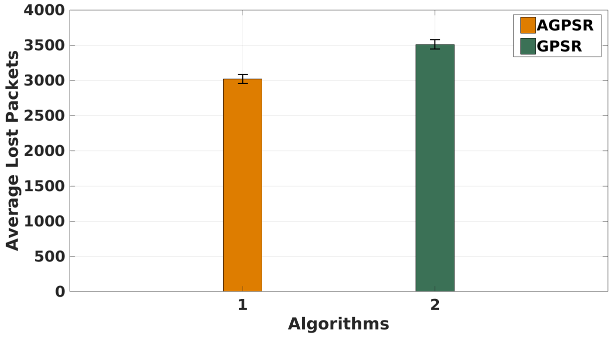


(b) Average packet delivery ratio.

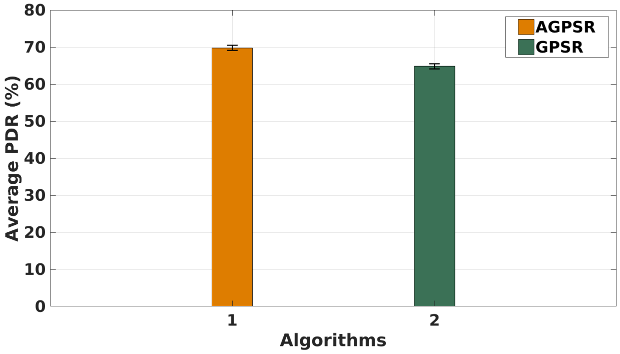


(c) Average hop count.

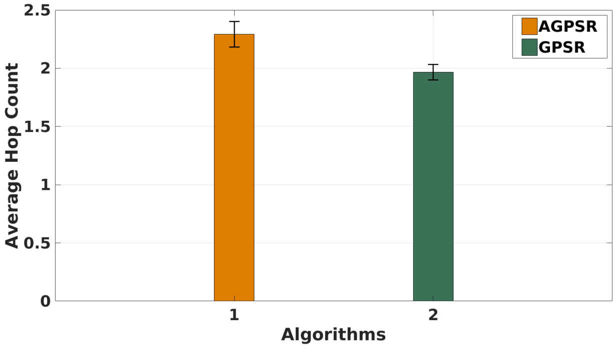
Fig. 6: Performance Comparison for Static Nodes Scenario.



(a) Average lost packets.



(b) Average packet delivery ratio.



(c) Average hop count.

Fig. 7: Performance Comparison for Mobility Nodes Scenario.

In Fig. 7, we present average results for mobility nodes scenario, with respective confidence intervals. The reductions of AGPSR comparing to GPSR for the lost packet (Fig. 7a) had an average of 15%. For packet delivery ratio, AGPSR had a gain of about of 8% (Fig. 7b). The smaller values of gains and reductions for this scenario is mainly because we do not have the exact control of the traffic jamming. For the mobility scenario, we use the NS-3 random mobility model. Thus, the chances of traffic jamming happening depend on the nodes random mobility. If no traffic jamming occurs, AGPSR has lower performance. Besides, in some situations, the nodes (source-destination) pair can be side-by-side, and transfer a large number of packets. This contributes to reducing differences between two algorithms. For this scenario, the AGPSR hop count is slightly higher than GPSR (Fig. 7c). This behavior can be explained because AGPSR receives more packets and only these received packets are used to calculate the hop count. Packets that performed a long travel on GPSR were dropped, while AGPSR was able to deliver those packets.

C. Runtime Comparison

All the simulations were run on a desktop with 4.2GHz x 8 Intel Core i7-7700K processor, equipped with 16GB of RAM and running Ubuntu 16.04 with kernel 4.4.0-116. The runtime comparison between both algorithms is shown in Fig. 8. It can be noted that the AGPSR performs better with respect to GPSR in both scenarios. The AGPSR algorithm obtains the best performance because it losses less packets than GPSR (as shown in Fig. 6a and Fig. 7a). Thus, less computational efforts are necessary.

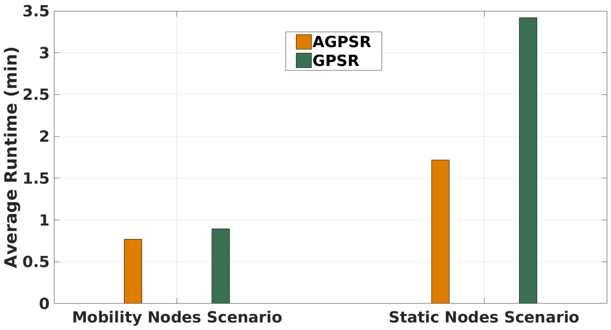


Fig. 8: Average runtime per simulation.

V. CONCLUSION

Designing a new routing protocol for VANETs is a challenging task owing to topology changes and high-speed mobility of vehicles, as well as the limited radio range of each vehicle in the network. In this paper, we described a modification for the well-known GPSR protocol, exploiting information about neighbors nodes during the selection of one-hop forwarding node. Our proposed algorithm adds a new field on NT avoiding nodes delivering packets in recovery mode. We successfully simulated our proposed algorithm in two scenarios (static and mobility nodes) in NS-3. Through extensive simulation results, we have demonstrated that the proposed protocol shows performance improvement over conventional GPSR protocol regarding packet delivery ratio, hop count and lost packets. As a continuation of this work, we intend to compare our proposed AGPSR with recent routing methods presented in the literature and improve our algorithm taking into account the speed, direction and nodes density. We are also obtaining additional results incorporating a probabilistic trust status filed instead of a deterministic one as well as adding more realistic scenarios to our investigation.

REFERENCES

[1] Falko Dressler, Hannes Hartenstein, Onur Altintas, and Ozan Tongus. Inter-vehicle communication: Quo vadis. *IEEE Communication Magazine*, 52(6):170–177, 2014.

[2] Kun Hua. Vehicular delay tolerant and sensor networks: Protocols and applications.

[3] Atul B Kathole and Yogadhar Pande. Survey of topology based reactive routing protocols in vanet. *International Journal of Scientific & Engineering Research*, 4(6), 2013.

[4] Degui Xiao, Lixiang Peng, Clement Ogugua Asogwa, and Lei Huang. An improved gpsr routing protocol. *Int. J. Adv. Comput. Technol*, 3(5):132–139, 2011.

[5] Lili Hu, Zhizhong Ding, and Huijing Shi. An improved gpsr routing strategy in vanet. In *Wireless Communications, Networking and Mobile Computing (WiCOM), 2012 8th International Conference on*, pages 1–4. IEEE, 2012.

[6] Brad Karp and Hsiang-Tsung Kung. Gpsr: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254. ACM, 2000.

[7] Christian Lochert, Martin Mauve, Holger Füßler, and Hannes Hartenstein. Geographic routing in city scenarios. *ACM SIGMOBILE mobile computing and communications review*, 9(1):69–72, 2005.

[8] Kevin C Lee, Jérôme Härri, Uichin Lee, and Mario Gerla. Enhanced perimeter routing for geographic forwarding protocols in urban vehicular scenarios. In *Globecom Workshops, 2007 IEEE*, pages 1–10. IEEE, 2007.

[9] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.

[10] The NS-3 Network Simulator. http://www.nsnam.org/.