IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 July 2023, accepted 21 July 2023, date of publication 27 July 2023, date of current version 3 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3299230

RESEARCH ARTICLE

Agent-Based Reactive Geographic Routing Protocol for Internet of Vehicles

MOHAMED MAZOUZI[1], KHALEEL MERSHAD[2], (Member, IEEE), OMAR CHEIKHROUHOU[3], AND MONIA HAMDI[4]

[1]CES Laboratory, National School of Engineers of Sfax, University of Sfax, Higher Institute of Business Administration of Sfax, Sfax 1013, Tunisia
[2]Computer Science and Mathematics Department, School of Arts and Sciences, Lebanese American University, Chouran Beirut 1102 2801, Lebanon
[3]CES Laboratory, National School of Engineers of Sfax, University of Sfax, Higher Institute of Computer Science of Mahdia, University of Monastir, Mahdia 5111, Tunisia
[4]Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia

Corresponding author: Monia Hamdi (mshamdi@pnu.edu.sa)

This research was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R125), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

ABSTRACT The design of an efficient routing algorithm for Internet of Vehicles (IoV) is a challenging research issue due to the inherent characteristics of IoV network, such as high-speed mobility of nodes, frequent topology change, link instability, and the presence of radio obstacles. Geographic routing protocols are a promising solution for the IoV, since they are based mainly on the location information which can be easily obtained through a location-based service. However, location services and traffic status measurements generate high network overhead that overloads the network traffic and introduces routing latency and packet loss. In this paper, we first propose a novel lightweight location service that permits to discover all the geographical paths between two vehicles based on smart mobile agents. Second, we proposed *ARGENT* an Agent-Based Reactive Geographic Routing Protocol that couples the routing process with the novel lightweight location service. We leverage the IoV environment and multiagent characteristics to adapt to the high-speed mobility of vehicles, improve packet delivery, and reduce the number of control messages overhead. The performance evaluation shows significant improvements in terms of query success ratio, data packet delivery, and end-to-end delay while reducing communication interruptions and overhead.

INDEX TERMS Internet of vehicles, routing protocol, location-based service, multiagent systems.

I. INTRODUCTION

Internet of Vehicles (IoV) is an essential component for the realization of applications of Intelligent Transport System (ITS) and other vehicular services. IoV aims to improve road safety, road planning, navigation, entertainment, relieve traffic congestion, optimize traffic flow, etc. [24], [47], [60]. Thus, IoVs are known as one of the most promising applications of mobile networks where the mobile nodes are represented by a moving vehicle characterized by high-speed mobility. As a result, new problems are introduced due to this feature such as consecutive link breaks, frequent disconnection, and quick and frequent topology changes. These issues complicate the design of an efficient routing protocol for the IoV and make the traditional routing protocols of ad-hoc networks inappropriate for it [59], Which requires the design of new protocols that are distinct and tailored for the IoV environment [64].

Basically, vehicular routing protocols are classified into two large categories: position-based and topology-based [12]. Topology-based routing protocols are based on information about the links to route the data packets. However, link instability in IoV environments causes a significant drop in the global performance of these protocols. Position-based routing protocols are created to overcome the problems caused by link instability. Indeed, data packet routing in this category is based only on the immediate neighbors' positions of the forwarding nodes, provided through Hello messages exchanged

The associate editor coordinating the review of this manuscript and approving it for publication was Chih-Min Yu.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

between neighbors, and the target node position. Hence, each sender node in this category requires, before sending any data packets, sufficiently precise knowledge about the geographical coordinates of the target node. Usually, these coordinates are made available through a location-based service which is intended to answer every request about nodes' positions. Consequently, position-based routing protocols require an efficient location-based service to ensure the smooth running of its process. However, the large-scale dynamic network and the high speed of vehicles cause several communication interruptions, increase the sending frequency of the location update/request messages, and add overhead in order to find/update nodes' positions. To resolve this issue, we optimized the network resources usage (wireless bandwidth) by removing the blind flooding mechanism used in classical location services. Moreover, although the geographic routing efficiency directly depends on the used location service efficiency [11], existing geographic routing protocols did not consider the effectiveness of the location services and traffic status measurement [9]. To resolve this problem, we integrate our location service directly into the routing process. Moreover, the proposed routing protocol constructs a geographical path according to road traffic density, location information validity, and number of zones in the path. This permits to reduce communication interruptions, and therefore, improve the query success ratio, and the latency.

More precisely, in this paper, we present a novel geographic routing protocol suitable for IoV environments called **ARGENT**: **A**gent-Based **R**eactive **GE**ographic RoutiNg ProTocol. The main contribution of this paper can be subdivided into two levels:

- Firstly, we design a novel lightweight reactive location service based on vehicles' zones rather than their positions, which increased the validity period of location information and reduced the overhead of control messages.
- Secondly, we design a novel geographic routing protocol (called ARGENT) based on information retrieved through our proposed location service. ARGENT is able to route data packets toward the destinations with low packet loss and reduced end-to-end delay.

The rest of this paper is organized as follows. We start in Section II by presenting some related works in location-based services and geographic routing protocols. In Section III, we briefly discuss the network environment considered for this work, then we detail our proposed routing protocol. Section IV presents the simulations that we conducted and their results that prove the interest in this work. Finally, Section V concludes the paper with a summary of our contributions and future work.

II. RELATED WORKS

As previously mentioned, position-based routing protocols are proposed to overcome the problems caused by the creation or maintenance of the links between vehicles, so they are considered the most suitable solution for large-scale dynamic networks [13], [15], [46], [48], [49]. All position-based routing protocols need the destination position to forward data packets. For this reason, we discuss in this section some work on the location-based service and the position-based routing algorithms [39].

A. OVERVIEW OF EXISTING LOCATION-BASED SERVICES

Location-based services are designed to answer all queries about the location of any node in the network. Usually, they are classified into two main classes: flooding-based (or push-based) and rendezvous-based (or pull-based) [30].

Flooding-based location services do not use any specific structure or organization (e.g. cluster, backbone…). This class can be classified into proactive and reactive services. Proactive flooding-based location-based services allow each node to periodically flood the network by its geographical information so that other nodes will be able to retrieve information about its location. However, the flooding mechanism overloads the network with high overhead traffic with each sending of location update messages. In order to reduce this problem, several techniques have been proposed. The most used techniques are: the temporal resolution, in which location update sending frequency is adjusted with the node's mobility rate (i.e. the higher the speed is, the more frequent the location update messages are sent and vice versa) [50]. The second technique is often known as spatial resolution [45], the main idea of this technique is that the maintained node position accuracy varies with the distance variation that separates them (i.e. a smaller distance between the two nodes means that each one of them must maintain precise location information of the other). This technique is also known as the distance effect [9]. The most known example of proactive flooding-based location-based service is Distance Routing Effect Algorithm for Mobility (DREAM) [10], [57]. Reactive Flooding-based location-based services are created to avoid the unnecessary overhead introduced by the location update messages. So, each node sends its location information only when it receives a location request message. However, like any reactive method, this method causes several communication interruptions and so it increases the network latency. Reactive Location Service RLS [38], [51] is the most-known example of this class of location service.

Rendezvous-based location-based services are designed to avoid the blind flooding mechanism used in the first class. The main idea of this method is that each node elects one or more other nodes to be its location servers. The elected location servers of a node are known by all other nodes since they agree on a unique mapping for the location server election process. The operation mechanism of this class is that each node must periodically update its location information maintained in its location servers using the location update messages. Nodes wishing to retrieve the location information of a certain node N send a request to one of the location servers elected by N (generally the nearest location server). The location servers are responsible for answering the requests about the position of the node that elected them. Generally, rendezvous-based location-based services are classified into two families. The first family includes systems based on a centralized indexing mechanism such as the quorum approach [27], clustering [5], or using roadside units (RSUs) [52]. The second family includes systems based on a decentralized indexing mechanism such as Grid Location Service (GLS) [41] and Hierarchical Location Service (HLS) [1], [62]. For example, the Mobile Group-based Location Service (MoGLS) [62] defines a two-level hierarchy: a lower level that contains groups of vehicles that are on the same trajectory, and a higher level in which the map is divided into fixed geographic regions. An RSU in each region assumes the role of the Region Head (RH), while a vehicle in each group is selected as the Group Leader (GL). Each vehicle periodically uploads its location to the GL, which in turn sends the locations to the RH. The GLs collaborate to answer location queries within the region, while RHs work together to resolve location queries across regions. A Comparison of RLS, HLS, and GLS is carried out in [7]. The authors in [31] proposed a system (PETAL) in which the concept of collective abstract information is applied such that all nodes who have any information about the destination location collaborate to identify its current location.

B. OVERVIEW OF EXISTING GEOGRAPHIC ROUTING PROTOCOLS

Location-Aided Routing (LAR) [40] is the first routing algorithm that uses geographical information when routing data packets. This is done to reduce the number of route discovery messages flooded in the network. Indeed, LAR uses information about the target node position to create a first area called Expected Zone (EZ), known as a small circle where the target node can be located, and a second area called Request Zone (RZ) which is known as the rectangle in which the EZ and the source node are located. Once a sender node needs to build a route for reaching another destination node, only the nodes belonging to the RZ area are allowed to forward the route discovery messages. Greedy Perimeter Stateless Routing (GPSR) [37] is a reactive routing protocol based on a greedy forwarding approach where the data packet forwarding is always through the neighbor node that is geographically closer to the target node. A recovery mode ''perimeter routing'' is used whenever a data packet gets stuck in a local optimum. This protocol gives an interesting performance in MANET environments. However, it does not consider the urban environment characteristics and the radio obstacles' existence that can adversely affect its global preference [39]. A combination of GLS and GPSR (labeled HRGLS), and HLS and GPSR (named HRHLS) is presented in [6]. In [58], Silva et al. analyze the shortcomings of GPSR and propose a new protocol 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. It can also eliminate packet routing loops to avoid delivering the packet to the same neighbor and help to overcome link breakages due to unavoidable reasons, such as road accidents or dead-end roads.

Geographic Source Routing (GSR) [43] has been designed as a routing protocol suitable for city settings since it merges the urban topology with position-based routing. After obtaining the target node position, the sender node specifies a sequence of intersections on its digital maps (known as GSR anchors) through which the data packets should pass. The sender applies the Dijkstra algorithm to find the shortest path toward the destination. In addition, data packet forwarding between two successive intersections is performed through the greedy forwarding method. The major disadvantage of GSR is that it does not consider the case of a sparse network, where the number of vehicles in a road segment is insufficient to forward the data packets.

Like GSR, Anchor-based Street Traffic Aware Routing (A-STAR) [56] uses the Dijkstra algorithm to route the data packets from a sender node towards another destination node. In addition, road traffic density is considered during the use of the Dijkstra algorithm since all roads are weighted by the number of bus lines crossing them. In the A-STAR paper, the authors did not specify the location service used in this protocol. However, traditional location services are intended to provide the position of a node in the network and cannot provide information about the traffic conditions.

Improved Greedy Traffic Aware Routing (GyTAR) [35] requires the use of fixed infrastructures (APs) in junctions to collect information about the traffic density of each road segment and to forward the data packets from one segment to another. When the data packets arrive at an AP, this latter must select the next junction following the curve metric distance (between the target node and the next junction candidate) and the road segment traffic density. GyTAR presents two major problems: First, the use of APs introduces bottlenecks since all nearby communications will pass through them [21]. Second, the next junction selection process is done without considering the vehicle's direction. Hence, the risk of packet loss will be increased when the number of vehicles that move to the destination is insufficient to forward the data packets.

Similar to GyTAR, the system in [32] is based on the concept that at the onset of a new connection, all vehicles in a certain route report their locations using GPS to a Location Verification System (LVS) which is installed at a base station. The LVS verifies the reported locations and broadcasts the decisions to all vehicles in the transmission range. Via local updating of all routing tables, all vehicles are notified of the decisions. The authors investigate this method in the context of a well-known position-based routing protocol, namely, the Hybrid Location-based Routing (HLAR) protocol [4] and they proved that advanced location verification techniques can be seamlessly integrated into the hybrid routing protocols embedded in vehicular networks.

A location-based opportunistic geographic routing (LOGR) protocol is proposed in [42]. Here, the sending node does not determine the next-hop node. Instead, it broadcasts the forwarding rules and the data packet together, utilizing the broadcast nature of wireless transmission. Then, each node receiving the packet will judge whether it has the right to forward the packet according to the preset rules. This protocol is based on the priority assignment of opportunistic forwarding and the adopted absolute priority method. Any candidate receiving the packet can derive its own priority from its location information and forwarding rules, and determine the forwarding time, without knowing the information of other candidates. In this way, the LOGR protocol eliminates the need for topology detection or location update, and avoids the network overhead incurred in pairwise comparison of nodes for priority assignment.

In [36], the authors present a Predictive Geographic Routing Protocol (PGRP) that improves path connectivity. In PGRP, every vehicle gives a weight to each neighbor according to the direction and the angle of the vehicle's movement. PGRP can predict the location of a vehicle based on the last hello packet parameters such as the vehicle's acceleration. The protocol forwards packets according to the predicted location of the destination vehicle after a short interval.

Lu et al. [44] present a position-based routing scheme called improved geographic routing (IGR) for inter-vehicle communications in city environments. IGR uses vehicular fog computing to make the best utilization of vehicular communication and computational resources. The protocol selects the routing path according to the link error rate and vehicle density of the streets. It uses the street map and vehicle density to determine the next junction that the transmitted packet should follow. In [54], The authors focus on an energy consumption location-based QoS routing (ELQR) protocol to establish routes from a source to a destination assisted by routers to save energy and overcome the deadlock problem.

In [16], a distributed routing protocol DGGR was proposed, which takes into account sparse and dense environments to make routing decisions. Each Road segment is assigned a Weight (RWE) based on two delay models via exploiting the real-time link property when connected or historic traffic information when disconnected to determine the routing path that can greatly alleviate the risk of local maximum and data congestion. Chen et al. divided the road map into a series of Grid Zones (GZs). They classify the packet transmission into intra-grid and inter-gird zone transmission to deal with the big network scale by dividing the road map into a series of GZs. Based on the position of the destination, the packets can be forwarded among different GZs instead of the whole city map to reduce the computation complexity, where the best path with the lowest delay within each GZ is determined.

The Beaconless Traffic-Aware Geographical Routing Protocol (BTA-GRP) is proposed in [19] for both dense and sparse traffic conditions. It addresses delay, disconnection, and packet dropping issues by considering traffic density, distance, and direction for the next forwarder node and route selection. BTA-GRP utilizes RTS/CTS control packets for the nodes which are located at or between intersections. When the node reaches the next intersection, it determines the traffic density and selects the road with maximum vehicle nodes towards the destination. Also, the protocol selects the direction that avoids looping problems due to bi-directional vehicle movement. The RSUs are used at intersections to update the traffic conditions based on the map segmentation method, where they detect the passing-by vehicles and broadcast this information to the intersection area vehicle nodes.

To deal with high mobility and frequent link disconnections in VANETs, an artificial Spider Geographic Routing in urban VANETs (ASGR) [17] was proposed to establish a reliable route for delivering packets. First, from the point of bionic view, Chen et al. construct the spider web based on the network topology to initially select the feasible paths to the destination using artificial spiders. Next, the connection-quality model and transmission-latency model are established to generate the routing selection metric to choose the best route from all the feasible paths. In [29], the authors propose a multi-metric technique for next hop selection. It selects next hop vehicles from dynamic forwarding regions, and considers major parameters of urban environments including received signal strength, future position of vehicles, and critical area vehicles at the border of the transmission range, in addition to the speed, distance, and direction. The authors in [2] and [3] proposed RTISAR a real-time intersection-based segment-aware routing for VANETs. This protocol chooses the next intersection by considering the traffic segment status. RTISAR presents a new formula for assessing segment status based on connectivity, density, load segment, and cumulative distance toward the destination.

The All-Round and Highly Privacy-Preserving Location-Based Routing for VANETs (ARPLR) was proposed by Wang et al. [61] to address an issue in the existing location-based routing schemes which ignore the location privacy protection of vehicles, leading to malicious users tracking the drivers' movements. ARPLR first proposes an RSU-assisted location management with location privacy protection that prevents the destination vehicle's location from being leaked by the arbitrary query. Location management is the basis of location-based routing and it includes two processes: location update and location query. A message routing based on location ciphertext with high privacy protection is designed by order revealing encryption, in which a multi-hop routing between the source and destination vehicle is established only by comparing the encrypted locations between intermediate vehicles. In [28], an onion-based routing protocol was proposed which uses the concept of location-based dynamic relay groups. In this concept, vehicles dynamically form groups around specific locations to act as cryptographic onion relays. The proposed protocol satisfies source, destination, and route anonymity. The Employment of pseudo-IDs further helps in keeping the real identity of vehicles anonymous.

A hierarchical hybrid routing protocol called Dynamic Real-time Multimodal Routing (DREAMR) was proposed in [18]. It uses an Online Stochastic Shortest Path problem to execute two processes: an open-loop layer that determines optimal paths on the network using graph exploration techniques, making decisions regarding the appropriate modes of transportation and their durations. The output is a series of sub-routes, each belonging to a specific mode. Next, a closed-loop layer carries out real-time agent control actions using policies for the sub-routes that consider the possible uncertainties. DREAMR utilizes a time-dependent directed acyclic graph (DAG) model with nonstationary edge weights for route planning. The protocol achieves a good success rate. However, the routing latency was not tested by the authors.

C. DISCUSSIONS

Usually, the location-based service and routing protocol are separate programs. However, the two main processes performed by these two algorithms are very attached. Firstly, the location service is triggered to find the target node position, then a position-based routing protocol is used for routing data packets towards the source node (according to the location information retrieved through the used location service). This separation hides the real performance of the used position-based routing protocol since all the overhead introduced by the location service will not be considered. In addition, the distributed nature of this latter does not allow it to retrieve information about traffic conditions. Hence, position-based routing protocols need some additional control messages (considered as overhead) to retrieve this information.

The reactive class of location service is characterized by a large number of control messages generated when locating a certain node in the network. This feature is mainly due to the blind flooding mechanism used by this class when disseminating the location request messages. In addition, this feature is emphasized when the sender nodes start long communication sessions with the target nodes since they must send new location requests whenever the validity period (freshness) of the old locations are expired. Moreover, the location request sending frequency will be increased in the IoV since the high speed of nodes significantly reduces the validity period of the maintained location information. This leads to several communication interruptions and to large signaling overhead to retrieve the new destination node location. All these problems make this class of location services inappropriate for the IoV environment [7].

III. ARGENT: AGENT-BASED REACTIVE GEOGRAPHIC ROUTING PROTOCOL

A. NETWORK ENVIRONMENT

We consider an IoV that is composed of a set of heterogeneous vehicles in movement randomly placed on roads according to the streets' rules. We assume that only V2V communications are used and all vehicles are equipped with an ad-hoc wireless communication device, GPS receivers (Geographical Positioning System), and digital maps. To be close as possible to the real-world environment, we assume again that the vehicles have different speed variances and different transmission ranges. Thus, it is assumed that road width is lower than the transmission range of vehicles. In this proposed work, the space containing the network is subdivided into several road segments. Each segment is related to other neighboring segments through an intersection zone. Vehicles located in intersection zones are assumed to be located in the same previous segment until they leave this zone area. A road segment is defined here as a geographical zone placed between two successive intersections. In addition, each segment is known by a unique identifier and is delimited by two intersections. Each vehicle can retrieve the coordinates of any road segment (segment identifier, segment length, and the two intersections' identifiers that delimit it) through its digital maps.

B. ARGENT OVERVIEW

ARGENT aims to eliminate the previously mentioned problems of geographic routing protocols. To reach this goal, we took advantage of the road topology to fight against problems caused by the radio obstacles placed on the road edges and to increase the validity period of location information provided to vehicles. This is done to increase the routing adaptability with the city settings and also to reduce the sending frequency of location request messages caused by communication interruptions. We have also benefited from a special feature of the IoV, in which the road width is smaller than the transmission range of vehicles [63], to avoid the blind flooding mechanism when disseminating location request messages.

To achieve this goal, our proposed ARGENT uses a Path Discovery Mobile Agent (PDMA) triggered when a sender vehicle should start a communication session with another destination vehicle, and it does not have valid (fresh) information about the location of this latter. Unlike traditional reactive location-based services, the main purpose of PDMA is to find the segment identifier in which the target destination is located and not the geographical coordinates of this latter. Then, the freshness of location information provided to the sender vehicle becomes proportional to the stay period of the target vehicle in this segment and not to a small approximate period fixed by the protocol constructor (see [25]). In order to reduce the sparse vehicular traffic effects, PDMA is used also to collect information about vehicle density in the road segments, so that the target destination can be able to retrieve the optimum geographical paths that link it with the sender vehicle. Then, a Path Creation Mobile Agent (PCMA) is triggered by the destination vehicle to provide to the sender vehicle the required routing information (such as the segment coordinates in which the target destination is located, the optimum geographical path for reaching this latter, and the validity period of this routing information). Hence, the sender node can send its data packets toward the destination node according to the routing information retrieved from the received PCMA(s). A geographical path is defined here as an intersections sequence that links both segments in which the sender vehicle and the destination vehicle are located. The geographical path is optimum if it minimizes the distance between the sender vehicle and the destination vehicle and maximizes traffic density through this path. When the source vehicle receives the PCMA agent packet, it utilizes the routing information that was collected by the PCMA agent to create a Data Forwarding Mobile Agent (DFMA) that will carry the source vehicle's data and route it to the destination vehicle according to the optimum geographical path that was determined by the PCMA. Figure 1 illustrates the order of the three agents and the source and destination of each agent. In the next section, we describe the complete details of each agent.

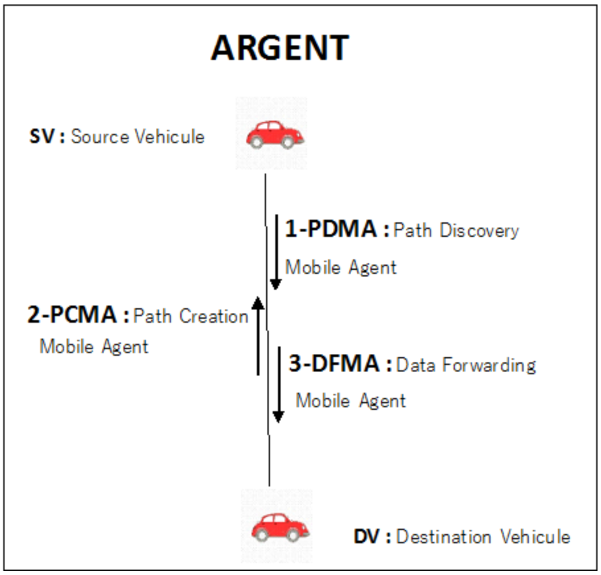

FIGURE 1. ARGENT protocol overview.

C. ARGENT AGENTS

Recently, agent technology is considered the best alternative to the traditional programming paradigm since it adds features such as flexibility, scalability, learning, and cooperation to software development. Moreover, they facilitate all the requirements of Component-Based Software Engineering (CBSE) [14], [26], [33] and sophisticated software development [23], [55].

An agent is defined as an autonomous program that can perceive its environment and act upon this latter using its knowledge base to achieve its own goal [20], [22], [34]. Multi-Agent System (MAS) is an environment (application) comprised of a set of agents that cooperate to perform a common task or a set of tasks. MAS implements several complex interactions such as cooperation, negotiation, and coordination [20].

A mobile agent is a roaming agent able to move an entire process from one host to another host where this process is split into several instances that are executed on different hosts. The software mobile agent paradigm is considered as a special case of remote evaluation paradigm that allows the sending hosts to move their computations (in the form of programs) towards the destination hosts that they maintained the required data and so locally executes this program and returns results to the sending hosts. This flexibility can significantly reduce the network load since the required number of messages exchanged between hosts to accomplish such a task will be decreased, which is important in ad-hoc wireless communication modes where resources may be limited (e.g. bandwidth). Moreover, mobile agents help to reduce network latency since they can directly operate on the mobile hosts where the resources are located.

In our work, we defined a geographic routing agency that consists of a knowledge base (KB) and two types of agents (stationary and mobile). It is responsible for performing the entire routing process proposed in ARGENT. The agency components and interactions can be described as follows:

1) *Vehicle Manager Static Agent (VMSA)*: It is a stationary agent that resides in each vehicle. The functions of VMSA are as follows: (1) VMSA creates Path Discovery Mobile Agent (PDMA), Path Creation Mobile Agent (PCMA), Data packets Forwarding Mobile Agent (DFMA), and knowledge base (KB). (2) VMSA uses information provided through its digital maps and GPS to compute vehicle velocity and Minimum Stay Duration (MSD), which is the remaining time before the vehicle leaves its current road segment.
2) *Path Discovery Mobile Agent (PDMA)*: Is a mobile agent triggered by VMSA of a sender vehicle in order to find the target destination and provides it with the weight factor of the traveled geographical path.
3) *Path Creation Mobile Agent (PCMA)*: Is a mobile agent triggered by VMSA of the destination vehicle in order to respond to a request received through PDMA. PCMA is triggered when the received PDMA has the most weighted geographical path than those who preceded it.
4) *Data packets Forwarding Mobile Agent (DFMA)*: Is a mobile agent triggered by the source vehicle's VMSA in order to route data packets toward the destination vehicle.
5) *Knowledge base (KB)*: Is a software object deployed at each vehicle. It maintains information about itself such as vehicle identifier, position, velocity, MSD and current moving segment information (current segment identifier, identifiers of the two intersections delimiting the current segment and the last traversed intersection identifier). KB maintain also information about each neighboring vehicle (vehicle identifier, velocity,

TABLE 1. Notations used in the paper.

Notation	Definition
SV	Source Vehicle
DV	Destination Vehicle
CV	Current vehicle identifier
EV	The closest vehicle to one of the two intersections that delimit this segment
NFV	Next-Forwarding-Vehicle
TR	The trace of another PDMA that has the same DISCOVERY-ID as the current PDMA
TR_{wf}	The weight factor of PDMA where TR represents its trace
CSI	The segment identifier in which CV is located
NL[]	Neighbors-list
KB	Knowledge Base
SZ	Segment zone located between CV and PDMA's NFV
MSD	Minimum Stay Duration
VMSA	Vehicle Manager Static Agent
PDMA	Path Discovery Mobile Agent
PCMA	Path Creation Mobile Agent
DFMA	Data Forwarding Mobile Agent
PDMA_wf	PDMA weight factor measurement
PDMA_ds	density zones of the segments traveled by PDMA
PDMA_nzones	number of zones traversed by PDMA
PDMA_nhops	number of hops performed by PDMA

segment identifier, MSD, and position) and information provided through the received PDMA and PCMA. KB can be read and updated by all agents.

The proposed routing protocol is combined with a new reactive location-based service used on demand to locate a destination vehicle and links it with the sender vehicle through a geographical path that must be optimum in terms of distance and traffic density. If a Source Vehicle (SV) needs to send data packets to another Destination Vehicle (DV), the VMSA of SV checks firstly if it has fresh routing information in its KB about DV or not. If yes, SV immediately starts sending data to DV. If not, the VMSA of SV triggers a PDMA with a new DISCOVERY-ID to retrieve the required routing information for reaching DV. During its migrations, PDMA uses a next-hop selection method inspired by the greedy forwarding mechanism. However, the high-speed variance of vehicles reduces significantly the performance of this method since the maintained neighbor-nodes positions will be changed after receiving their last HELLO messages [21]. In order to solve this problem and to improve the protocol's adaptability with the high speed of vehicles, PDMA updates the neighbor list of each visited forwarding vehicle before migrating. In this proposed work, PDMA is the main contributor to abandoning the blind flooding mechanism. To reach this purpose, we make it able to select, at each visited forwarding vehicle, its Next-Forwarding-Vehicle (NFV) from which it can perform its next migration.

In what follows we describe the PDMA journey for finding the targeted destination. Then, the PCMA journey and the DFMA journey. The notations that will be used are summarized in Table 1.

Algorithm 1 PDMA Code

INPUT:
- Data collected by PDMA (explained in (1)),
- KB of CV

OUTPUT:
- List of NFVs

```
Struct PDMAdata                    ▷ Data used by PDMA
    PDMA_wf                        ▷ PDMA's weight factor
    PDMA_ds                        ▷ Traversed-path traffic density
    NFV                            ▷ Next Forwarding Vehicle
    TIL:ARRAY                      ▷ Traversed Intersection List
    DV                             ▷ Destination Vehicle identifier
    SV                             ▷ Source vehicle identifier
    NIC                            ▷ Next Intersection Candidate
    LSI                            ▷ Last visited Segment Identifier
    SZ_ds                          ▷ Segment Zone Traffic Density
EndStruct
compute PDMA_wf
if EXISTS TR IN PDMA_wf > TR_wf then
    PDMA adds its trace to CV's KB
    if CV ≠ NFV then
        PDMA expires
    else
        if CV = DV then
            CALL AddIntersection(NIC,TIL)
            VMSA generates a new PCMA
        else       ▷ PDMA must continue its migrations
            CALL UpdateNeighborList(NL)
            if CV is not an EV then        ▷ Internal migration
                if CSI ≠ LSI then
                    NIC ← SelectNIC(NIC,CSI)
                end if
                if DV IN NL then
                    NFV ← DV
                else
                    NFV ← MinDist(n: {n ∈ NL}, NIC)
                end if
                SZ_ds ← GetDensity(CV, NFV, NL)
                PDMA_ds ← PDMA_ds + SZ_ds
                INCREMENT PDMA_nhops
                INCREMENT PDMA_nzones
                LSI ← CSI
                PDMA migrates to all neighbors in SZ_ds
            else                           ▷ External migration
                if CV is not an EV and CSI = LSI then
                    CALL AddIntersection(NIC, TIL)
                    CALL SelectNFV(CSI, NL,NIC)
                    INCREMENT PDMA_nhops
                    LSI ← CSI
                    PDMA migrates to all selected NFVs
                end if
            end if
        end if
    end if
end if
```

D. PATH DISCOVERY MOBILE AGENT JOURNEY

Algorithm 1 shows the tasks performed by a PDMA after it is triggered. A PDMA interacts with the VMSA of each visited vehicle (*CV*) to perform the following tasks: Firstly, it computes its weight factor ($PDMA_{wf}$). $PDMA_{wf}$ is used here to measure the effectiveness of the geographical path traveled by the PDMA and it is obtained through (1).

$$PDMA_{wf} = \left(\frac{PDMA_{ds}}{PDMA_{nzones}}\right) \times \left(\frac{1}{PDMA_{nhops}}\right) \quad (1)$$

where $PDMA_{ds}$ is the global density of zones in segments traveled by PDMA from its triggering until reaching CV, $PDMA_{nzones}$ is the number of zones traversed by PDMA and $PDMA_{nhops}$ is the number of hops on the path traveled by PDMA.

After that, PDMA checks if CV has a trace TR where TR_{wf} is greater than $PDMA_{wf}$ or not (i.e. CV is already visited by another PDMA that has the same DISCOVERY-ID and has a more efficient geographic path than the current PDMA). If yes, PDMA immediately expires from CV. If not, it adds its trace (DISCOVERY-ID and $PDMA_{wf}$) in the KB of CV and checks again if CV is the NFV selected at the last visited vehicle or not. If not, PDMA does not migrate from CV. Rather, it immediately expires. If yes (i.e., CV is the selected NFV), PDMA checks again if CV is the destination vehicle (DV) or not. If yes, PDMA adds its NIC to its TIL (this is done to specify the segment coordinates in which DV is located) and interacts with VMSA of CV to trigger a new PCMA (Algorithm 2). If CV is not DV, PDMA must continue its migrations from CV. Before migrating, it updates CV's neighbor list by predicting the movements made by each neighbor after receiving its last HELLO message and removing the neighbors located outside CV's transmission range. PDMA predicts the current position of CV's immediate neighbors according to (2).

$$V_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}; V_i' = \begin{bmatrix} x_i + v_{x,i}\delta t \\ y_i + v_{y,i}\delta t \end{bmatrix} \quad (2)$$

where $\delta t = (t_c - t_{ri})$, t_c is the current time, t_{ri} is the time on which CV received the last HELLO message from the neighbor vehicle i. V_i is a vector of vehicle i's previous position (at time t_{ri}) and V_i' is a vector of vehicle i's predicted current position (at time t_c). $v_{x,i}$ and $v_{y,i}$ represent the velocity of vehicle i respectively in the x and y directions. After that, PDMA distinctly computes the curve metric distance between CV and all its neighbors and removes all the distant neighbors (separated by a distance greater than CV's transmission range) from CV's neighbor list. In ARGENT, two types of PDMA migrations are defined: internal migration and external migration.

1) INTERNAL MIGRATION

This migration type is performed by the PDMA between vehicles located in the same segment. The main purpose of this migration type is to seek DV in this segment and to collect information about its traffic density. With each use of this migration type, PDMA must select a single vehicle to be its NFV. Moreover, the selected NFV must be the closest neighbor (among neighbors located in the same segment as CV) to PDMA's next intersection candidate (NIC). With the special feature of IoV in which the road width is smaller than the vehicle transmission range, it can be shown that all the vehicles located in the Segment Zone (SZ) are among CV's neighbors-list (SZ is the zone located between CV and the selected NFV where its length is proportional to the distance between CV and NFV and its width is proportional to the segment width). Hence, DV seeking process and traffic density measurement in this segment zone will be made locally in CV. This is done to facilitate DV seeking process and traffic density measurement in all these road segments, since the latter will be subdivided into a set of zones from which PDMA perceives these segment components. The neighbors' vehicles considered as not NFV must be notified by PDMA to avoid routing loops caused by other PDMAs that have the same DISCOVERY-ID. PDMA performs an internal migration in two cases:

1) *If CV was not an Extremity Vehicle (EV)*: In this case, PDMA checks if DV is among CV's neighbors list or not. If so, it chooses DV as its NFV. If not, it chooses the closest neighbor to its NIC as its NFV. After selecting its NFV, PDMA computes SZ traffic density and adds it to the total traffic density of the traversed zones ($PDMA_{ds}$). SZ traffic density (SZ_{ds}) is computed through (3).

$$SZ_{ds} = \left(\frac{n}{D_{CV,NFV}}\right) \times \left(\frac{\sum_{i=1}^{n} MSD_i}{n}\right) \quad (3)$$

where n represents the number of neighboring vehicles located in the zone SZ, $D_{CV,NFV}$ is the curve-metric distance between CV and the selected NFV (i.e. SZ Length), MSD_i is the minimum stay duration of the neighbor vehicle i. After that, PDMA creates clones and selectively floods them through the neighbors located in SZ.

Agent cloning technique is used to duplicate an agent, where the cloned agent is like the original agent (parent agent) since it contains the code and data of this latter [53]. In this work, the cloned agent has only one parent residing at each visited vehicle, so that the cloned agent can communicate to the VMSA of each visited vehicle for the different cloning levels.

2) *If CV is in a segment other than that of the last visited vehicle* (i.e., $CSI \neq LSI$): In this case, PDMA selects its NIC and performs the same tasks described in the first case.

2) EXTERNAL MIGRATION

PDMA uses this migration type when it reaches an EV (i.e. PDMA reaches the closest vehicle to its NIC. Hence, DV is not located in the current segment). External migration is done to move PDMA from one segment to the other neighboring segments. If CV is an EV and CV is located in the same

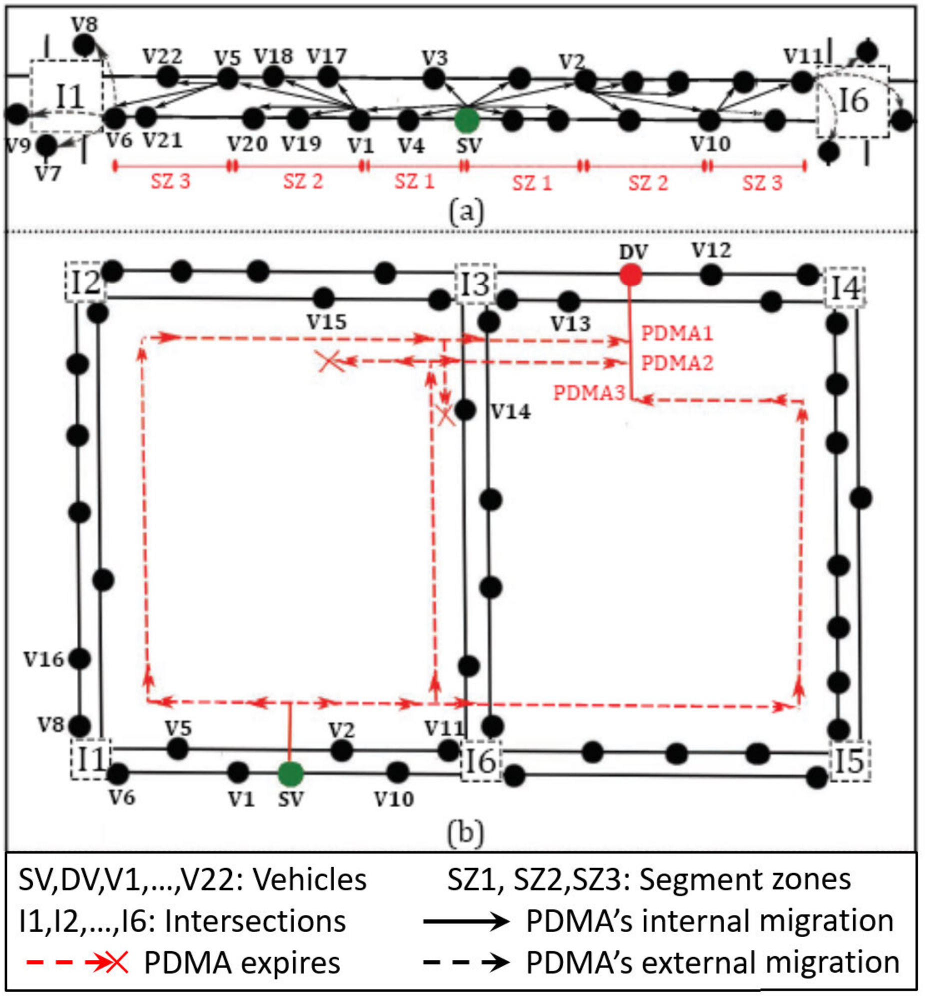

FIGURE 2. Example of PDMA journey.

segment as the last visited vehicle (i.e., $CSI = LSI$), PDMA adds its NIC to its TIL and chooses, among CV's neighbors, all the EVs located in the neighbors' segments as its NFVs. After that, PDMA creates clones and selectively sends them to all the selected NFVs.

Figure 2 shows an example scenario of the path discovery process performed by PDMA. In this example, we assume that the source vehicle (SV) needs to send data packets to the destination vehicle (DV) and does not have fresh routing information to reach it. Initially, the SV's VMSA creates PDMA with a new DISCOVERY-ID and provides it with the address of the destination vehicle (DV) and SV's segment identifier (S1). Thus, VMSA initializes $PDMA_{ds}$, $PDMA_{nhops}$, $PDMA_{nzones}$, and PDMA's traversed intersections list (TIL) with zeros. Then, VMSA triggers PDMA towards the two intersections that delimit the current segment (I1 and I6). To reach I1 (NI = I1), PDMA performs the tasks presented in Algorithm 1. Firstly, it updates SV's neighbor list and chooses V1 as its NFV since V1 is the closest neighbor to I1 (internal migration). After that, PDMA computes SZ traffic density (SZ_{ds}) and adds it to $PDMA_{ds}$, increments $PDMA_{nhops}$ and $PDMA_{nzones}$, creates clones and selectively floods them to all the neighbors located in SZ (V3, V4, and V1). When PDMA reaches V3 and V4, it finds that this vehicle was not the NFV chosen at the last visited vehicle (SV) and does not have a trace *TR*. So, it adds its trace in the KB of this vehicle and expires. From V1, PDMA chooses V5 as its NFV and repeats the same tasks performed at SV (internal migration) since V5 is the selected NFV and was not an EV (has closest neighbors to I1 than itself). At the next NFV, which is V6, PDMA performs an external migration from V6 since it is an EV. So, it adds I1 to its TIL and chooses, from V6's neighbor list, all the EVs located in the neighboring segments as its NFVs. Then, PDMA creates clones and selectively floods them to the selected NFVs. At V8, PDMA finds that the last visited vehicle (V6) is located in a different segment than that of V8 (i.e., $CSI \neq LSI$). Hence, it chooses I2 as its NIC (the current segment's intersection that PDMA had not traversed), selects V16 as its NFV and performs an internal migration. PDMA continues its migrations in the same way until reaching V13 (DV's neighbor). Hence, it chooses DV as its NFV and performs an internal migration. When PDMA reaches DV, it adds its NIC (I4) to its TIL (I4 is added here to identify the segment in which DV is located and so specify DFMA's future direction when it reaches I3) and adds its collected information to the KB of DV such as its TIL (I1, I2, I3, I4), its weight factor $PDMA_{wf}$ and the segment coordinates in which SV is located (S1).

Other PDMA clones reach DV at different moments. In this figure, we assume that PDMA with TIL equal to (I1, I2, I3, I4) is the first arrival at DV (PDMA1). PDMA with TIL equal to (I6, I3, I4) is the second arrival at DV and it has the greater $PDMA_{wf}$ (PDMA2). PDMA with TIL equal to (I6, I5, I4, I3) is the third arrival at DV and it has the lowest $PDMA_{wf}$ (PDMA3).

When PDMA reaches a vehicle that is visited by another PDMA with a greater impact factor, it immediately expires. For example, when PDMA with TIL equal to (I6) and NIC equal to (I3) reaches V14 (its selected NFV), it finds that V14 maintained a trace TR of another PDMA (PDMA with TIL equal to (I1, I2, I3) and NIC equal to (I6)) where TR_{wf} is greater than its weight factor. So, it immediately expires.

E. PATH CREATION MOBILE AGENT JOURNEY

PCMA is triggered by DV when it receives the first PDMA. Before triggering PCMA, the VMSA of DV must specify the geographical path intersections list (IL) that future DFMAs should follow to reach DV. The IL provided to PCMA is the inverse of the TIL retrieved from PDMA. In addition, VSMA adds to IL's end the intersection-identifier which was not mentioned in PDMA's TIL and which is one of the intersections that delimit the segment in which SV is located.

For example, in Figure 2, the PCMA that is triggered as a response for PDMA1 is provided with this IL (I4, I3, I2, I1, I6) since SV is located in S1 (delimited by I6 and I1) and PDMA1's TIL is (I1, I2, I3, I4). This is done to specify the intersections between which SV is located. PCMA starts its migrations with the second element of its IL (NIC=I3 in this example) since the first element is added to specify the segment coordinates in which DV is located.

Algorithm 2 shows the tasks performed by PCMA after it is triggered. Similar to PDMA, it interacts with the VMSA of each visited vehicle (CV) to perform the following tasks: Firstly, it checks if CV is the source vehicle (SV) or not. If yes, it provides CV with the required routing information for reaching DV such as the geographical path SV-DV (PCMA's IL) and the MSD of DV in its segment (which indicates the validity period of this routing information). If not, PCMA must continue its migrations until reaching SV. So, it updates CV's neighbor list (in the same way as PDMA), then it selects its next hop and migrates to it. The next-hop selection process is done in this way: PCMA firstly checks if SV is among CV's neighbor list or not. If yes, it chooses it as its next hop. If not, it checks if CV is an EV or not. If so, it firstly chooses its next intersection (NIC) from its intersection list (IL), then it selects the closest neighbor (EV neighbor) to the selected NIC as its next hop. If CV is not an EV, PCMA selects the nearest neighbor to its NIC as its next hop.

If PCMA gets stuck in a local optimum, where it does not find the neighbor that is closest than CV to its NIC among CV's neighbor-list, it uses a recovery strategy to overcome this problem. The main goal of this proposed recovery strategy is to reroute PCMA from an alternative geographical

Algorithm 2 PCMA Code

INPUT:
- Geographical path intersections list (IL),
- KB of CV

OUTPUT:
- Selected NFV,
- Geographical routing path (SV-DV),
- MSD of DV

```
Struct PCMA data                      ▹ Data used by PCMA
    SV                                ▹ Source Vehicle identifier
    IL:ARRAY                          ▹ Intersection List
    TIL:ARRAY                         ▹ Temporary Intersection List
    MSD                               ▹ DV's Minimum Stay Duration
    DSI                               ▹ DV's Segment Identifier
    NIC                               ▹ Next Intersection Candidate
    PNIC                              ▹ Previous Next Intersection Candidate
EndStruct
CALL UpdateNeighborList(NL)
if CV = DV then
    PCMA adds its routing information to CV's KB
    PCMA expires
else
    if CV IN NL then
        NFV ← SV
    else
        if CV is not an EV then
            NFV ← MinDist(n: {n ∈ NL}, NIC)
        else
            if CV is an EV then
                NIC ← SelectNIC(NIC, IL)
                NFV ← MinDist(n: {n ∈ NL and n is an
                EV}, NIC)
            else                      ▹ Local-optimum occurs,
                                      ▹ use recovery strategy
                CALL MakeOutOfService(CSI)
                TIL ← SelectWeightedPath(PNIC, IL)
                IL ← UpdateIL(IL,TIL)
                NIC ← PNIC
                NFV ← MinDist(n: {n ∈ NL and n is an
                EV}, NIC)
            end if
        end if
    end if
end if
CALL Migrate(PCMA, NFV)
```

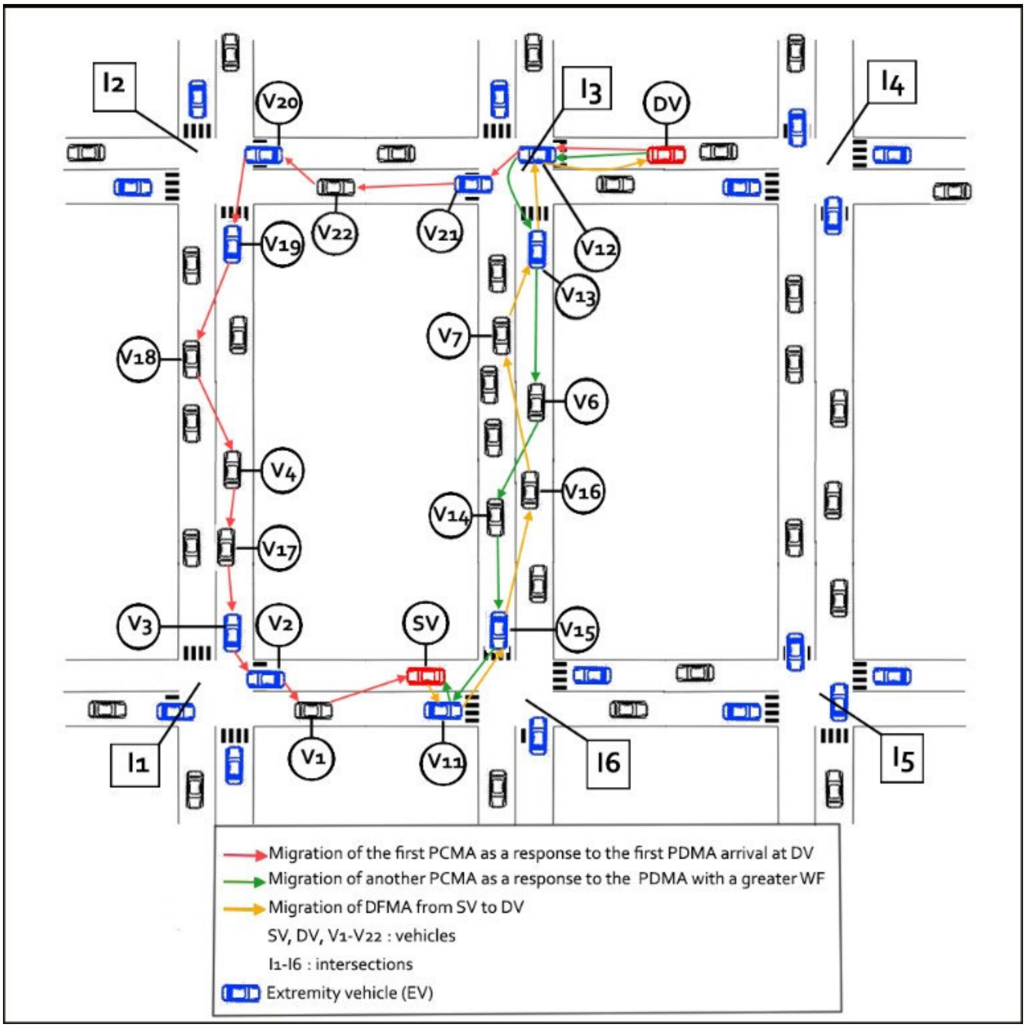

FIGURE 3. Examples of PCMA and DFMA journeys.

path when its path is interrupted and to reduce the number of migrations performed by PCMA in the recovery mode. Hence, when PCMA encounters a local optimum, it firstly marks the edge between its NIC and PNIC (PCMA's Previous NIC) as inaccessible and traces back towards the nearest vehicle to this PNIC. Before migrating, PCMA removes all the edges that it marked as out-of-service and applies the Dijkstra algorithm to find the shortest geographical paths between PNIC and each other intersections that belong to its IL and it was not able to cross them. After that, PCMA continues its migrations according to the shortest geographical path among the extracted paths. For example, suppose that the IL provided to PCMA is (I1, I2, I3, I4, I5, I6) and this latter gets stuck in a local optimum when its NIC is I3 (so its PNIC is I2). Hence, PCMA marks the edge (I2, I3) as out of service. After that, it applies the Dijkstra algorithm for finding the shortest geographical paths between (I2, I3), (I2, I4), (I2, I5), and (I2, I6) while considering the inaccessibility of the edge (I2, I3). After that, PCMA chooses among them the shortest path (e.g. (I2, I9, I4)) as its Temporary Intersection List (TIL) and merges it with its IL (in this example PCMA's IL becomes (I1, I2, I9, I4, I5, I6)). Then, PCMA continues its migrations according to this updated IL while starting with (NIC = I2). For example, if PCMA finds that the path between (I2, I6) is the shortest path among the extracted paths, it adds I5 to the end of the updated IL (I1, I2, …, I6, I5), since the main goal of PCMA's migrations is to cross the segment (edge) delimited by I5 and I6 (the segment in which SV is located).

F. DATA FORWARDING MOBILE AGENT JOURNEY

Data Packets Forwarding Mobile Agent (DFMA) is a mobile agent used for routing data from SV towards DV according to the optimum geographical path retrieved from the received PCMA(s). The tasks performed by DFMA at each visited vehicle are like those performed by PCMA. The main difference between them is the information type that is encapsulated in the agent: while PCMA aggregates the required routing information, DFMA contains the data that will be sent. Usually, a recovery strategy is used only in the routing

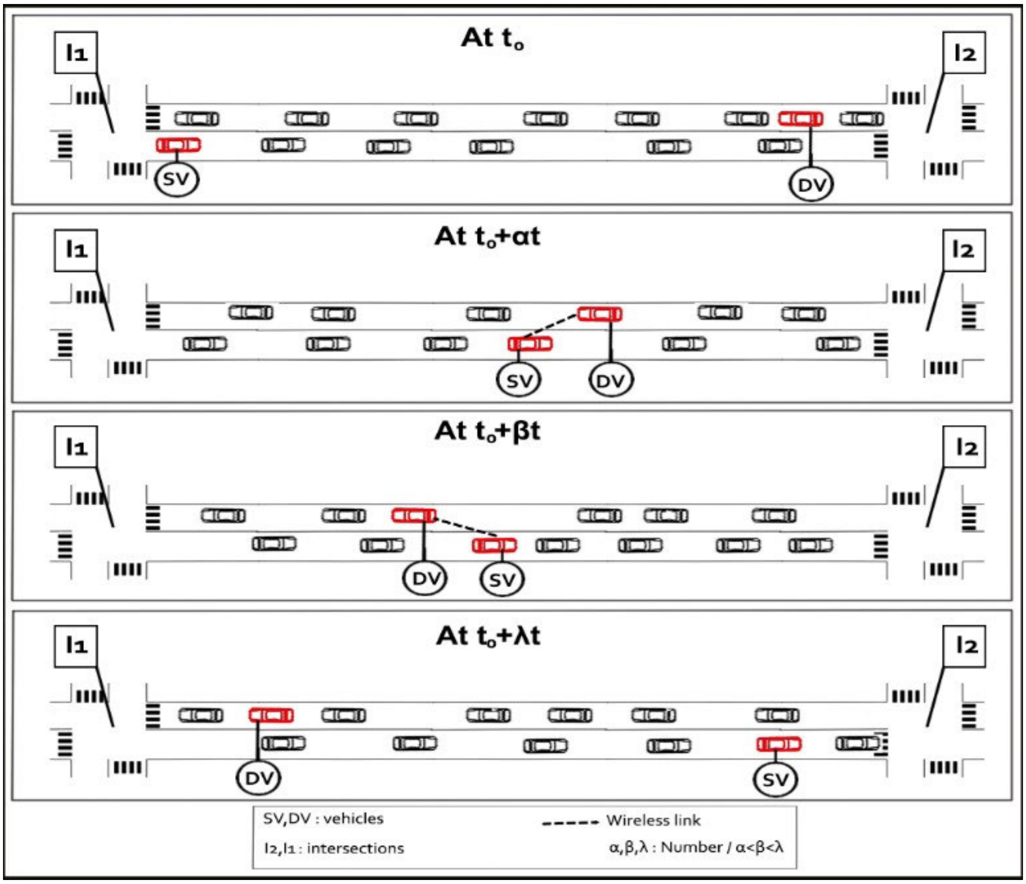

FIGURE 4. Example of routing between vehicles located in the same segment.

protocol process in order to improve the data packet delivery ratio. However, in our case, our recovery strategy is used again in the location service process in order to improve the query success ratio.

Figure 3 is a continuation of Figure 2. It illustrates the process performed by the VMSA of DV when receiving PDMA1, PDMA2, and PDMA3. It also shows the migrations performed by the PCMAs triggered from DV's VMSA towards SV and the migrations performed by the DFMA triggered from SV's VMSA towards DV.

In Figure 3, after receiving PDMA1 (The first PDMA that arrives at DV among PDMA1, PDMA2, and PDMA3), DV's VMSA creates a new PCMA and provides it with the address of SV, the MSD of DV and the inverse of the TIL retrieved from PDMA1 (I4, I3, I2, I1, I6). After triggering this PCMA, this latter chooses I3 as its NIC and starts its migrations according to Algorithm 2 until reaching SV. Firstly, PCMA updates DV's neighbor list and migrates to V12 (the closest neighbor to I3 among the neighbors located in the same segment as DV). From V12 (EV), PCMA selects I2 as its NIC and migrates to V21 (the nearest EV neighbor to I2) after updating V12's neighbor list.

This PCMA continues its migrations in the same way until reaching V1 (SV's neighbor). So, it directly migrates to SV and provides it with the required routing information. The main idea used here, so that PCMA finds SV when it reaches I1, is as follows: If SV is located in the segment S1 (delimited by I1 and I6) and PCMA has the ability to perceive all the vehicles in this segment when it passes from I1 towards I6 or vice versa (the same idea in which PDMA can perceive all the vehicles located in a road segment), then it will surely find SV as one of its next-hop neighbors.

When receiving PDMA2, DV's VMSA finds that this latter has a greater weight factor than the PDMA that preceded it (i.e., PDMA1). So, it immediately responds with a new PCMA supported with the inverse of the TIL retrieved from PDMA2 (I4, I3, I6, I1). During its migrations, this PCMA performs the same tasks as the previous PCMA. DV's VMSA will not respond with another PCMA when it receives a PDMA3 with a smaller weight factor than PDMA2.

The data packets will be routed through DFMA according to the geographical path retrieved from the last received PCMA (i.e. the PCMA triggered as a response to the most weighted PDMA). In this example, DFMA is provided with this IL (I1, I6, I3, I4) from which it performs its migrations (in the same way as PCMA) until reaching DV.

The validity period of the routing information retrieved through each received PCMA is proportional to the *MSD* encapsulated in this PCMA. If SV leaves its current segment (where it was located when it receives PCMA), it adds the last crossed intersection identifier in the IL retrieved from this PCMA.

Figure 4 shows an example scenario for DFMA's IL updates when SV and DV are located in the same segment. We assume that at t_0, the IL retrieved from the last received PCMA is (I2, I1). So, at this time, SV's VMSA triggers DFMA towards I2 (NIC = I2) for reaching DV. At $(t_0 + \alpha t)$, SV and DV become neighbors and DV is still closer to I2 than SV. Hence, the geographical path that indicates the accessibility of DV remains equal to (I1, I2). At $(t_0 + \beta t)$, SV and DV are also neighbors, but in this case, SV becomes closer to I2 than DV. Then SV reverses the geographical path that indicates the accessibility of DV (I2, I1). At $(t_0 + \lambda t)$ SV and DV are not neighbors anymore; and for SV, the geographical path that indicates the accessibility of DV remains (I2, I1). Hence, DFMA will be triggered towards I1 (NIC = I1) for reaching DV.

TABLE 2. Simulation parameters.

Parameter	Value
Channel type	Channel/WirelessChannel
Propagation model	Propagation/TwoRayGround
Network interface	Phy/WirelessPhyExt
MAC layer	802.11p
Interface queue type	Queue/DropTail/PriQueue
Link layer	LL
Antenna model	Antenna/OmniAntenna
Interface queue length	512 packets
Transmission range	250 m
Area	10 × 10 km^2
Number of nodes	50, 100, 150, 200, 250 and 300
Average node speed	10, 20, 30, 40, 50 and 60 m/s
Simulation time	10,000 s
Beacon interval	0.5 s
CBR traffic	[1, ... , 7] x 20 packets/node
CBR packet size	128B
CBR send interval	100 ms
Mobility model	Manhattan Model
Number of lanes in a segment	2
Total number of segments	40
Number of intersections	25
Length between intersections	500 m

IV. SYSTEM EVALUATION

A. SIMULATIONS SETTINGS

The simulations described below were performed using the NS-3 simulator (version 3.27). All agents used in ARGENT are implemented as NS-3 objects. In this simulation study, we have compared ARGENT with three systems: the Path Aware Greedy Perimeter Stateless Routing (PA-GPSR) that was proposed in [58], the Mobile Group-based Location Service (MoGLS) published in [62], and the Dynamic Real-time Multimodal Routing (DREAMR) protocol from [18]. In the first part of the simulations, we compare the location service of ARGENT with the Reactive Location Service (RLS) [38] that was coupled with PA-GPSR and with MoGLS. In the second part, we compare the routing characteristics of ARGENT with those of PA-GPSR and DREAMR. We chose RLS and MoGLS to compare with since they represent different approaches for location services in the IoV. From the routing perspective, PA-GPSR was chosen since it is designed as an improvement of one of the main geographic routing protocols (i.e., GPSR), while DREAMR was selected since it utilizes a dynamic hierarchical routing model.

Table 2 summarizes the parameters used in the conducted simulations. The area chosen is a 10 × 10 km^2 of the Manhattan mobility model which is used to generate roads and intersections topology [8]. This model is generated-map-based, introduced to generate moving vehicles in an urban environment.

In the simulation scenarios, each node initiates between 1 and 7 Constant Bit Rate (CBR) traffic consisting of 20 packets respectively towards $\{1, \ldots, 7\}$ random destination nodes. Each data packet has a size of 128 bytes. The sending interval between every two successive packets is set to 100 ms. The CBR traffic can be considered as an audio or video streaming that can be used, for example, in public safety applications and also in entertainment-based applications. Note that the simulation parameters were carefully selected in order to test the main objectives of ARGENT. For example, we simulated different scenarios in which the number of nodes is varied between 50 and 300 in order to test different vehicle densities and ensure the ability of our protocol to provide the required location in both sparse and dense conditions. In addition, we simulated seven scenarios with different numbers of CBR connections in order to test the performance of various applications with different data rates. We also executed separate scenarios in which the average speed of nodes was varied between 10 and 60 m/s in order to test different mobility settings. Finally, we tested several scenarios in which the distnce between the source and destination vehicles was varied between 50m and 10km. Note that each simulation scenario was repeated 10 times and the average of the ten scenarios was taken. The results of each set of scenarios are shown and discussed next.

B. SIMULATION RESULTS AND ANALYSIS

In this section, the obtained simulation results will be detailed. We have compared ARGENT with PA-GPSR combined with RLS, MoGLS, and DREAMR. The comparison is based on two performance criteria, the location service efficiency and the routing protocol efficiency.

1) LOCATION SERVICE EFFICIENCY

Location service efficiency measurement is done to compare the performance of the location service of ARGENT with those of RLS and MoGLS. We have used the number of location requests sent, the location service overhead, the Query Success Ratio (QSR), and the Location Response Time (LRT) as performance criteria for these three location services.

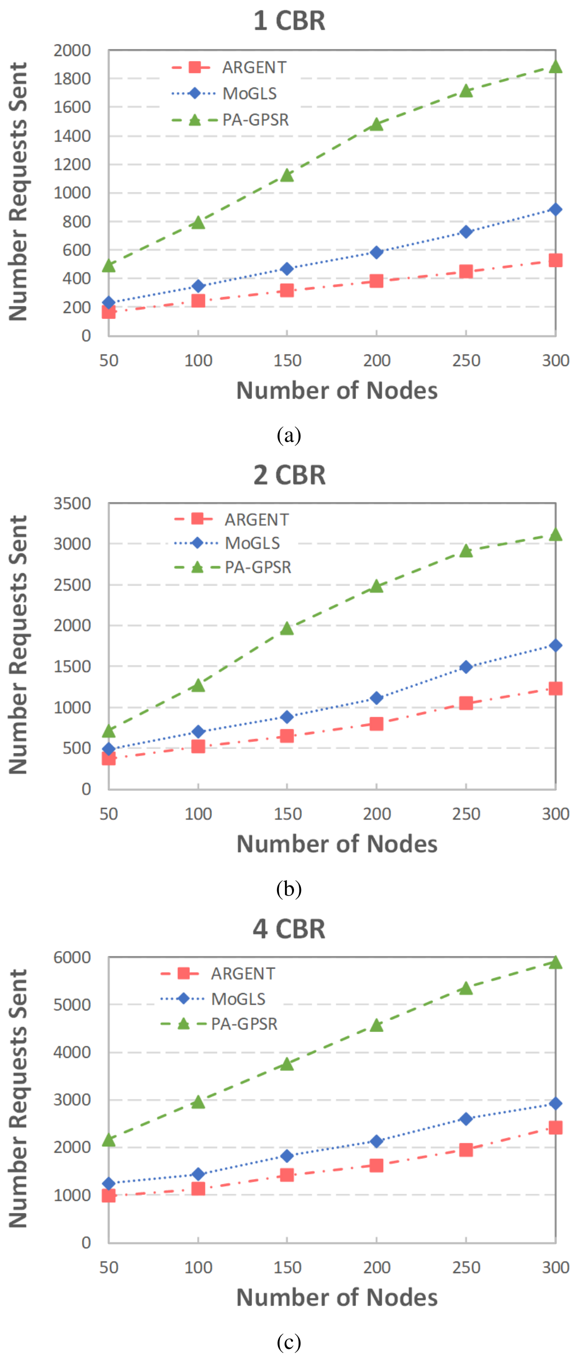

FIGURE 5. Location efficiency results: number of sent location requests for 1, 2, and 4 CBR connections.

a: NUMBER OF SENT LOCATION REQUESTS

Location requests are sent by a sender node each time it needs to send data to another destination node and it does not have fresh routing information about this latter (position for RLS and MoGLS and geographical path for ARGENT). Here, the more location requests that are sent means that communication interruptions are happening more frequently, and as a result, the number of location control messages is significantly increased. As shown in Figure 5, ARGENT reduces the number of location requests sent for the different numbers of CBR connections. On average, PA-GPSR with RLS generates two times location-request messages than ARGENT. On the other hand, the number of location-request messages generated by MoGLS is 35% greater than ARGENT, on average. This is mainly due to two main factors. Firstly, the sender node in PA-GPSR and MoGLS requires the geographical coordinates of the target node to route the data packets, so the maintained position will be invalid with each remarkable movement of this latter. In addition, the location information in MoGLS is maintained and updated at two levels: GL and RH. Hence, any inconsistency between the two levels causes the location information to become invalid, which requires the sender to request the new location of the destination.

Since the sender node in ARGENT requires the segment coordinates of the target node, the maintained position remains valid until this latter leaves its segment. Consequently, the validity period (freshness) of the retrieved routing information is generally more extended in ARGENT than in PA-GPSR and MoGLS. Secondly, the number of location requests that are not answered (presented in Figure 7) significantly increases the number of requests sent since, in our simulation scenario, each CBR connection is constructed of 20 data packets sent to a random destination node. So, if the sender node did not receive the destination node location when it needs to send a data packet, it will send another location-request message before sending the next data packet.

b: LOCATION OVERHEAD

The overhead generated by ARGENT, PA-GPSR, and MoGLS is measured here as the total number of control packets sent and forwarded during the location queries/replies process. The generated overhead directly depends on the number of requests sent, since the control messages generated by each of the three location services are produced whenever the sender node sends a location-request message. Figure 6 shows that ARGENT produces a slightly smaller number of control messages than MoGLS and less than half the control message overhead of PA-GPSR, on average. This is obtained in ARGENT due to the abandonment of the blind flooding mechanism when disseminating the location request messages and especially by reducing the sending number of these latter. On the other hand, MoGLS generates much less control message overhead than PA-GPSR due to the process of limiting the flooding of location information in MoGLS to the small group of GLs and to the RH in each region. However, MoGLS still produces a slightly higher number of control messages than ARGENT due to this limited flooding, as shown in Figure 6.

c: QUERY SUCCESS RATIO (QSR)

The QSR represents the ratio of the queries answered with valid location information from all those sent. As depicted in Figure 7, ARGENT achieves a higher QSR than PA-GPSR for the different vehicular density scenarios (Up to 33% relative improvement). On the other hand, the QSRs of ARGENT and MoGLS are similar, on average. The QSR of ARGENT

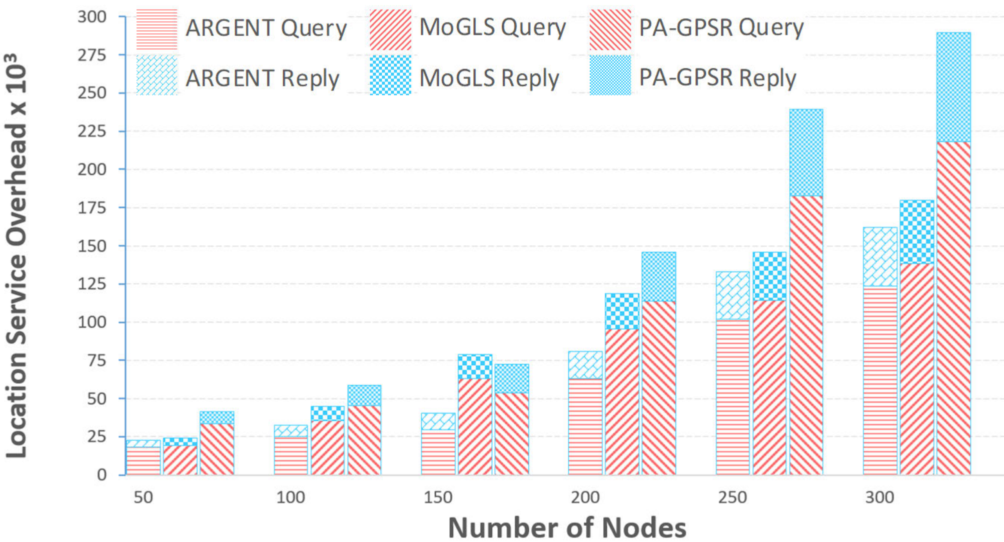

FIGURE 6. Location efficiency results: location overhead [2 CBR conn].

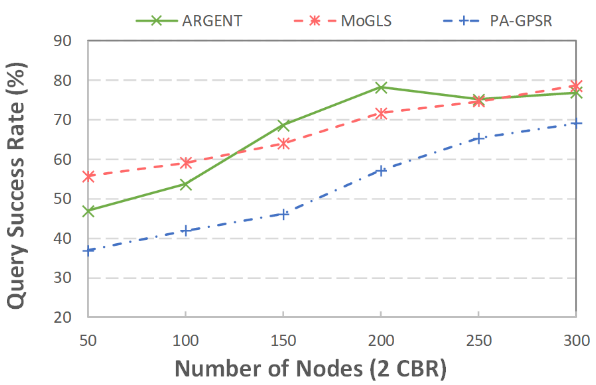

FIGURE 7. Location efficiency results: location query success ratio while varying the number of nodes.

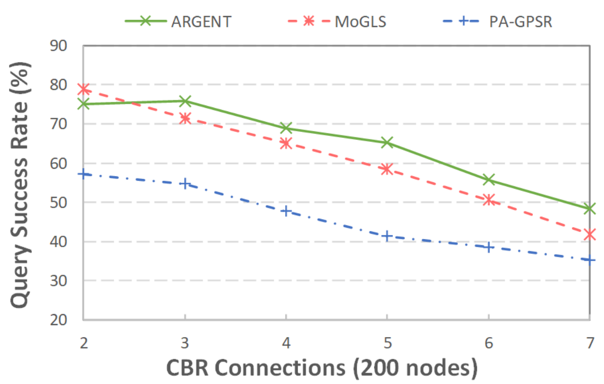

FIGURE 8. Location efficiency results: location query success ratio while varying the number of CBR connections.

is high thanks to several factors inherent in ARGENT's operation mechanism such as the multiplicity of responses sent by the target node to answer a single request sent by the source node, the movement absorption mechanism that reduces the number of lost packets and the recovery strategy used whenever the response gets stuck in a local optimum. With respect to MoGLS, it also achieves a high QSR due to the efficiency of the two-level hierarchical location update mechanism. However, the QSR of ARGENT is higher than or equal to that of MoGLS when the vehicle density is medium or high. Finally, it is observed in this figure that location requests are answered more successfully as the vehicular density increases. This applies to all the location services. This is mainly due to the fact that when the number of nodes increases, the network connectivity will be improved and so, the probability of encountered responses in a local optimum will be decreased.

At high network densities (especially when the number of nodes exceeds 200) the QSR improvement ratio of ARGENT is relatively reduced. This is expected since the generated communication overhead is rapidly increased when reaching this number of nodes, which causes radio interference and collisions. This result is more observed when we increase the number of CBR connections, as depicted in Figure 8. For 3 CBR connections, ARGENT achieves a 76% of query success rate, MoGLS achieves 71%, while 55% is achieved by PA-GPSR. When increasing the number of CBR connections to 7, ARGENT's query success rate is decreased to 48%, while those of MoGLS and PA-GPSR drop to 42% and 35%, respectively. The figure illustrates that ARGENT's location service is less affected by the increase in network traffic than MoGLS and RLS.

d: LOCATION RESPONSE TIME (LRT)

The LRT represents the waiting period that passes after the sender node sends a location request until receiving a response. The LRT is considered here as an indicator of the effective response time of the compared location services (RLS, MoGLS, and ARGENT). As shown in Figure 9,

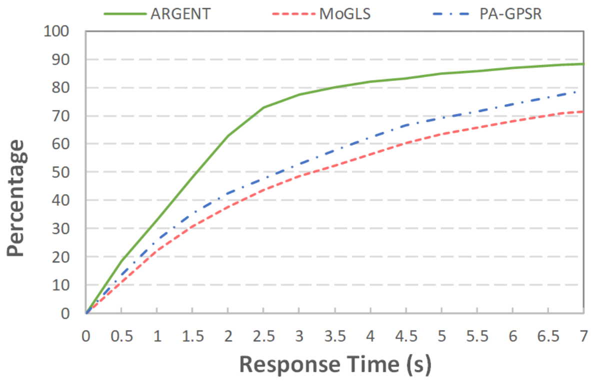

FIGURE 9. Location efficiency results: Location response time [200 nodes].

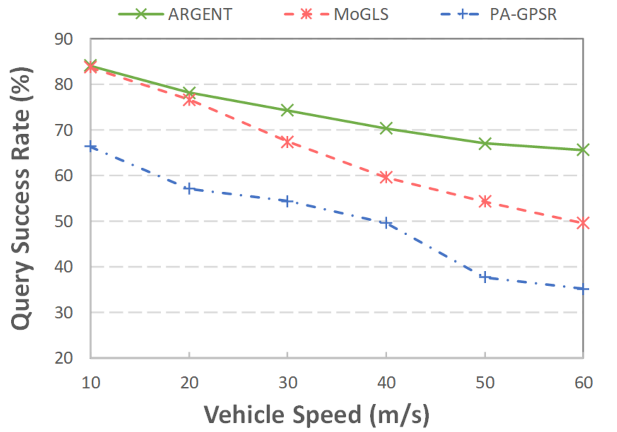

FIGURE 10. Location efficiency results: location query success ratio while varying the average vehicle speed.

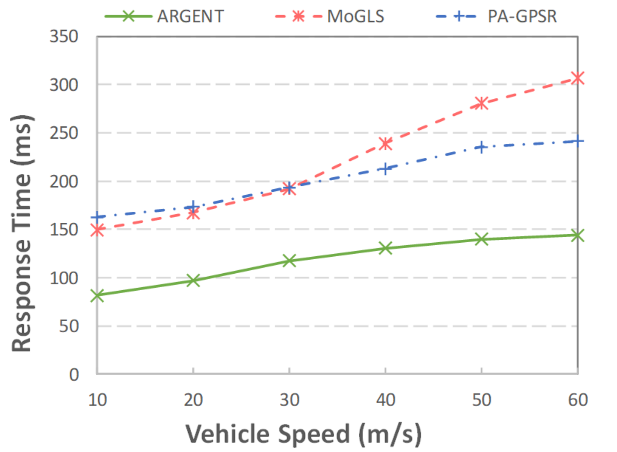

FIGURE 11. Location efficiency results: location query response time while varying the average vehicle speed.

only 31% and 35% of the responses are received after 1.5s of the start of request sending in MoGLS and PA-GPSR, respectively. However, 47% are received after the same time interval in ARGENT. The processes of internal and external agent migration speed up the location request and reply in ARGENT and enable the source to receive the destination's location much faster than traditional location request/reply in PA-GPSR. On the other hand, the process of searching for the location information at the GL and RH levels increases the location request time in MoGLS significantly, as illustrated in Figure 9.

e: VEHICLES' SPEED

In the previous results, the average speed of vehicles was set to 20 m/s in the simulation scenarios. One of the main objectives of ARGENT is to perform efficiently during various circumstances, including the case when vehicles are moving at a high speed. In fact, a major factor that reduces the efficiency of many location services and routing protocols is their reduced performance at high vehicles' speeds. For this purpose, we conducted several simulation scenarios in which the average vehicle speed was varied between 10 and 60 m/s. Figures 10 and 11 illustrate the effect of increasing the speed on the location query success ratio and the LRT.

As the average speed of vehicles increases, the disconnectivity rate between them increases. In other words, the higher speed makes the average connection time between two vehicles shorter. This decreases the time during which a route is valid, as the probability of disconnection between two vehicles on the route increases. Hence, each vehicle will send location requests at a higher rate. In addition, the frequent disconnection makes the lifetime of a routing path shorter. Hence, the QSR decreases. This is shown in Figure 10, which illustrates that ARGENT's QSR decreases from 84% to 66%, while that of MoGLS decreases from 83% to 49%, and that of PA-GPSR decreases from 67% to 35%. Hence, we notice that ARGENT is less affected by high speeds, mainly due to its higher QSR in normal conditions and the higher lifetime of its routing paths, as previously explained. At low speeds, ARGENT and MoGLS achieve similar QSRs. However, the QSR of MoGLS decreases quickly when the speed increases, while that of ARGENT remains much higher. MoGLS is much affected by high speed since in such cases, the destination vehicle has a higher probability to move from one region to another or from one vehicle group to another, which requires updating its location information at all GLs in the region and at the RH.

As for the LRT, we notice from Figure 11 that location responses are delivered by ARGENT much faster than MoGLS and PA-GPSR. In addition, the LRTs of MoGLS and PA-GPSR increases much more than ARGENT at high speeds. The difference between the LRTs of the ARGENT and PA-GPSR is equal to 100ms when the average speed is 60 m/s, while that between the LRTs of ARGENT and MoGLS is 163ms, which makes the LRTs of PA-GPSR and MoGLS approximately double that of ARGENT at this speed. This illustrates the ability of ARGENT to perform efficiently at high vehicles' speeds, due to the utilization of smart agents that make full use of the roads structure to reduce the packet delay and increase the route lifetime. With respect to MoGLS, it was designed for low vehicle speeds (up to 14m/s, as the

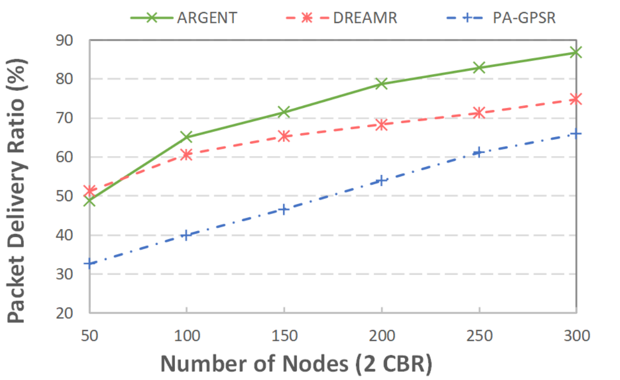

FIGURE 12. **Routing efficiency results: packet delivery ratio while varying the number of nodes.**

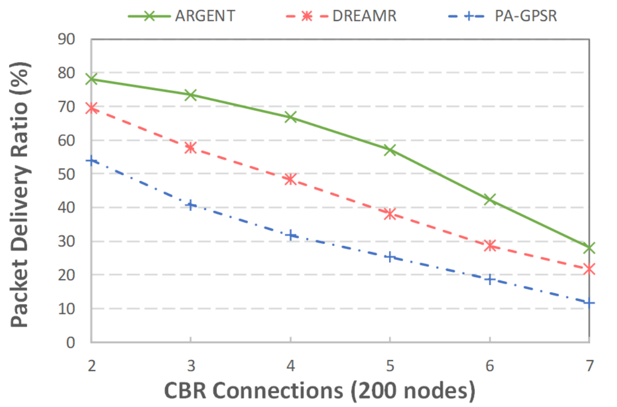

FIGURE 13. **Routing efficiency results: packet delivery ratio while varying the number of CBR connections.**

authors of [62] explain). When tested in high-speed scenarios, the hierarchical model of MoGLS shows to be less performance efficient, as Figures 10 and 11 illustrate.

2) ROUTING EFFICIENCY

Geographic routing efficiency depends directly on the location information provided through the location service. In general, more efficiency at the location service level results in more efficiency at the position-based routing level. We evaluated the routing efficiency using the Packet Delivery Ratio (PDR) and the average CBR latency as performance criteria. We compared ARGENT with the PA-GPSR and DREAMR routing protocols.

a: PACKET DELIVERY RATIO (PDR)

The PDR represents the reception rate of CBR packets with respect to all the CBR packets sent. Figure 12 shows that ARGENT has a higher PDR than PA-GPSR for the different values of vehicle density. As for DREAMR, it achieves a similar PDR to ARGENT for low vehicle densities. However, ARGENT has a much higher PDR than DREAMR when the vehicle density is high. On average, ARGENT improves the PDR by about 22% compared to PA-GPSR and 7% compared to DREAMR. This is mainly due to the fact that ARGENT routes its data packets following the geographical path that has the best connectivity degree, and due to other characteristics presented in Section III-C, such as using the same algorithm when forwarding location response and data packets, and maintaining the correctness of the routing path as long as the vehicle stays on the same road section. Moreover, this figure shows the impact of the control message overhead generated by RLS on the delivery ratio. Indeed, the results produced by the authors of [58] show that PA-GPSR produces a PDR of 63% when it doesn't consider the overhead produced by the RLS, which is 23% higher than its PDR in Figure 13 at the same number of nodes. This impact is reduced at higher network densities due to the availability of more routing paths, which mitigates the effect of the control traffic overhead. As for DREAMR, its PDR is affected by the increase in vehicle density due to the overhead that the latter adds to the optimization problem that is used by DREAMR to determine the optimal routing path. As explained in [18], DREAMR uses graph exploration to determine all possible routes. As the vehicle density increases, the number of possible routes highly increases, which increases the time required to find the optimal route, increases the latency (see Figure 14), and makes the duration of the correctness of the routing path smaller. This in turn affects the PDR, which decreases as the optimal routing path is more frequently calculated.

In Figure 13, the impact of location-service overhead on the delivery ratio is more remarkable. In this figure, it is observed that all three protocols are highly affected by CBR connection augmentation. When changing the number of CBR connections from 2 to 7, ARGENT's PDR decreases from 78% to 28%, while the PDRs of DREAMR and PA-GPSR decrease from 69% to 21% and 54% to 12%, respectively. The figure illustrates that regardless of the protocol's routing approach, the increase in the network traffic decreases the probability of successful data packet delivery, since the network congestion leads to the dropping of data packets when the queue of a node is full.

b: AVERAGE CBR LATENCY

The average CBR packet latency is calculated as the average time needed by a data packet to reach the destination node. As depicted in Figure 14, the average latency is decreased, on average, from 4.6s in PA-GPSR and 4.1s in DREAMR to 2.68s in ARGENT. This is because ARGENT generates less control message overhead, so it causes less network overload. When the number of nodes is small, the three protocols produce similar latencies. However, as the number of nodes increases, the latencies of PA-GPSR and DREAMR increase much more than that of ARGENT. At 300 nodes, the latency of the latter is 33.5% less than that of PA-GPSR and 22% less than the latency of DREAMR, which indicates the efficiency of the DFMA approach applied by ARGENT and its ability to deliver the data packet to the destination much faster than the routing approaches of PA-GPSR and DREAMR. In addition, when the number of nodes is high, the traffic density

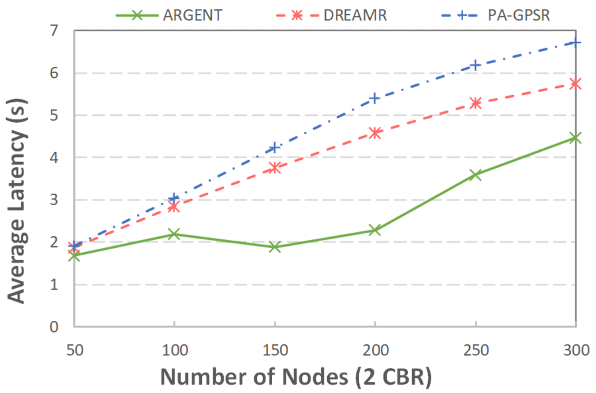

FIGURE 14. Routing efficiency results: average CBR packets latency while varying the number of nodes.

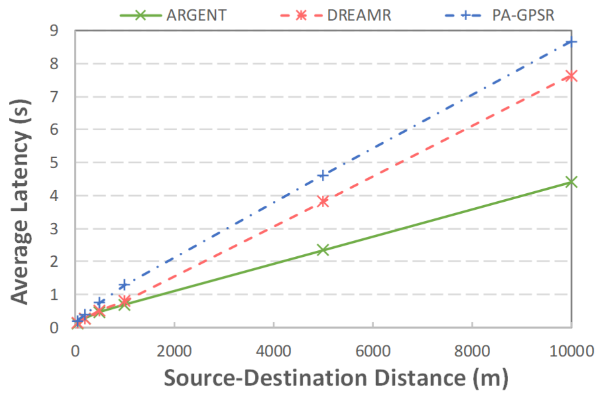

FIGURE 16. Routing efficiency results: average CBR packets latency while varying the S-D Distance.

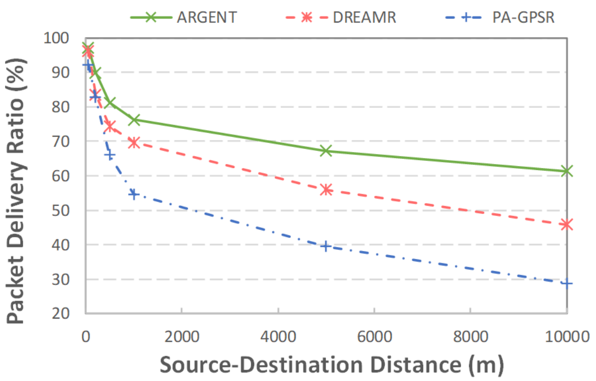

FIGURE 15. Routing efficiency results: packet delivery ratio while varying the S-D Distance.

helps to decrease the network latency of ARGENT since the recovery strategy, which adds some additional latency in order to reroute data from an alternative path, will be less used. However, the traffic density has a negative effect on DREAMR, as we explained in the previous paragraphs.

c: VARYING THE SOURCE-TO-DESTINATION (S-D) DISTANCE

In the last set of simulation scenarios, we study the effect of the distance between the source and destination vehicles on the routing protocol performance. For this purpose, we conducted a set of simulation scenarios in which the source vehicle selects a destination vehicle that is a specific distance away from it. The S-D Distance was varied between 50m and 10km. Figure 15 shows that the PDRs of the three protocols decrease as the S-D Distance increases. However, the PDR of ARGENT decreases less than those of PA-GPSR and DREAMR. This shows that ARGENT has a higher ability to deliver packets successfully to distant destinations. This is mainly due to the routing approach of ARGENT that routes the packet to the destination segment and then to the destination vehicle, which avoids updating the whole routing path as long as the destination vehicle remains on the same road segment.

Finally, Figure 16 illustrates that the CBR latency increases linearly with the S-D Distance for the three protocols. The latter achieve similar latencies when the S-D Distance is small. However, as the S-D Distance increases, the CBR latencies of PA-GPSR and DREAMR increase much more than the latency of ARGENT. This can be considered one of the main advantages of ARGENT, which is the ability to deliver data packets to far destinations much faster than the other protocols. As explained previously, as long as the destination vehicle remains on the same road segment, the routing path at the source vehicle remains valid, and the data packet can be delivered successfully to the destination segment. After that, the vehicles at the destination segment, who update their routing paths to the destination vehicle more frequently, can successfully route the packet within the segment to the destination vehicle. This approach avoids recalculating the whole path at the source vehicle, as other routing protocols do, and plays an important factor in decreasing the routing latency, especially for distant destinations.

V. CONCLUSION

In this paper, we presented an Agent-based Reactive Geographic Routing Protocol (ARGENT) which aims to improve the geographic routing performance through a smart combination of routing mechanisms with a novel lightweight location-based service. The proposed routing protocol works into two combined processes. First, it is used to search for the location of a target node and to find the optimal geographical path (in terms of distance and density) to reach it. Second, it is used to route the data packets from a source vehicle to the target vehicle according to the optimum geographical path. The multiagent-based approach integrates static and mobile agents in order to provide more adaptability, flexibility, and personalization of routing services within IoV environments. Simulations showed that ARGENT outperforms other location-based services, mainly RLS and MoGLS, in terms of communication interruptions, Query Success Ratio, Location Response Time, and Location overhead. In addition, ARGENT outperforms recent routing protocols, such as DREAMR and PA-GPSR, in terms of data packet delivery ratio and data packet end-to-end latency. Some enhancements will be made in the future extension of this work, such as studying the security of ARGENT exchanged messages, and exploring the effect of varying the routing path caching time on the protocol performance.

ACKNOWLEDGMENT

The authors would like to acknowledge Abdelmonom Hajjej from FSG, Gabes, Tunisia for his help in this work.

REFERENCES

[1] R. Aissaoui, A. Dhraief, A. Belghith, H. Menouar, H. Mathkour, F. Filali, and A. Abu-Dayya, ''HCBLS: A hierarchical cluster-based location service in urban environment,'' *Mobile Inf. Syst.*, vol. 2015, pp. 1–16, Jan. 2015.

[2] Y. R. B. Al-Mayouf, N. F. Abdullah, O. A. Mahdi, S. Khan, M. Ismail, M. Guizani, and S. H. Ahmed, ''Real-time intersection-based segment aware routing algorithm for urban vehicular networks,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 7, pp. 2125–2141, Jul. 2018.

[3] Y. R. B. Al-Mayouf, O. A. Mahdi, and D. M. Uliyan, ''An intersection-based segment aware algorithm for geographic routing in VANETs,'' in *Proc. 9th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2018, pp. 66–71.

[4] M. Al-Rabayah and R. Malaney, ''A new hybrid location-based ad hoc routing protocol,'' in *Proc. IEEE Global Telecommun. Conf. GLOBECOM*, Dec. 2010, pp. 1–6.

[5] S. Asoudeh, M. Mehrjoo, N.-M. Balouchzahi, and A. Bejarzahi, ''Location service implementation in vehicular networks by nodes clustering in urban environments,'' *Veh. Commun.*, vol. 9, pp. 109–114, Jul. 2017.

[6] M. Ayaida, M. Barhoumi, H. Fouchal, Y. Ghamri-Doudane, and L. Afilal, ''Joint routing and location-based service in VANETs,'' *J. Parallel Distrib. Comput.*, vol. 74, no. 2, pp. 2077–2087, Feb. 2014.

[7] M. Ayaida, H. Fouchal, L. Afilal, and Y. Ghamri-Doudane, ''A comparison of reactive, grid and hierarchical location-based services for VANETs,'' in *Proc. IEEE Veh. Technol. Conf.*, Sep. 2012, pp. 1–5.

[8] F. Bai, N. Sadagopan, and A. Helmy, ''IMPORTANT: A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks,'' in *Proc. 22nd Annu. Joint Conf. IEEE Comput. Commun. Societies*, Mar. 2003, pp. 825–835.

[9] S. Basagni, I. Chlamtac, and V. R. Syrotiuk, ''Geographic messaging in wireless ad hoc networks,'' in *Proc. IEEE 49th Veh. Technol. Conf.*, May 1999, pp. 1957–1961.

[10] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, ''A distance routing effect algorithm for mobility (DREAM),'' in *Proc. 4th Annu. ACM/IEEE Int. Conf. Mobile Comput. Netw.*, Oct. 1998, pp. 76–84.

[11] F. Belamri, S. Boulfekhar, and D. Aissani, ''A survey on QoS routing protocols in vehicular ad hoc network (VANET),'' *Telecommun. Syst.*, vol. 78, no. 1, pp. 117–153, Sep. 2021.

[12] A. Bengag and M. El Boukhari, ''Classification and comparison of routing protocols in VANETs,'' in *Proc. Int. Conf. Intell. Syst. Comput. Vis. (ISCV)*, Apr. 2018, pp. 1–8.

[13] A. Boukerche, *Algorithms and Protocols for Wireless and Mobile Ad Hoc Networks*. Hoboken, NJ, USA: Wiley, 2008.

[14] C. Castelfranchi and E. Lorini, ''Cognitive anatomy and functions of expectations,'' in *Proc. Workshop Cognit. Model. Agents Multi-Agent Interact.*, 2003, pp. 1–9.

[15] O. Cheikhrouhou, K. Mershad, F. Jamil, R. Mahmud, A. Koubaa, and S. R. Moosavi, ''A lightweight blockchain and fog-enabled secure remote patient monitoring system,'' *Internet Things*, vol. 22, Jul. 2023, Art. no. 100691.

[16] C. Chen, L. Liu, T. Qiu, D. O. Wu, and Z. Ren, ''Delay-aware grid-based geographic routing in urban VANETs: A backbone approach,'' *IEEE/ACM Trans. Netw.*, vol. 27, no. 6, pp. 2324–2337, Dec. 2019.

[17] C. Chen, L. Liu, T. Qiu, K. Yang, F. Gong, and H. Song, ''ASGR: An artificial spider-web-based geographic routing in heterogeneous vehicular networks,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1604–1620, May 2019.

[18] S. Choudhury, J. P. Knickerbocker, and M. J. Kochenderfer, ''Dynamic real-time multimodal routing with hierarchical hybrid planning,'' in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 2397–2404.

[19] S. Din, K. N. Qureshi, M. S. Afsar, J. J. P. C. Rodrigues, A. Ahmad, and G. S. Choi, ''Beaconless traffic-aware geographical routing protocol for intelligent transportation system,'' *IEEE Access*, vol. 8, pp. 187671–187686, 2020.

[20] A. Dorri, S. S. Kanhere, and R. Jurdak, ''Multi-agent systems: A survey,'' *IEEE Access*, vol. 6, pp. 28573–28593, 2018.

[21] A. Fonseca and T. Vazão, ''Applicability of position-based routing for VANET in highways and urban environment,'' *J. Netw. Comput. Appl.*, vol. 36, no. 3, pp. 961–973, May 2013.

[22] S. Franklin and A. Graesser, ''Is it an agent, or just a program? A taxonomy for autonomous agents,'' in *Proc. Int. Workshop Agent Theories, Architectures, Lang.* Cham, Switzerland: Springer, 1996, pp. 21–35.

[23] S. Fünfrocken and F. Mattern, ''Mobile agents as an architectural concept for internet-based distributed applications,'' in *Kommunikation in Verteilten Systemen (KiVS)*. Berlin, Germany: Springer, 1999, pp. 32–43.

[24] V. Gautam, ''Analysis and application of vehicular ad hoc network as intelligent transportation system,'' in *Mobile Radio Communications and 5G Networks*. Berlin, Germany: Springer, 2021, pp. 1–17.

[25] M. Gillani, H. A. Niaz, M. U. Farooq, and A. Ullah, ''Data collection protocols for VANETs: A survey,'' *Complex Intell. Syst.*, vol. 8, no. 3, pp. 2593–2622, 2022.

[26] M. L. Griss and G. Pour, ''Accelerating development with agent components,'' *Computer*, vol. 34, no. 5, pp. 37–43, May 2001.

[27] Z. J. Haas and B. Liang, ''Ad hoc mobility management with uniform quorum systems,'' *IEEE/ACM Trans. Netw.*, vol. 7, no. 2, pp. 228–240, Apr. 1999.

[28] M. S. Haghighi and Z. Aziminejad, ''Highly anonymous mobility-tolerant location-based onion routing for VANETs,'' *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2582–2590, Apr. 2020.

[29] A. N. Hassan, A. H. Abdullah, O. Kaiwartya, Y. Cao, and D. K. Sheet, ''Multi-metric geographic routing for vehicular ad hoc networks,'' *Wireless Netw.*, vol. 24, no. 7, pp. 2763–2779, Oct. 2018.

[30] E. Heidari, A. Gladisch, B. Moshiri, and D. Tavangarian, ''Survey on location information services for vehicular communication networks,'' *Wireless Netw.*, vol. 20, no. 5, pp. 1085–1105, Jul. 2014.

[31] A. R. Ilkhechi, I. Korpeoglu, U. Güdükbay, and Ö. Ulusoy, ''PETAL: A fully distributed location service for wireless ad hoc networks,'' *J. Netw. Comput. Appl.*, vol. 83, pp. 1–11, Apr. 2017.

[32] W. Jabbar, R. Malaney, and S. Yan, ''A location verification based hybrid routing protocol for VANETs,'' in *Proc. IEEE 92nd Veh. Technol. Conf. (VTC-Fall)*, Nov. 2020, pp. 1–6.

[33] N. R. Jennings, ''An agent-based approach for building complex software systems,'' *Commun. ACM*, vol. 44, no. 4, pp. 35–41, Apr. 2001.

[34] N. R. Jennings, K. Sycara, and M. Wooldridge, ''A roadmap of agent research and development,'' *Auto. Agents Multi-Agent Syst.*, vol. 1, pp. 7–38, Jan. 1998.

[35] M. Jerbi, S.-M. Senouci, R. Meraihi, and Y. Ghamri-Doudane, ''An improved vehicular ad hoc routing protocol for city environments,'' in *Proc. IEEE Int. Conf. Commun.*, Jun. 2007, pp. 3972–3979.

[36] R. Karimi and S. Shokrollahi, ''PGRP: Predictive geographic routing protocol for VANETs,'' *Comput. Netw.*, vol. 141, pp. 67–81, Aug. 2018.

[37] B. Karp and H.-T. Kung, ''GPSR: Greedy perimeter stateless routing for wireless networks,'' in *Proc. 6th Annu. Int. Conf. Mobile Comput. Netw.*, 2000, pp. 243–254.

[38] M. Käsemann, H. Füßler, H. Hartenstein, and M. Mauve, ''A reactive location service for mobile ad hoc networks,'' Dept. Comput. Sci., Univ. Mannheim, Mannheim, Germany, Tech. Rep. TR-02-014, 2002.

[39] T. Kayarga and S. A. Kumar, ''A study on various technologies to solve the routing problem in Internet of Vehicles (IoV),'' *Wireless Pers. Commun.*, vol. 119, no. 1, pp. 459–487, Jul. 2021.

[40] Y.-B. Ko and N. H. Vaidya, ''Location-aided routing (LAR) in mobile ad hoc networks,'' in *Proc. 4th Annu. ACM/IEEE Int. Conf. Mobile Comput. Netw.*, Oct. 1998, pp. 66–75.

[41] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, ''A scalable location service for geographic ad hoc routing,'' in *Proc. 6th Annu. Int. Conf. Mobile Comput. Netw.*, Aug. 2000, pp. 120–130.

[42] L. Li, X. Wang, and X. Ma, ''Design of a location-based opportunistic geographic routing protocol,'' *Comput. Commun.*, vol. 181, pp. 357–364, Jan. 2022.

[43] C. Lochert, H. Hartenstein, J. Tian, H. Fussler, D. Hermann, and M. Mauve, ''A routing strategy for vehicular ad hoc networks in city environments,'' in *Proc. IEEE 4th Intell. Vehicles Symp.*, Jun. 2003, pp. 156–161.

[44] T. Lu, S. Chang, and W. Li, ''Fog computing enabling geographic routing for urban area vehicular network,'' *Peer-Peer Netw. Appl.*, vol. 11, no. 4, pp. 749–755, Jul. 2018.

[45] M. Mauve, J. Widmer, and H. Hartenstein, ''A survey on position-based routing in mobile ad hoc networks,'' *IEEE Netw.*, vol. 15, no. 6, pp. 30–39, Nov. 2001.

[46] K. Mershad, "PROACT: Parallel multi-miner proof of accumulated trust protocol for Internet of Drones," *Veh. Commun.*, vol. 36, Aug. 2022, Art. no. 100495.
[47] K. Mershad, O. Cheikhrouhou, and L. Ismail, "Proof of accumulated trust: A new consensus protocol for the security of the IoV," *Veh. Commun.*, vol. 32, Dec. 2021, Art. no. 100392.
[48] K. Mershad and H. Dahrouj, "Blockchain model for environment/infrastructure monitoring in cloud-enabled high-altitude platform systems," *Veh. Commun.*, vol. 42, Aug. 2023, Art. no. 100627.
[49] K. Mershad and B. Said, "A blockchain model for secure communications in Internet of Vehicles," in *Proc. IEEE/ACS 17th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Nov. 2020, pp. 1–6.
[50] P. Mühlethaler, É. Renault, and S. Boumerdassi, "Design and evaluation of flooding-based location service in vehicular ad hoc networks," *Sensors*, vol. 20, no. 8, p. 2389, Apr. 2020.
[51] M. S. D. Nadella, M. R. Vatambeti, and M. V. Bobba, "Enhanced reactive location service for packet delivery in vehicular ad-hoc networks," *Int. J. Eng. Technol.*, vol. 9, no. 4, pp. 2984–2989, Aug. 2017.
[52] T. Nebbou, M. Lehsaini, H. Fouchal, and M. Ayaida, "An urban location service for vehicular area networks," *Concurrency Comput., Pract. Exper.*, vol. 31, no. 24, p. e4693, Dec. 2019.
[53] V. Anh Pham and A. Karmouch, "Mobile software agents: An overview," *IEEE Commun. Mag.*, vol. 36, no. 7, pp. 26–37, Jul. 1998.
[54] Y. Pramitarini, T. Tran, and B. An, "Energy consumption location-based QoS routing protocol for vehicular ad-hoc networks," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2021, pp. 1266–1270.
[55] S. Schmidt and A. Scott, "QoS support within active LARA++ routers," in *Proc. GEMISIS*, 2000, pp. 1–7.
[56] B.-C. Seet, G. Liu, B.-S. Lee, C.-H. Foh, K.-J. Wong, and K.-K. Lee, "A-star: A mobile ad hoc routing strategy for metropolis vehicular communications," in *Proc. Int. Conf. Res. Netw.* Cham, Switzerland: Springer, 2004, pp. 989–999.
[57] H. Shanthi and E. M. Anita, "Secure and efficient distance effect routing algorithm for mobility (SE_DREAM) in MANETs," in *Proc. 3rd Int. Symp. Big Data Cloud Comput. Challenges*. Cham, Switzerland: Springer, 2016, pp. 65–80.
[58] A. Silva, N. Reza, and A. Oliveira, "Improvement and performance evaluation of GPSR-based routing techniques for vehicular ad hoc networks," *IEEE Access*, vol. 7, pp. 21722–21733, 2019.
[59] C. Sommer, I. Dietrich, and F. Dressler, "Simulation of ad hoc routing protocols using OMNeT++," *Mobile Netw. Appl.*, vol. 15, no. 6, pp. 786–801, Dec. 2010.
[60] K. S. Kumar, A. R. Mani, S. Sundaresan, and T. A. Kumar, "Modeling of VANET for future generation transportation system through edge/fog/cloud computing powered by 6G," in *Cloud and IoT-Based Vehicular Ad Hoc Networks*, 1st ed. Hoboken, NJ, USA: Wiley, Apr. 2021, ch. 6, pp. 105–124.
[61] Y. Wang, X. Li, X. Zhang, X. Liu, and J. Weng, "ARPLR: An all-round and highly privacy-preserving location-based routing scheme for VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16558–16575, Sep. 2022.
[62] H. Woo and M. Lee, "A hierarchical location service architecture for VANET with aggregated location update," *Comput. Commun.*, vol. 125, pp. 38–55, Jul. 2018.
[63] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (VANETS): Status, results, and challenges," *Telecommun. Syst.*, vol. 50, no. 4, pp. 217–241, Aug. 2012.
[64] X. Zou, B. Ramamurthy, and S. Magliveras, "Routing techniques in wireless ad hoc networks classification and comparison," in *Proc. 6th World Multiconference Systemics, Cybernetics Informatics*, 2002, pp. 1–6.

MOHAMED MAZOUZI was born in Sfax, Tunisia, in 1980. He received the degree in computer science engineering from the National School of Computer Sciences, Tunisia, in 2004, and the master's degree in new technologies of dedicated computer systems and the Ph.D. degree in computer science from the National School of Engineering of Sfax, in 2006 and 2013, respectively. Since 2013, he has been an Assistant Professor with the Computer Science and Quantitative Methods Department, Higher Institute of Business Administration of Sfax. His current research interests include industrial networks for real-time, distributed applications, and vehicular ad-hoc networks.

KHALEEL MERSHAD (Member, IEEE) received the B.E. degree in computer engineering and informatics from Beirut Arab University, Lebanon, in 2004, and the M.E. and Ph.D. degrees in computer and communications engineering from the American University of Beirut (AUB), in 2007 and 2012, respectively. He was a Research Director with AUB, from 2012 to 2013, and Qatar University, from 2013 to 2016. He was the Chairperson of the Computer Science Department, Modern University for Business and Science (MUBS), from 2020 to 2022. He is currently an Assistant Professor with Lebanese American University (LAU). He has been involved in many research projects in the fields of mobile computing, cybersecurity and blockchain, and wireless networks. He published more than 50 scientific papers in highly reputed journals and conferences related to computer science and engineering. His research interests include ad hoc networks, vehicular communications, the Internet of Things, parallel and distributed computing, cloud computing, blockchain, and educational technology.

OMAR CHEIKHROUHOU received the Ph.D. degree in computer science from the National School of Engineers of Sfax, in March 2012. His Ph.D. deals with the security of wireless sensor networks and more precisely in secure group communication in wireless sensor networks. He is currently an Assistant Professor with the Higher Institute of Computer Science of Mahdia, University of Monastir, Tunisia. He is also a Researcher with the Computer and Embedded System Laboratory (CES Laboratory), University of Sfax, National School of Engineers, Tunisia. He has several publications in high-quality international journals and conferences. His research interests include wireless sensor networks, cybersecurity, edge computing, blockchain, multi-robot system coordination, and smart and secure healthcare. He has received some awards, including the "Governor Prize" from the Governor of Sfax, in 2005. He was selected amongst the top 2% of the most cited research, in 2021.

MONIA HAMDI received the B.Eng. degree in information technology from Telecom SudParis, Paris-Saclay University, France, in 2008, the M.Sc. degree in telecommunications and networks from Institut National Polytechnique, Toulouse, France, in 2008, and the Ph.D. degree in computer science from the University of Rennes 1, France, in 2012. From 2012 to 2017, she was an Assistant Professor with the Higher Institute of Computer Science and Multimedia, Gabés University, Tunisia. From March 2015 to August 2015, she was a Visiting Researcher with the Department of Science and Technology, Linköping University, Sweden. She is currently an Associate Professor with the College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Saudi Arabia. Her research interests include mobile communications, wireless sensor networks, edge computing, and blockchain.