IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 29, 2020, accepted September 7, 2020, date of publication September 18, 2020,
date of current version September 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3024541

Efficient Time-Stable Geocast Routing in Delay-Tolerant Vehicular Ad-Hoc Networks

LUIS R. GALLEGO-TERCERO, ROLANDO MENCHACA-MENDEZ, (Member, IEEE), MARIO E. RIVERO-ANGELES, (Member, IEEE), AND RICARDO MENCHACA-MENDEZ, (Member, IEEE)

Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico City 07738, Mexico

Corresponding author: Rolando Menchaca-Mendez (rmen@cic.ipn.mx)

This work was supported in part by the Mexican National Council for Science and Technology (CONACyT) and in part by the Mexican National Polytechnic Institute (IPN).

ABSTRACT Time-Stable Geocast is a group communication pattern where the destination is composed of a set of nodes that, at any time during a predefined time interval, have been located within a given geographic destination region. Time-Stable Geocast is well suited to supporting Smart City and Intelligent Transportation Systems (ITS) applications such as road safety, crowdsensing, entertainment, and business applications that can benefit from disseminating position and time-based abiding information. In this paper, we present GeoTemporal-cast, an opportunistic time-stable geocast routing protocol for Delay-Tolerant Vehicular Ad Hoc Networks (DT-VANETs) that, unlike previous proposals, does not assume that either data messages are generated inside the geographic destination region or that an underlying routing protocol will deliver the packets there. GeoTemporal-cast computes shortest-path trees over the street-layout graph that are rooted at a connected component that includes all the nodes representing intersections located inside the destination region. By traveling along the edges of these trees, data messages can reach the destination region from all the streets with access to the destination region. We evaluate the performance of GeoTemporal-cast using detailed simulations in NS-3 that consider city-wide realistic scenarios involving up to 9,080 vehicles. Our results show that GeoTemporal-cast outperforms a set of time-stable protocols for Delay-Tolerant Networks, including Time-Stable Geographically-Restricted Epidemic and variations of Spray & Wait, in terms of delivery ratio, delivery delay, and overhead.

INDEX TERMS Routing, VANET, DTN, time-stable geocast, smart city, intelligent transportation system.

I. INTRODUCTION

Delay-Tolerant Vehicular Ad-hoc Networks (DT-VANETs) is a technology that integrates the capabilities of ad hoc wireless networks into vehicles to provide ubiquitous connectivity to mobile users while on the road, even if there are no contemporaneous end-to-end paths connecting data sources to destinations. DT-VANETs can incorporate Vehicle-to-Vehicle (V2V), Vehicle-to-Roadside (V2R) and Vehicle-to-Infrastructure (V2I) communications to support a growing number of Intelligent Transportation Systems (ITS) [1], [2] and Smart City applications [3].

DT-VANETs and regular Vehicular Ad-hoc Networks (VANETs) belong to a subcategory of Mobile Ad-hoc Networks (MANETs) because they share similar general features including self-organization, self-management, and restricted access to bandwidth. However, VANETs and DT-VANETs experience unique conditions such as frequent topological changes, large variations in the speeds of nodes, and restricted mobility patterns imposed by the streets and roads [4], [5]. VANETs and DT-VANETS can also be distinguished from other forms of Mobile Ad-hoc Networks because their nodes have less stringent constraints in terms of energy consumption, storage, and computing power. Additionally, VANET and DT-VANET nodes usually have access to a variety of sensors that produce information that can be used in the routing process, for example, a Global Positioning System (GPS) receiver that provides vehicles with their geographical position [6]. DT-VANETs are further characterized by low node densities and the fact that they do not assume the existence of contemporary end-to-end paths from sources to destinations.

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

In the context of Smart Cities and ITS, DT-VANETs can help offloading delay-tolerant traffic from the infrastructure-based networks but perhaps more importantly, serve as the communication platform connecting a swarm composed of vehicles, personal mobile devices, Internet of Things (IoT) devices installed at buildings and roadside furniture, and the Cloud [7]. This, to support a growing number of position- and time-based cooperative applications such as Amber Alerts [8], crowdsensing applications (e.g. [9]–[17]), virtual warning signs [18], and traffic management applications [12], [19].

Traditionally, this type of application employ geocast routing protocols for VANETs (e.g., [20]–[22]) to deliver information to groups of nodes located at a given geographic destination region (also referred as the Region of Interest). Unfortunately, traditional geocast communication does not consider the temporal scope of data, and hence, data is delivered to the nodes that happen to be located at the region of interest (RoI) at the time the packets arrive. This is particularly disadvantageous for delay-tolerant communications because it can be the case that at that time, no node is inside the RoI. Moreover, when the network is intermittently connected, the set of nodes that compose the destination is not well specified because the time in which the packet will arrive at the RoI is highly unpredictable. Time-Stable Geocast (also referred to as abiding geocast) solves this problem by explicitly defining the data messages' temporal scope in terms of a time interval (t_0, t_1) during which a data packet has to be delivered to any node inside the RoI.

As concrete examples of applications of the time-stable geocast communication pattern, we identify the following motivating scenarios:

- Amber Alerts. An Amber Alert is sent to the devices in the vicinity of the last place a missing child was seen. The data messages that carry information about the alert remain active around the place during the alert's lifetime, even if there are neither intelligent roadside furniture nor vehicles in the vicinity for an arbitrarily long time. This way, during the time defined in the alert, any device entering the geographic region will receive the alert.
- Delivery of sensing requests in crowdsensing platforms. A crowdsensing application uses sensing request messages to recruit vehicles and pedestrians to monitor a region of interest during a predefined time interval. The request is retained in the vicinity of the area of interest during the request's lifetime, even if there are no nodes nearby for arbitrarily long times.
- Delivery of requests for data muling. Similarly to the crowdsensing application, a data collection application can recruit vehicles to act as data mules for the information collected by sensors located in an area of interest. This functionality is well suited to collect data from cheap sensors with no direct connection to the Internet.
- Virtual signs. Applications can temporarily post information to specific regions of the City. For instance, a data analytics application running on the Cloud can post findings obtained from analyzing crowd-sensed data to specific geographic areas of a City. This information remains posted for as long as it is relevant. For instance, an application can alert drivers about a predicted surge in ground-level pollution that will take place between 5:00 and 7:00 PM. Drivers can also post virtual signs, for instance, to notify other drivers about road hazards.

In this paper, we present GeoTemporal-cast, an opportunistic time-stable geocast routing protocol that, unlike previous proposals, does not assume that either data messages are generated inside the geographic destination region (e.g., [8], [19], [23]–[29]) or that an underlying routing protocol will deliver the packets to the destination region (e.g., [18], [30]). Section II presents a more detailed analysis of these proposals.

The proposed protocol takes advantage of the information available to current navigation systems to compute shortest-path trees over the street-layout graph. These trees are rooted at a connected component that includes all the nodes representing intersections located inside the destination region. By traveling along the edges of these trees, data messages can reach the destination region from all the streets with access to the region. Moreover, in situations where no vehicles are inside the destination region, GeoTemporal-cast will use these trees to keep messages as close as possible to that geographic region.

GeoTemporal-cast disseminates data messages towards the destination regions according to the packets' priorities, which can be either high or regular. Nodes use a priority queue that also considers the distance to the destination region in the street-layout graph, the traveling direction of the vehicles, and the number of hops that packets have traversed so far. When two vehicles meet, high-priority packets are exchanged before regular priority packets. Low-priority packets are also discarded first when the data queues get full. The objective is to maximize the delivery probability while minimizing the delay with which high-priority packets are delivered to their intended receivers.

Our experimental results, based on detailed simulations in NS-3 that consider real city layouts and real mobility traces involving up to 9,080 vehicles, show that GeoTemporal-cast performs well in realistic scenarios. It attains better delivery ratio and delivery delay while inducing far less overhead than a group of time-stable protocols for Delay Tolerant Networks, including geographically-restricted epidemic and variations of Spray & Wait. To the best of our knowledge, this is the largest-scale simulation-based study for time-stable geocast routing protocols presented to date.

The rest of this paper is organized as follows. Section II presents a sample of the work devoted to time-stable geocasting. Section III presents the system model and the formulation of the problem of routing time-stable traffic in Delay-Tolerant VANETs. Section IV presents the proposed protocol and related algorithms as well as a series of theorems that characterize their temporal complexity. Section V presents the results of our simulation experiments and Section VI presents our concluding remarks.

II. RELATED WORK

In this section, we present a sample of the work devoted to designing time-stable (also referred to as abiding) geocast routing protocols. This analysis reveals that previous work assumes either that the data messages are generated inside the geographic destination region (see Section II-A) or that there is an underlying geocast routing protocol that delivers the packets to the destination region (see Section II-B). Both assumptions limit the applicability of these protocols to very specific scenarios.

Based on these assumptions, previous proposals focus only on retaining the time-stable packets within their destination region during their predefined lifetime.

A. PROTOCOLS THAT RETAIN PACKETS WITHIN THE DESTINATION REGION

In [24], the authors present a protocol based on the IEEE 802.11 technology that delivers abiding messages to nodes located within a destination region. Nodes carrying an abiding message, dynamically define a forwarding region that contains the destination region and whose size depends on the perceived vehicle density. The objective of having an extended forwarding region is to improve the probability of nodes receiving the packet before entering the destination region. For each destination region, the protocol sets a unique Service Set Identifier (SSID) and nodes inside the destination region form an Independent Basic Service Set (IBSS). When a node in the forwarding region detects a new node within its communication range, it relays the message. If the receiver node does not already possess the message, it checks if it is inside the destination region, and if so, it passes the packet to upper layers and joins the IBSS. Nodes keep abiding data messages for as long as they remain in the forwarding region and while the lifetime of the packets has not expired.

In [25], the authors propose a timer-based protocol designed to disseminate and retain safety information along a bidirectional stretch of road. When a vehicle detects an emergency, it starts broadcasting a warning message to inform other vehicles about the event. Upon receiving a warning message, vehicles become active relays responsible for relaying the warning message to vehicles traveling in the opposite direction. The protocol also defines a forwarding region where the warning message is disseminated. The computation of this forwarding region assumes that vehicles enter the road from both ends following a Poisson distribution. When nodes receive a message from a vehicle traveling in the same direction, they wait for a dynamically computed time before the next broadcast. This time depends on the vehicles' direction and speed, their relative positions, and the distance to the boundaries of the forwarding region.

The Dynamic Time Stable Geocast (DTSG) [26] protocol is also a timer-based protocol designed for two-way highways. As in the previous approaches, the protocol defines an extended forwarding region that covers the original destination region where vehicles disseminate time-stable messages. The dynamic behavior of the protocol allows reducing, extending, or canceling the packets' lifespan. The protocol has two phases, the pre-stable phase, which ends when the message has been disseminated along the destination region, and the stable phase, which is in charge of retaining the message during its lifespan. In the pre-stable phase, the source vehicle generates and broadcasts a message upon encountering an event. The source vehicle periodically broadcasts the message until at least one helping vehicle moving in the opposite direction receives the message. Each time a vehicle receives a message, it immediately broadcasts the message, thus acknowledging the reception. The helping vehicle periodically broadcasts the packet until it reaches the end of the destination region, where a *extra region* begins. At this moment, it changes the status flag of the message to the stable phase. Later, it starts broadcasting the message again, until at least one vehicle moving towards the event receives it. Then, this receiver vehicle stops broadcasting the message until it reaches the extra region. In this region, the vehicle periodically broadcasts the message until it receives an acknowledgment. This cycle continues until the message expires or is canceled. The Traffic light Time-Stable Geocast (T-TSG) [28] protocol is an extension to DTSG for urban environments that takes into account the state of the traffic lights when selecting the forwarding vehicles.

The Abiding Geocast for Warning Message Dissemination (AGWMD) [27] protocol is a timer-based protocol that assumes that a roadside unit (SRU) initiates the dissemination of a warning message. The process starts when the RSU transmits an advertisement (ADV) message, which includes the next two forwarders' identity. When a selected vehicle receives a message transmitted by the RSU, it waits for a time proportional to its current distance to the center of the destination region before forwarding the message. This message also contains the list of the next forwarder vehicles selected by the current forwarder. If the current forwarder receives another broadcast from a vehicle traveling ahead in the same direction before the timer expires, it cancels its timer. This process is repeated until the message reaches the destination region's boundary and for as long as the message is still active.

The Semantic and Self-Decision Geocast Protocol (SAS-GP) [8] is also a timer-based protocol where vehicles disseminate a warning message when encountering an accident, to inform other vehicles located in the area affected by the event. This area, referred to as the Semantic Geocast Domain (SGD), includes all the streets that lead to the location of the event, and its size depends on the current traffic information. SAS-GP operates in three phases, namely, the spread, preserve, and assurance phases. During the spread phase, the packet is disseminated up to the borders of the SGD. Then, in the preserve phase, the message is retained in the SGD during the message lifetime. The assurance phase's objective is to guarantee that the message is delivered to all the vehicles inside the SDG. The abiding geocast protocol based on carrier sets (AG-CS) [29] divides the destination area into street segments covered by a set of carrier vehicles. The vehicles in a carrier set store and periodically retransmit abiding messages. These vehicles are selected by other carrier vehicles based on the stability estimation index, which is a function of the link's duration and connectivity. In [19], the authors propose a probabilistic timer-based approach to retain packets within the destination region by periodically retransmitting data messages. Time is divided into time frames, and at the beginning of each frame, the protocol randomly determines the next transmission time within the frame. The protocol adapts the transmission probabilities based on the density of neighboring vehicles. It controls the data transmission period according to the vehicle's degree of contribution to the coverage, which is proportional to the size of the area where the communication range of the vehicle does not overlap with that of its nearest neighbor.

B. PROPOSALS THAT ASSUME AN UNDERLYING GEOCAST ROUTING PROTOCOL

In [18], [23], Maihöfer *et al.* propose three approaches for implementing abiding geocasting, namely, server-based, election-based, and neighbor-based approaches. In the server-based approach, packets are unicasted to a geocast server responsible for periodically flooding the destination region. Alternatively, the geocast server can transmit packets on-demand to respond to requests transmitted by nodes within the destination area. The election-based approach assumes that packets are delivered to the destination area by an underlying geocast routing protocol. A dynamically elected node within the destination region stores and delivers the message to other nodes in the destination region either on-demand or periodically with one-hop broadcasts. The neighbor-based approach is similar to the previous approach, but in this case, every node within the destination area stores and delivers packets to neighbors located within the destination region.

The Opportunistic Spatio-Temporal Dissemination (OSTD) [30] protocol disseminates event information in a destination geographic region during the lifetime of the event. OSTD follows a published/subscriber approach where events are delivered only to vehicles that have subscribed to the event's topic. Similar to previous approaches, OSTD assumes that either the packet is generated inside the destination region or that there is an underlying geocast routing protocol for DT-VANETs (e.g., GeOpps [31]). OSTD keeps multiple replicas of the messages at designated vehicles, which periodically broadcast the message to cover the areas where intended receivers are likely to be. When a vehicle carrying a replica moves outside of the destination region, it has to transfer the replica to a vehicle moving back inside.

III. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we present the notation used throughout the paper and a precise formulation of the problem of routing time-stable geocast traffic in Delay-Tolerant VANETs.

Let $G_{STREET} = (I, S)$ be a weighted directed graph that describes the street layout of a geographic region, where I is the set of street intersections, and S is the set of road segments connecting the intersections. An edge $(i_u, i_v) \in S$ indicates that there is a road segment that vehicles can use to get to intersection i_v from intersection i_u. The weights of the edges are determined by a function $l : S \to \mathbb{R}^+$ based on the length of the corresponding street segments. We use $p(i_u)$ to denote the position of intersection i_u, and $x_{p(i_u)}$ and $y_{p(i_u)}$ to denote the x and y coordinates of $p(i_u)$.

Let $G_{VANET}(t) = (V(t), E(t))$ be a time-varying directed graph that represents the topology of the VANET at time t, where $V(t)$ is a dynamic, time-dependent set of vehicles, and E is also a dynamic, time-dependent set of wireless links that connect two vehicles $v_u, v_v \in V(t)$ at time t if v_v can decode a packet received from from v_u at time t. We say that there is a *contact opportunity* $CO_{\{v_u,v_v\}}$ between vehicles v_u and v_v during time interval $(t0_{\{v_u,v_v\}}, t1_{\{v_u,v_v\}})$, if $\forall t \in (t0_{\{v_u,v_v\}}, t1_{\{v_u,v_v\}})$ exist $e_{uv} = (v_u, v_v)$ and $e_{vu} = (v_v, v_u)$ in $E(t)$. During the lifetime of the contact opportunity, vehicles v_u and v_v can exchange either control or data packets.

We assume that vehicles are equipped with an onboard Navigation System with a GPS receiver. We further assume that an instance of the protocol running on vehicle $v_u \in V(t)$ can access the graph G_{STREET} and the current geographic position of v_u denoted by $p(v_u, t)$. We use $x_{p(v_u,t)}$ and $y_{p(v_u,t)}$ to denote the x and y coordinates of the position of node v_u at time t.

A time-stable geocast data flow generated by source vehicle $v_s \in V(t)$, denoted by f_{v_s}, is a sequence of $n = |f_{v_s}|$ data messages $\langle m_1, m_2, \ldots, m_n\rangle$. Each data message m is $|m|$ bytes long, and contains the following information:

- A unique identifier composed of the source address and a sequence number.
- A rectangular geographic destination region, denoted by $DR(m) = \{(x0_{(m)}, y0_{(m)}), (x1_{(m)}, y1_{(m)})\}$, where $(x0_{(m)}, y0_{(m)})$ and $(x1_{(m)}, y1_{(m)})$ are the coordinates of its south-west and north-east corners, respectively.
- The packet's temporal scope, denoted by $\tau(m) = (t0_{(m)}, t1_{(m)})$, which is defined by the time interval between $t0_{(m)}$ and $t1_{(m)}$.
- A priority denoted by $PRIO(m)$, which can be either *HIGH* or *REGULAR*.
- The number of hops m has traversed so far, denoted by $HOP(m)$.
- The payload denoted by $P(m)$.

We use M to denote the set of all time-stable geocast data messages generated by all the sources in the network.

We define the *set of intended receivers of a time-stable geocast message m*, as $IR_m = \{v \in V(t) : x0_{(m)} \leq x_{p(v,t)} \leq x1_{(m)} \wedge y0_{(m)} \leq y_{p(v,t)} \leq y1_{(m)} \wedge t0_{(m)} \leq t \leq t1_{(m)}\}$. We use $p(v_u, t) \in$

$DR(m)$ to denote that vehicle v_u is within the destination region of message m at time t. We also define the indicator function $\mathbf{I} : M \times V(t) \rightarrow \{0, 1\}$ that equals 1, if and only if message $m \in M$ is received by intended receiver $v_u \in IR_m$ at time t, such that $t_{0_{(m)}} \leq t \leq t_{1_{(m)}}$. Lastly, the delivery delay of message m, experienced by an intended receiver v_u, denoted by $\delta(m, v_u)$, is computed as the time elapsed between the arrival of v_u at the destination region $DR(m)$ and the time in which message m was received by v_u.

From the previous definitions, we can formulate the *problem of time-stable geocast routing in DT-VANETs* in terms of the messages that are transmitted by the vehicles during their contact opportunities. The problem consist in, given a DT-VANET G_{VANET} and a set of N data flows $\{f^1, f^2, \ldots, f^N\}$, determine a sequence $Tx_{CO_{\{v_u,v_v\}}} = \langle m_1, m_2, \ldots, m_l \rangle$ of data packet transmissions that occur in each contact opportunity $CO_{\{v_u,v_v\}}$ in G_{VANET}, such that:

- The packet delivery ratio (DR), as defined in (1), is maximized.

$$DR = \frac{\sum_{i=1}^{N} \sum_{m \in f^i} \sum_{v_u \in IR_m} I(m, v_u)}{\sum_{i=1}^{N} \sum_{m_j \in f^i} |IR_m|} \quad (1)$$

- The average delivery delay (D), as defined in (2), is minimized.

$$DR = \frac{\sum_{i=1}^{N} \sum_{m \in f^i} \sum_{v_u \in IR_m} \delta(m, v_u)}{\sum_{i=1}^{N} \sum_{m \in f^i} \sum_{v_u \in IR_m} \mathbf{I}(m, v_u)} \quad (2)$$

- The data overhead (DO), as defined in (3), is minimized.

$$DO = \frac{\sum_{CO_{\{v_u,v_v\}} \text{ in } G_{VANET}} \sum_{m \in Tx_{CO_{\{v_u,v_v\}}}} |m|}{\sum_{i=1}^{N} \sum_{m \in f^i} \sum_{v_u \in IR_m} \mathbf{I}(m, v_u)|m|} \quad (3)$$

By using the indicator function $\mathbf{I}$, the numerator of (1) only considers the messages m that were actually received by an m's intended receiver $v_u \in IR_m$ during m's temporal scope. The denominator of (1) is the sum of the cardinalities of the sets of intended receivers IR_m of all the messages that compose the set $\{f^1, \ldots f^N\}$ of data flows. Similarly, (2) employs function δ to compute the sum of the delivery delays experienced by all the messages that were received by an intended receiver. Lastly, the data overhead is computed in (3) as the ratio between the sum of the length in bytes of all the messages that were exchanged during all the contact opportunities $CO_{\{v_u,v_v\}}$ in G_{VANET}, and the length of the data packets that were received by the intended receivers.

Table 1 summarizes the notation introduced in this section.

TABLE 1. Summary of notation.

G_{STREET}	Graph representing the street layout
I	Set of street intersections
S	Set of road segments
$l((i_u, i_v))$	Length of the road segment connecting intersections i_u and i_v.
$G_{VANET}(t)$	Time-varying graph representing the topology of the DT-VANET
$V(t)$	Time-varying set of vehicles
$E(t)$	Time-varying set of wireless links
$p(i_u)$	Position of intersection i_u
$p(v_u, t)$	Position of vehicle v_u at time t
f_{v_s}	Data flow generated by source v_s
$DR(m)$	Destination geographic region of packet m
$\tau(m) = (t_{0_{(m)}}, t_{1_{(m)}})$	Temporal scope of packet m
M	Set of all the time-stable geocast messages in an experiment
IR_m	Set of intended receivers of message m
$CO_{\{v_u,v_v\}}$	Contact opportunity between vehicles v_u and v_v
$Tx_{CO_{\{v_u,v_v\}}}$	Set of messages transmitted during a contact opportunity between vehicles v_u and v_v
$\mathbf{I}(m, v_u)$	Indicator function that indicates the correct reception of message m by receiver v_u
$\delta(m, v_u)$	Delivery delay of message m, experienced by an intended receiver v_u

IV. THE GEOTEMPORAL-CAST PROTOCOL

A. OVERVIEW

GeoTemporal-cast is an opportunistic time-stable geocast protocol that employs contact opportunities between vehicles to disseminate time-stable geocast messages towards their corresponding geographic destination regions and then retain the messages there during their lifetime. Vehicles use small `HELLO` packets to detect contact opportunities during which data messages are exchanged.

Vehicles route a time-stable geocast message m towards its destination region $DR(m)$ following a shortest-path tree $T_{DR(m)} \subseteq G_{STREET} = (I, S)$ that is rooted at a connected component that contains all the intersections $i_u \in I$ such that $x_{0_{(m)}} \leq x_{p(i_u)} \leq x_{1_{(m)}} \wedge y_{0_{(m)}} \leq y_{p(i_u)} \leq y_{1_{(m)}}$, namely, all the intersections inside m's destination geographic region. By traveling along the edges of $T_{DR(m)}$, a copy of message m can reach the $DR(m)$ from any of the streets with access to that geographic region (see Fig. 2). Moreover, during m's lifetime $\tau(m) = (t_{0_{(m)}}, t_{1_{(m)}})$, $T_{DR(m)}$ will keep *attracting* the replicas of m towards the streets with access to $DR(m)$. Once a replica of m reaches the *extended destination region* $DR'(m)$ that contains the connected component and the original destination region, GeoTemporal-cast operates in Epidemic mode [32] to inform all the vehicles about the packet. Please note that all the vehicles in $DR(m)$ during $\tau(m)$ are intended receivers of message m.

During a contact opportunity, GeoTemporal-cast employs a priority-based scheme to compute the messages' *forward priority* that is used to select which messages to exchange first. We define a total order relation over the packets that considers the packet type (high-priority packets are preferred over regular-priority packets), the positions of the vehicles (packets already in the extended destination region are preferred over packets traveling towards the region), the positions and traveling directions of the sender and receiver vehicles, and the number of hops that packets have traversed so far. This same ordering relation is used to discard packets when the data queue of a node gets full. Data messages are also discarded when their lifetime ends. Vehicles use the *Viable Carrier Condition* to forward data messages only to *viable carriers*, which are neighboring vehicles that make progress in $T_{DR(m)}$ towards the extended destination region.

TABLE 2. Summary of notation used in the protocol and algorithms specifications.

$T_{DR(m)}$	Shortest-path tree rooted at super node i_{super}
$DR'(m)$	Extended destination region of packet m
SV_{v_u}	Summary vector of vehicle v_u
DV_{v_u}	Disjoint vector of vehicle v_u
I_{SUPER}	Connected component of G_{STREET} that defines the extended destination region
i_{super}	A node that replaces I_{SUPER} in $T_{DR(m)}$

Lastly, in order to limit the amount of resources assigned to individual messages, GeoTemporal-cast restricts the number of replicas of regular-priority messages that vehicles can forward when they are located outside of the extended destination region.

Table 2 summarizes the notation used in this section.

B. INFORMATION EXCHANGED

In this section, we present the protocol implemented by GeoTemporal-cast to detect contact opportunities and exchange data messages. Our presentation is succinct because this message exchange is similar to the one used by Epidemic [32]. The main difference resides in how nodes compute the data messages' forward priority, and the use of the *Viable Carrier Condition* to determine if a given vehicle is a suitable next hop towards the destination region.

GeoTemporal-cast defines a DATA packet and five control packets.

- HELLO: Announces the presence of a vehicle to surrounding vehicles.
- REPLY: Response to a HELLO. It includes a *Summary Vector* with the identifiers of the DATA packets this vehicle *offers*.
- REPLY_BACK: Response to a REPLY. It includes a *Disjoint Vector* with the identifiers of the DATA packets that the node requests.
- ACK: Transmitted to *acknowledge* a DATA packet.
- DATA_ACK: A DATA packet piggybacked in an ACK packet.

All these packets contain the current geographical information (position and traveling direction) of the vehicle that transmitted them. The routing protocol uses this information to determine if a node is a *viable carrier* for the DATA packets. We describe this process in Section IV-C.

Fig. 1 illustrates the message exchange during a contact opportunity. When two vehicles, v_u and v_v, come into communication range of one another, an anti-entropy session initiates (Step 1). Nodes ignore HELLO packets broadcasted by nodes with a smaller id. In this example, the node with a larger identifier (node v_v) ignores a HELLO packet broadcasted by a node with a smaller identifier (node v_u). Then node v_v broadcasts a HELLO packet that is not ignored and is processed to initiate the exchange of control packets.

When a HELLO packet is processed, the protocol inserts the identifier of the node that broadcasted the packet into a

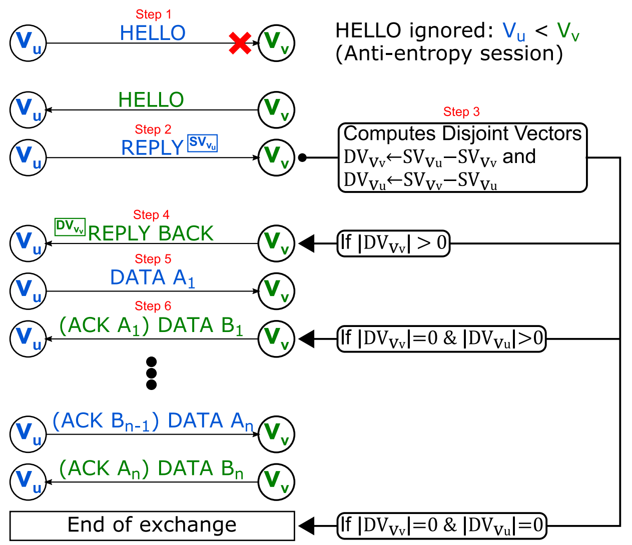

FIGURE 1. Sequence of packets exchanged during a contact opportunity between vehicles v_u and v_v. SV_{v_u} and SV_{v_v} are Summary Vectors, and DV_{v_u} and DV_{v_v} are Disjoint Vectors of nodes v_u and v_v, respectively.

cache of recently contacted neighbor nodes. Nodes included in this list are not contacted again within a configurable amount of time to avoid frequent redundant connections. In Step 2, as a response to the HELLO packet from node v_v, node v_u sends a REPLY packet with its *Summary Vector* SV_{v_u} that contains the identifiers of the DATA packets that v_u has in memory. In Step 3, node v_v uses the Summary Vector SV_{v_u} and its own Summary Vector SV_{v_v} to compute *Disjoint Vectors* $DV_{v_v} \leftarrow SV_{v_u} \setminus SV_{v_v}$ and $DV_{v_u} \leftarrow SV_{v_v} \setminus SV_{v_u}$. Node v_v also caches the Disjoint Vector of the other node (DV_{v_u}) and uses the cardinality of both disjoint vectors to decide which packets to exchange. In Step 4, node v_v sends a REPLY_BACK packet that includes its Disjoint Vector DV_{v_v} to node v_u. Node v_u caches the received Disjoint Vector in memory.

In Step 5, the sender node v_u uses the time- and position-dependent total order relation defined in (4) to establish an ordering over the packets $m \in DV_{v_v}$ and to decide which message is going to send first. As it can be seen in (4), GeoTemporal-cast prioritizes high-priority packets ($PRIO(m) = HIGH$), then packets which are currently in their destination region ($p(v_u, t) \in DR(m)$), then packets with lower hop count ($HOP(m)$), and then packets that have been reported by a fewer number of neighboring vehicles. Ties are broken based on the IP address of the source vehicle. The aim of this total order relation is to assign more network and computing resources to high-priority messages, and then to messages that are about to be delivered to intended receivers. It also promotes the dissemination of messages that have reached a small number of vehicles.

As it is described in Section IV-C, nodes always use the *Viable Carrier Condition* to check if the receiver node v_v is a *viable carrier* for the message before actually transmitting it. v_u also checks if the counter of available replicas of the selected packet is larger than zero, and if so, it decrements the counter in one unit and transmits the packet.

When node v_v receives the transmitted DATA packet replica, it increments the hops count of the packet by one, set its number of available replicas to the maximum value, and stores the packet in memory. This completes Step 5.

$$
\begin{aligned}
& m_i <_{v_u} m_j \\
& \quad \textbf{iff } PRIO(m_i) < PRIO(m_j) \textbf{ or} \\
& \qquad PRIO(m_i) = PRIO(m_j) \wedge p(v_u, t) \\
& \qquad\qquad \in DR(m_j) \wedge p(v_u, t) \notin DR(m_i) \textbf{ or} \\
& \qquad PRIO(m_i) = PRIO(m_j) \wedge p(v_u, t) \\
& \qquad\qquad \in DR(m_j) \wedge p(v_u, t) \in DR(m_i) \wedge \\
& \qquad HOP(m_i) > HOP(m_j) \textbf{ or} \\
& \qquad PRIO(m_i) = PRIO(m_j) \wedge p(v_u, t) \\
& \qquad\qquad \in DR(m_j) \wedge p(v_u, t) \in DR(m_i) \wedge \\
& \qquad HOP(m_i) = HOP(m_j) \wedge COPY(m_i) > COPY(m_j) \textbf{ or} \\
& \qquad PRIO(m_i) = PRIO(m_j) \wedge p(v_u, t) \\
& \qquad\qquad \in DR(m_j) \wedge p(v_u, t) \in DR(m_i) \wedge \\
& \qquad HOP(m_i) = HOP(m_j) \wedge COPY(m_i) \\
& \qquad\qquad = COPY(m_j) \wedge ID(m_i) < ID(m_j) \\
& \qquad PRIO(m_i) = PRIO(m_j) \wedge p(v_u, t) \\
& \qquad\qquad \notin DR(m_j) \wedge p(v_u, t) \notin DR(m_i) \wedge \\
& \qquad HOP(m_i) > HOP(m_j) \textbf{ or} \\
& \qquad PRIO(m_i) = PRIO(m_j) \wedge p(v_u, t) \\
& \qquad\qquad \notin DR(m_j) \wedge p(v_u, t) \notin DR(m_i) \wedge \\
& \qquad HOP(m_i) = HOP(m_j) \wedge COPY(m_i) > COPY(m_j) \textbf{ or} \\
& \qquad PRIO(m_i) = PRIO(m_j) \wedge p(v_u, t) \\
& \qquad\qquad \notin DR(m_j) \wedge p(v_u, t) \notin DR(m_i) \wedge \\
& \qquad HOP(m_i) = HOP(m_j) \wedge COPY(m_i) \\
& \qquad\qquad = COPY(m_j) \wedge ID(m_i) < ID(m_j)
\end{aligned} \tag{4}
$$

Afterward, in Step 6, the receiver node v_v transmits back either an ACK packet that contains the identifier of the DATA packet being acknowledged or a DATA_ACK packet that also contains a DATA piggybacked. Nodes use DATA_ACK packets when they also have packets to send, namely, when $|DV_{v_u}| > 0$. Nodes select the packet to be transmitted in a DATA_ACK packets in the same way as described in Step 5. Lastly, nodes repeat Step 6 until they have exchanged all the packets in DV_{v_v} and DV_{v_u}, or the contact opportunity ends. These two nodes do not initiate a new conversation with each other until the *recently contacted neighbor* timer expires.

C. VIABLE CARRIER CONDITION

Before relaying a message m to a requesting vehicle, nodes use Algorithm 1 to determine if that vehicle is a viable data-carrier for m. Algorithm 1 receives as input an extended destination area $DR'(m)$ that contains the original destination region $DR(m)$, a shortest-path tree $T_{DR(m)}$ rooted at the extended destination region $DR'(m)$, the current data-carrier node v_u, and the candidate data-carrier node v_v. Section IV-D presents the details on how the shortest-path tree $T_{DR(m)}$ and the extended destination region $DR'(m)$ are computed.

Algorithm 1 is based on the *Viable Carrier Condition* that states that a vehicle is a *Viable Carrier* if it meets any of the following criteria:

(*i*) Any of the two nodes is inside the *extended destination region* (Line 2). This is because nodes operate in Epidemic mode when they are inside this region.

(*ii*) The candidate data-carrier node v_v is closer, in $T_{DR(m)}$, to the extended destination region $DR'(m)$ than the current carrier v_u (Line 5), and v_v is not moving away from the extended destination region (Line 8). These distances (Line 5) are computed over the three $T_{DR(m)}$ to consider the actual distance that a vehicle should travel to get to the extended destination region.

(*iii*) The candidate data-carrier node v_v is farther away, in $T_{DR(m)}$, to the extended destination region $DR'(m)$ than the current carrier v_u (Line 6) but v_v is moving towards the destination region and v_u is moving away from the destination region (Line 10).

Algorithm 1 Determines if a Node is a Valid Packet Carrier

Input: Shortest-path tree $T_{DR(m)}$, current data-carrier node v_u, candidate data-carrier node v_v, extended destination area $DR'(m)$

Output: Returns `True` if node v_v is a valid packet carrier, `False` otherwise.

1 **Function** `IsValidPacketCarrier` (v_u, v_v, $DR'(m)$, $T_{DR(m)}$) **:**
2   **if** $p(v_u, t) \in DR'(m)$ **or** $p(v_v, t) \in DR'(m)$ **then**
3   **return** `True`
4   **end if**
5   **if** `distance`$(v_v, T_{DR(m)}) \leq$ `distance`$(v_v, T_{DR(m)})$ **then** closer ← `True`
6   **else** closer ← `False`
7   **if** *closer* **is** `True` **then**
8   **if** v_v is moving away from $DR'(m)$ in $T_{DR(m)}$ **and** v_u is moving towards $DR'(m)$ in $T_{DR(m)}$ **then** **return** `False`
9   **else**
10   **if** v_v is moving towards $DR'(m)$ *in* $T_{DR(m)}$ **and** v_u is moving away from $DR'(m)$ *in* $T_{DR(m)}$ **then** **return** `True`
11   **return** `False`
12   **end if**
13   **return** `True`
14 **end**

By selecting carriers according to (*ii*) and (*iii*) of the *Viable Carrier Condition*, GeoTemporal-cast relays DATA packets from sources to their corresponding destination regions by selecting vehicles that are closer to the extended destination region in terms of the actual distance that a vehicle has to travel to get there. On the other hand, the purpose of (*i*) is twofold. To disseminate replicas of the packet along the destination region and then retain those messages there during the packets' lifetime. In the case that no vehicle is inside the destination region, (*ii*) and (*iii*) will continuously attract the message replicas to the destination region because vehicles will keep forwarding the replicas to vehicles closer to that region.

D. EXTENDED DESTINATION REGION AND SHORTEST-PATH TREE COMPUTATION

GeoTemporal-cast uses Algorithm 2 to compute m's extended destination region $DR'(m)$, which is the minimum-size rectangular geographic region (see Lines 15–19) that contains: (*i*) m's original geographic destination region $DR(m)$, and (*ii*) a connected component $I_{SUPER} \subseteq I$ that is composed of the set of intersections $I_{IN} \subseteq I$ located inside the $DR(m)$ (see Lines 4–8) and a set of intersections laying in shortest-paths connecting them (see Lines 11–14). The purpose of having an extended destination region is to define a geographic region that includes streets segments connecting all the reachable points within the destination region $DR(m)$. This way, vehicles operating in Epidemic mode can disseminate data packet across the whole destination region $DR(m)$.

In order to compute the shortest-path tree $T_{DR(m)}$, Algorithm 2 operates over an undirected version G'_{STREET} of the street-layout graph G_{STREET}. This is because data messages, unlike vehicles, can travel in any of the directions of a street segment. Algorithm 2 replaces the connected component I_{SUPER} with a super node i_{super} where i_{super} is connected to all the intersections i_u such that there is street segment $(i_u, i_v) \in S$ with $i_u \notin I_{SUPER}$ and $i_v \in I_{SUPER}$ (see Lines 25 and 28). Lastly, Algorithm 2 simply employs Dijkstra's algorithm to compute a shortest path tree rooted at i_{super}.

Figures 2(a-d) illustrate the previous concepts. Fig. 2(a) shows, with black dashed lines, the geographic destination region of a packet m that contains intersections i_u, i_v and i_w. Fig. 2(b) shows in red the intersections contained in the connected component I_{SUPER} that includes i_u, i_v and i_w, as well as other intersection needed to keep them connected. Fig. 2(b) also shows the extended destination region $DR'(m)$ computed by Algorithm 2 that contains the intersections in I_{SUPER} and the original $DR(m)$. In Fig. 2(c), the connected component I_{SUPER} is substituted by a single supernode i_{super}. Lastly, Fig. 2(d) shows the resulting directed shortest-path tree where the arrows indicate the direction of the shortest-paths towards the extended destination region. In this last figure, we use red arrows to highlight that all the street segments with access to the extended destination region are directly connected to the supernode.

E. DATA PACKET REPLICAS

As a way to limit the amount of network and computing resources assigned to regular-priority time-stable geocast messages, GeoTemporal-cast nodes have a configurable maximum number of L message replicas that they can forward to vehicles located outside of the extended destination region. This limitation does not apply to packets inside the destination region because all the vehicles there are intended receivers.

GeoTemporal-cast does no limit the number of replicas of high-priority messages because this type of packet is typically used to support emergency-like applications.

Algorithm 2 Computes a Shortest-Path Tree Rooted at the Destination Region of Message m

Input: Street-layout graph $G_{STREET} = (I, S)$ and the destination region $DR(m)$ of packet m.

1 **Function** `StreetTree` (G_{STREET}, $DR(m)$) **:**
2 $G'_{STREET} = (I, S') \leftarrow$ undirected version of G_{STREET}
3 $I_{IN} \leftarrow \emptyset$
4 **foreach** *intersection* $i_u \in I$ **do**
5 **if** $p(i_u) \in DR(m)$ **then**
6 $I_{IN} \leftarrow I_{IN} \cup \{i_u\}$
7 **end if**
8 **end foreach**
9 $i_{centroid} \leftarrow i_u \in I_{IN}$ such that it is the closest to the centroid of $DR(m)$
10 $I_{SUPER} \leftarrow I_{IN}$
11 **foreach** *intersection* $i_u \in I_{IN} - \{i_{centroid}\}$ **do**
12 $P_{\{i_{centroid}, i_u\}} \leftarrow$ shortest path from $i_{centroid}$ to i_u in G'_{STREET}
13 $I_{SUPER} \leftarrow I_{SUPER} \cup P_{\{i_{centroid}, i_u\}}$
14 **end foreach**
15 $x_{min} \leftarrow \min\{x_{0_{(m)}}, \min_{i_u \in I_{SUPER}} \{x_{p(i_u)}\}\}$
16 $x_{max} \leftarrow \max\{x_{1_{(m)}}, \max_{i_u \in I_{SUPER}} \{x_{p(i_u)}\}\}$
17 $y_{min} \leftarrow \min\{y_{0_{(m)}}, \min_{i_u \in I_{SUPER}} \{y_{p(i_u)}\}\}$
18 $y_{max} \leftarrow \max\{y_{1_{(m)}}, \max_{i_u \in I_{SUPER}} \{y_{p(i_u)}\}\}$
19 $DR'(m) = \{(x_{min}, y_{min}), (x_{max}, y_{max})\}$
20 $I' \leftarrow I \setminus I_{SUPER}$; $I' \leftarrow I' \cup \{i_{super}\}$; $S'' \leftarrow \emptyset$
21 **foreach** *street* $(i_u, i_v) \in S'$ **do**
22 **if** $i_u \notin I_{SUPER} \wedge i_v \notin I_{SUPER}$ **then**
23 $S'' \leftarrow S'' \cup \{(i_u, i_v)\}$
24 **else if** $i_u \in I_{SUPER} \wedge i_v \notin I_{SUPER}$ **then**
25 $S'' \leftarrow S'' \cup \{(i_{super}, i_v)\}$
26 $l((i_{super}, i_v)) \leftarrow l((i_u, i_v))$
27 **else if** $i_u \notin I_{SUPER} \wedge i_v \in I_{SUPER}$ **then**
28 $S'' \leftarrow S'' \cup \{(i_{super}, i_u)\}$
29 $l((i_{super}, i_u)) \leftarrow l((i_u, i_v))$
30 **end if**
31 **end foreach**
32 $G_{SUPER} = (I', S'')$
33 $T_{DR(m)} \leftarrow$ `Dijkstra`(G_{SUPER}, i_{super})
34 **return** $T_{DR(m)}$, $DR'(m)$
35 **end**

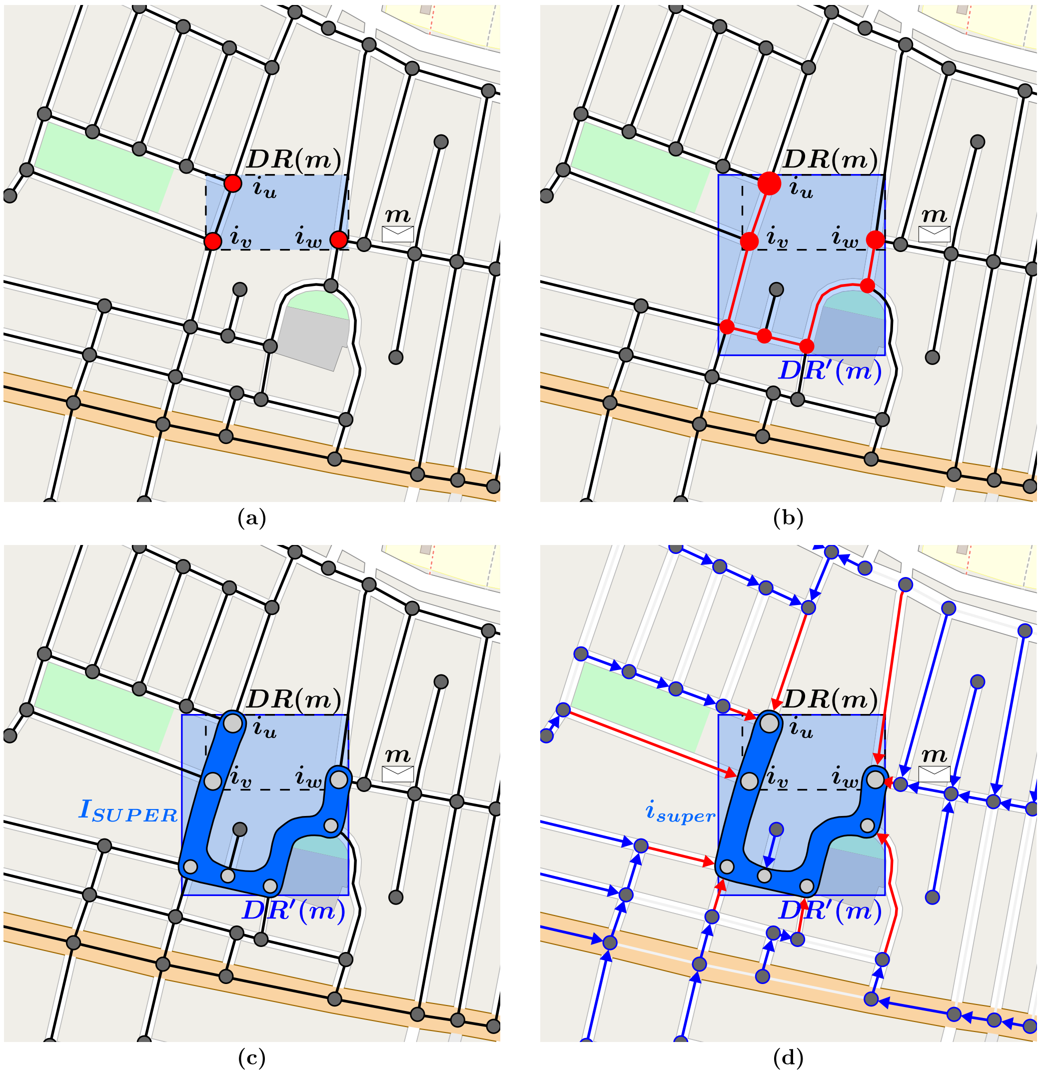

FIGURE 2. A visual example of the computation of the extended destination region and its corresponding shortest-path tree. (a) A black dashed line rectangle delimits the destination region $DR(m)$. (b) The extended destination region $DR'(m)$, in blue, contains the street intersection i_u, i_v and i_w that are located inside the $DR(m)$ as well as other intersections needed to connect them in the street-layout graph G_{STREET}. (c) The undirected street-layout graph, where I_{SUPER} is the connected component that contains all the paths needed to connect all the reachable points of the destination region. (d) The resulting shortest-path tree rooted at a supernode i_{super}. Red arrows indicate streets with access to the extended destination region.

F. PROPERTIES, COMPLEXITY, AND IMPLEMENTATION ASPECTS

We first present Theorem 1 and Theorem 2 that characterize the temporal complexity of the most relevant algorithms that implement the GeoTemporal-cast protocol. We also present a small discussion about implementation aspects of GeoTemporal-cast; in particular, we discuss strategies that can be used to perform the most computing-intensive tasks before their corresponding results are needed.

Lastly, we present Theorem 3 that shows that all the street segments with access to the destination region are either inside the extended destination region or directly connected to the supernode. The effect of this last property is that by following the gradient distances defined by the shortest-path tree $T_{DR(m)}$, packets will tend to be disseminated towards vehicles that are likely to enter the destination region.

Theorem 1: The complexity of Algorithm 1 is $O(|S|)$ where S is the set of road segments.

Proof: Computing the `distance()` function of Line 5 takes $O(|S|)$ because determining the street segment where a vehicle is located takes $O(|S|)$ and computing the distance from any node to the root of a tree also takes $O(|S|)$. All the other operations can be computed in constant time. □

Theorem 2: The complexity of Algorithm 2 is $\Theta((|S| + |I|)\log|I|)$ where S is the set of road segments and I is the set of street intersections.

Proof: Computing G'_{STREET} takes $O(|S| + |I|)$, computing I_{IN} takes $O(|I|)$, and determining $i_{centroid}$ takes $O(|I|)$.

Computing I_{SUPER} takes $\Theta((|S|+|I|)\log|I|)$ because we can use a single execution of Dijkstra's algorithm to compute and store the shortest paths from $i_{centroid}$ to all the nodes in I_{IN}. Determining the vertices of the extended destination region takes $O(|I|)$ because it involves computing max and min operations over sets of real number of cardinality $O(|I|)$. The **foreach** loop of Lines 21–31 can be implemented in $O(|S|\log|S|)$ if the set I_{SUPER} is stored in a self-balancing binary search tree. Therefore, the overall complexity of Algorithm 2 is $\Theta((|S|+|I|)\log|I|)$. □

Given that the size of the street-layout graph G_{STREET} can be in the order of thousands of intersections and street segments, computing Algorithm 2 can be considerably time-consuming. Fortunately, this computation can be performed in the background, before its result is needed. This is possible because the information needed to execute Algorithm 2 (i.e., $DR(m)$) is received with message m when the current node is elected as a viable carrier. The result of the execution of Algorithm 2, namely $T_{DR(m)}$, will be needed during a future contact opportunity when the current vehicle needs to determine if another vehicle is also a viable carrier. Moreover, nodes can store $T_{DR(m)}$ for as long as they have replicas to share of packets with the same destination region $DR(m)$.

This way, the overall complexity of determining the order in which data messages are forward during a contact opportunity (see Section IV-B, Steps 3 and 4) can be reduced to the cost of maintaining a fixed size priority queue, times the complexity of Algorithm 1 which is $O(|S|)$. It is important to point out that the execution time of Algorithm 1 can be further improved by executing a background process that prunes the street segments of $T_{DR(m)}$ that are far away from the current location of the vehicle.

Theorem 3: All the street segments (i_u, i_v) in $G'_{STREET} = (I, S')$ with access to the destination region $DR(m)$ are either inside the extended destination region $DR'(m)$ or directly connected to the super node i_{super} in $T_{DR(m)}$ (see Line 33 of Algorithm 2).

Proof: Let us proceed by contradiction and assume that there exist an $(i_u, i_v) \in S'$, such that $p(i_u) \in DR(m) \wedge p(i_v) \notin DR(m)$ ((i_u, i_v) has access to the destination region), and $i_u \notin I_{SUPER} \wedge i_v \notin I_{SUPER}$. This is not possible because by Lines 4–8 and 10 of Algorithm 2, i_u has to be in I_{SUPER}. Now, since all the nodes in I_{SUPER} are substituted by i_{super} (Lines 24–30), if $p(i_v) \notin DR'(m)$ then $i_v \notin I_{SUPER}$ and hence $(i_{super}, i_v) \in S''$ (Lines 25 and 28). □

V. PERFORMANCE EVALUATION

In this section, we present the results of a set of experiments based on detailed simulations in NS-3 [33], [34] and SUMO [35], [36]. For these experiments, we use the LuST Scenario [37] that is based in the city of Luxembourg, Luxembourg (Fig. 3). The map area is 155.95 km^2 (13.61 × 11.46 km), and has a 1,568.9 km road network. The LuST scenario contains the measured traffic load of the city during an entire day. From this information, we extracted mobility traces corresponding to three different periods. The one with the highest traffic load, the one with the lowest traffic load, and a typical interval with medium traffic load. Fig. 3 also shows the 12 destination regions used in the three scenarios. Their area size range from 1,760.78 m^2 to 5.18 km^2. Table 3 summarizes the number of vehicles involved in each scenario.

TABLE 3. Number of vehicles in every Luxembourg scenario.

Traffic load	Number of vehicles	
	Max. simultaneous	Total
Low	1,691	3,127
Medium	3,479	6,370
High	5,201	9,080

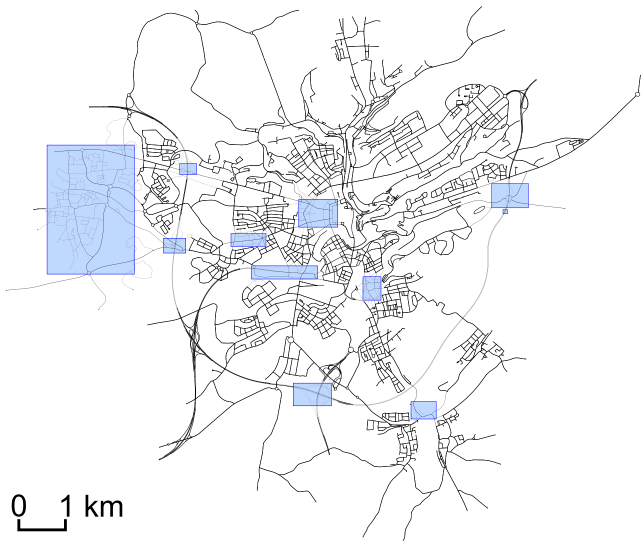

FIGURE 3. Realistic map of the city of Luxembourg, Luxembourg. The 12 blue rectangles represent the destination regions.

In all the scenarios, vehicles are equipped with IEEE 802.11b wireless radio interfaces with a maximum possible transmission range of 250 m. The reason for this selection is twofold. The first one is to consider heterogeneous scenarios where nodes can be the vehicles themselves, but also other types of devices, such as personal mobile devices that are transported in the vehicles. The second one is to obtain a baseline of the performance of the protocols when using off-the-shelf radios.

We compare the performance of GeoTemporal-cast against that of time-stable geocast variations of Spray & Wait [38], Binary Spray & Wait [38], Epidemic [32], and Geographically-Restricted Epidemic. The source code of all the protocols is available at `https://github.com/LuissRicardo/NS3-GeoTemporalProtocols`

These protocols operate as follows.

- *Time-Stable Epidemic*. Messages are disseminated epidemically until the expiration of their lifetime.
- *Geographically-restricted Time-Stable Epidemic*. Messages are disseminated epidemically until the expiration of their lifetime, but only within the minimum-size

TABLE 4. Simulation parameters.

Parameter	Value
Simulation time	600 seconds
Messages' lifetime	100 seconds
Emergency messages ratio	20% (rounded to nearest integer)
`HELLO` message interval	1 second
Packets queue size	128 packets
Wi-Fi interface	IEEE 802.11b
Transmission rate	1 Mbit/s
Transmission power	33 dBm
Energy detection threshold	-61.8 dBm
Transmission range	250 m (approximate)
Simulation independent runs	30
Confidence levels	95%
NS-3's version	3.29

rectangle that contains the destination region and the source node's location.

- *Time-Stable Spray & Wait*. Outside of the destination region, messages are disseminated using Spray & Wait. When a message reaches the destination region, it is disseminated epidemically until the expiration of their lifetime.
- *Time-Stable Binary Spray & Wait*. Outside of the destination region, messages are disseminated using Binary Spray & Wait. When a message reaches the destination region, it is disseminated epidemically until the expiration of their lifetime.

We use delivery ratio, delivery delay, data overhead, control overhead, and total overhead as performance metrics. The delivery ratio is computed according to (1), the delivery delay according to (2), and the data overhead according to (3). The total overhead is computed as the ratio between the total number of data and control bytes transmitted by all the vehicles, and the total number of data bytes successfully delivered at intended receivers. Similarly, the control overhead is computed as the ratio between the total number of control bytes transmitted by all the vehicles, and the total number of data bytes successfully delivered at intended receivers.

Table 4 summarizes the values of the remaining simulation parameters.

For all the results presented in the following sections, each colored line in the plots represents the average of the represented metric over thirty independent runs, and the shaded area represents the confidence interval with a confidence level of 95%. We performed a sensitivity analysis to determine adequate values for the most relevant configuration parameters of Time-Stable Epidemic, Geographically-Restricted Time-Stable Epidemic, Time-Stable Spray & Wait and, Time-Stable Binary Stray & Wait. The legends of the plots include, between parentheses, the values of these configuration parameters.

A. PERFORMANCE WITH INCREASING VEHICLE DENSITY

For this set of experiments, we vary the vehicle density to evaluate the protocols under different conditions related to the contact opportunities' length and frequency. There are 32 source vehicles, 16 of them are selected uniformly at random from vehicles that visit any of the destination regions. Upon arrival to their corresponding destination region, the source vehicles will generate 32 messages of 256 bytes. From these 32 messages, six (approximately 20%) are randomly marked as high-priority (Emergency) traffic, while the rest are marked as regular-priority (Normal) traffic. The temporal scope of these messages is set to $\tau = (time\_of\_arrival, time\_of\_arrival + 100$ seconds), which means that the messages have to be delivered right away and retained there for 100 seconds. The remaining 16 source vehicles are randomly selected from vehicles that do not visit a destination region. As in the previous case, these vehicles generate 32 messages where 20% are randomly marked as high-priority traffic. The destination region for these sources is selected uniformly at random from the set of destination regions. The temporal scope of these messages is set to $\tau = (current\_time + 30$ seconds$, current_time + 130$ seconds), which means that they have to be delivered 30 seconds after they were generated and retained there for 100 second. We executed each simulation for 600 seconds for each value of vehicle density (low, with 3,127 total vehicles participating in the simulation; medium, with 6,370 total vehicles; and high, with 9,080 total vehicles).

From Fig. 4(a) we can observe that GeoTemporal-cast for high-priority (Emergency) traffic outperforms the other protocols by consistently delivering 60% or more messages in the three scenarios, and 20% more messages than the second-best performing protocol in the high-density scenario. This indicates that by using the *Viable Carrier Condition* (see Section IV-C)) to select carrier vehicles, GeoTemporal-cast is able to take advantage of the contact opportunities to effectively transport and retain the time-stable geocast messages in their corresponding destination region during their lifetime. It is also worth noticing that even though regular-priority (Normal) traffic share the network with its high-priority counterpart, GeoTemporal-cast for normal traffic still attains a delivery ratio that is similar or better than that of the other protocols that are agnostic to the priority of the packets. It is also interesting to note that Time-Stable Epidemic performs slightly better than Geographically-Restricted Time-Stable Epidemic. The reason is that due to the way in which Geographically-Restricted Time-Stable Epidemic computes the forwarding regions, it is possible for street segments needed to get to the destination region to be left out of the forwarding region. In these situations, messages are not always forwarded towards the destination region. Lastly, the figure also shows that Time-Stable Spray & Wait and Time-Stable Binary Spray & Wait are the worst-performing protocols. This is due to the arbitrary way messages are forwarded by these protocols. It is common that none of the $L = 128$ replicas arrive on time to the destination region. A detailed analysis revealed that most of the messages that reach the destination region are generated by vehicles that are either close or inside their corresponding destination region.

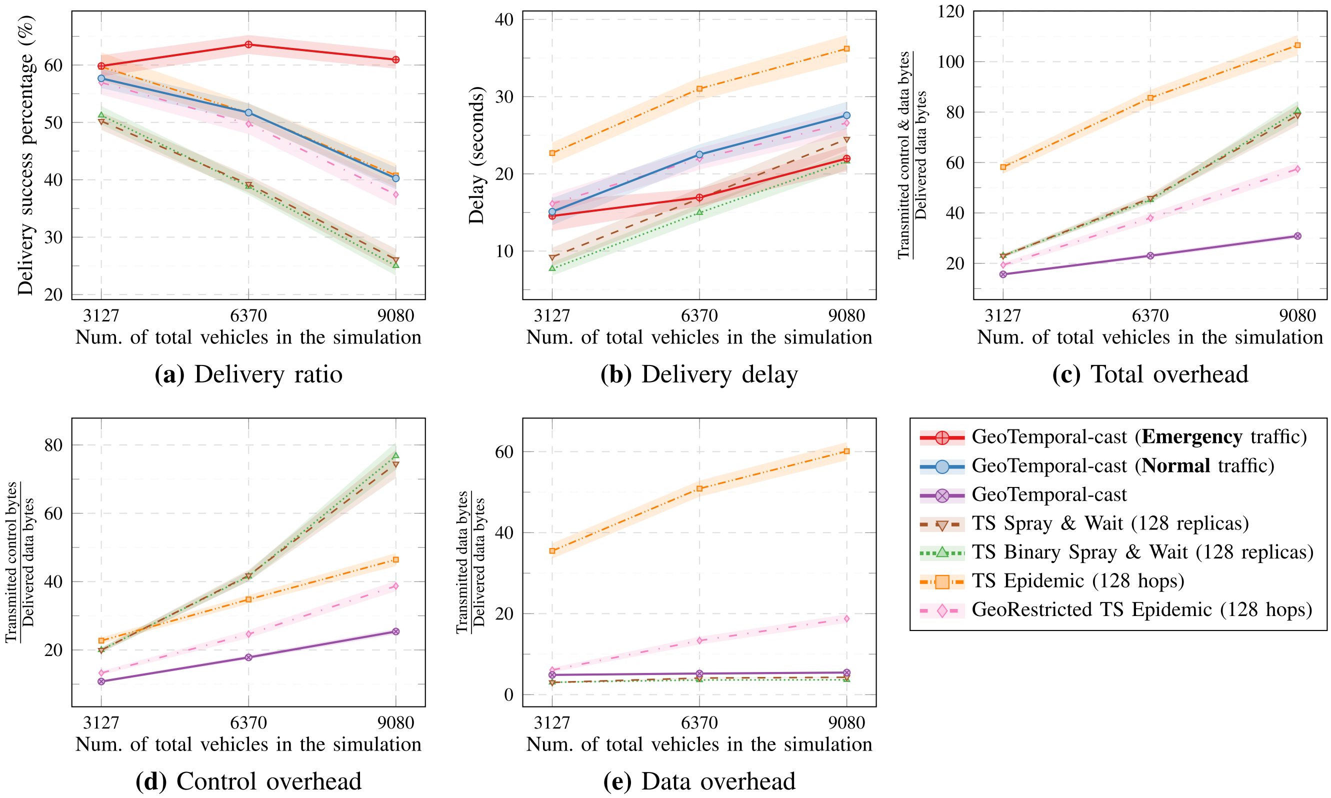

FIGURE 4. Performance with increasing node density. (a) Delivery ratio, (b) Delivery delay, (c) Total overhead, (d) Control overhead and, (e) Data overhead.

Fig. 4(b) shows that Time-Stable Binary Spray & Wait and Time-Stable Spray & Wait attain the lowest delays. However, as we argued in the previous paragraph, most of the messages that these two protocols deliver in time to their corresponding destination regions are generated by vehicles that are either close or inside these regions. The figure also shows that GeoTemporal-cast for emergency traffic attains better delivery delays than the Epidemic-based protocols. These results indicate that by forwarding packets according to the *Viable Carrier Condition*, GeoTemporal-cast effectively reduces the number of message replicas, which reduces the queuing delay experienced by them. As in the previous metric, even though emergency traffic is always forwarded before normal traffic, GeoTemporal-cast for normal traffic attains delays similar to those of the Epidemic-based protocols.

From Fig. 4(c) we can observe that GeoTemporal-cast nodes transmit consistently fewer bytes, per each byte of data that is received at an intended receiver. This is more apparent for the high-density scenario where GeoTemporal-cast induces close to half of the total overhead induced by Geographically-Restricted Time-Stable Epidemic, the second-best performing protocol for this metric. Similar to the previous metrics, the reason behind the efficiency exhibited by GeoTemporal-cast is that by using the *Viable Carrier Condition* to select carrier nodes, GeoTemporal-cast nodes only forward messages to vehicles which are closer to the destination region, and hence more likely to deliver the message to the intended receivers. This contrasts with the brute force approach of Time-Stable Epidemic that induces up to four times more overhead.

Figures 4(d) and (e) confirm the previous notion. Fig. 4(d) shows that GeoTemporal-cast consistently induces less control overhead than the other protocols. This indicates that GeoTemporal-cast effectively takes advantage of the extra control bytes containing the position and traveling direction of vehicles included in the control packets to efficiently deliver messages to the intended receivers. From the figure, we can also notice that, for this metric, the Spray & Wait-based protocols are the worst-performing protocols. The reason is that nodes that have spent their L replicas tend to transmit useless control packets. Lastly, from Figure 4(e), we can notice that the data overhead incurred by GeoTemporal-cast is similar to that of the Spray & Wait-based protocols that strictly limit the total number of replicas of any given packet in the network. As expected, this figure also shows that Time-Stable Epidemic is the least efficient protocol.

B. PERFORMANCE WITH INCREASING NUMBER OF SOURCE VEHICLES

In this set of experiments, we evaluate the protocols' performance as we increase the number of vehicles that generate

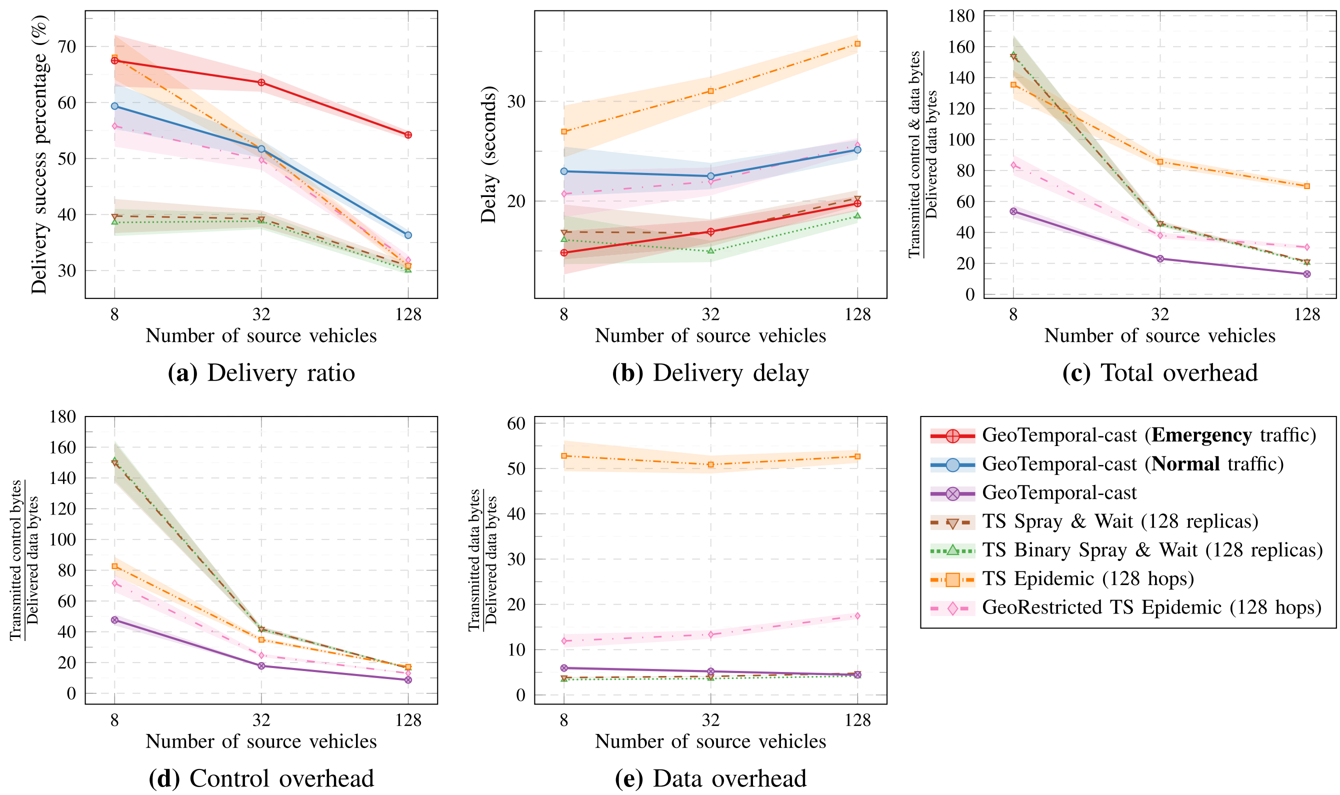

FIGURE 5. **Performance with increasing number of source vehicles. (a) Delivery ratio, (b) Delivery delay, (c) Total overhead, (d) Control overhead and, (e) Data overhead.**

time-stable geocast messages. The propose is to evaluate the ability of the protocols to handle an increasing data traffic load. We use the scenario with medium vehicle density (6,370 total vehicles) and vary the number of sources from 8 to 128. All the remaining simulation parameters are the same as in the previous section.

Fig. 5(a) shows that GeoTemporal-cast for emergency traffic and Time-Stable Epidemic attain a similar delivery ratio under light traffic load. Unfortunately, due to its brute force approach, the performance of Time-Stable Epidemic drops drastically when more vehicles inject messages into the network. This is not the case of GeoTemporal-cast for emergency traffic that manages to deliver up to 20% more messages that the second-best performing protocol when there are 128 source vehicles. From the figure, we can also notice that the performance of GeoTemporal-cast for normal traffic is similar or better than that of Geographically Restricted Time-Stable Epidemic, and consistently better than that of the Spray & Wait-based protocols. It is important to remark that GeoTemporal-cast for normal traffic achieves these results while sharing the network resources with the high-priority emergency traffic. In general, GeoTemporal-cast scales better than the other protocols thanks to the street-layout-aware way in which it selects message-carrier vehicles.

The results presented in Figs. 5(b-e) are consistent with those of the previous section, namely, that GeoTemporal-cast attains similar or better delays than that of the other protocols but at a smaller network cost.

VI. CONCLUSION

We introduced GeoTemporal-cast, an opportunistic time-stable geocast routing protocol for DT-VANETs that takes advantage of the street-layout topological information available to current navigation systems to select, according to the *Viable Carrier Condition*, message-carrier vehicles. These vehicles disseminate time-stable geocast messages towards their corresponding geographic destination regions and then retain the messages there during the messages' lifetime. GeoTemporal-cast is the first time-stable geocast routing protocol for DT-VANETs that does not assume that either the messages are generated inside the destination region or that an underlying routing protocol will deliver the packets there. A detailed large-scale simulation study based on real mobility traces revealed that GeoTemporal-cast outperforms Time-Stable versions of Epidemic, Geographically-Restricted Epidemic, Binary Spray & Wait and Spray & Wait by consistently delivering, in time, more messages to their intended destinations while inducing less overhead. These results indicate that by using the *Viable Carrier Condition*, GeoTemporal-cast effectively selects message-carrier vehicles that are likely to reach the destination region.

An important implementation aspect is that the computation of the shortest-path tree rooted at a connected component that includes all the relevant street intersections, which is the most computing-intensive task, can be performed between contact opportunities. This way, the result can be available before the next contact opportunity when it is actually needed.

Due to its efficacy and efficiency, GeoTemporal-cast can support a growing class of Smart City and Intelligent Transportation Systems applications that require efficient dissemination of position- and time-based abiding information.

REFERENCES

[1] O. K. Tonguz, "Red light, green light—No light: Tomorrow's communicative cars could take turns at intersections," *IEEE Spectr.*, vol. 55, no. 10, pp. 24–29, Oct. 2018.

[2] E. Schoch, F. Kargl, and M. Weber, "Communication patterns in VANETs," *IEEE Commun. Mag.*, vol. 46, no. 11, pp. 119–125, Nov. 2008.

[3] R. Almeida, R. Oliveira, M. Luis, C. Senna, and S. Sargento, "Forwarding strategies for future mobile smart city networks," in *Proc. IEEE Veh. Technol. Conf.*, Jun. 2018, pp. 1–7.

[4] B. T. Sharef, R. A. Alsaqour, and M. Ismail, "Vehicular communication ad hoc routing protocols: A survey," *J. Netw. Comput. Appl.*, vol. 40, no. 1, pp. 363–396, Apr. 2014.

[5] S. Boussoufa-Lahlah, F. Semchedine, and L. Bouallouche-Medjkoune, "Geographic routing protocols for vehicular ad hoc networks (VANETs): A survey," *Veh. Commun.*, vol. 11, pp. 20–31, Jan. 2018.

[6] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *IEEE Veh. Technol. Mag.*, vol. 2, no. 2, pp. 12–22, Jun. 2007.

[7] R. Oliveira, M. Luís, and S. Sargento, "On the performance of social-based and location-aware forwarding strategies in urban vehicular networks," *Ad Hoc Netw.*, vol. 93, Oct. 2019, Art. no. 101925.

[8] B. Alsubaihi and A. Boukerche, "Semantic and self-decision geocast protocol for data dissemination over VANET (SAS-GP)," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2015, pp. 1948–1953.

[9] F. Bai and B. Krishnamachari, "Exploiting the wisdom of the crowd: Localized, distributed information-centric VANETs [topics in automotive networking," *IEEE Commun. Mag.*, vol. 48, no. 5, pp. 138–146, May 2010.

[10] M. Zappatore, A. Longo, and M. A. Bochicchio, "Using mobile crowd sensing for noise monitoring in smart cities," in *Proc. Int. Multidisciplinary Conf. Comput. Energy Sci. (SpliTech)*, Jul. 2016, pp. 1–6.

[11] C. Pereira, J. Rodrigues, A. Pinto, P. Rocha, F. Santiago, J. Sousa, and A. Aguiar, "Smartphones as M2M gateways in smart cities IoT applications," in *Proc. 23rd Int. Conf. Telecommun. (ICT)*, May 2016, pp. 1–7.

[12] J. Timpner and L. Wolf, "Query-response geocast for vehicular crowd sensing," *Ad Hoc Netw.*, vol. 36, pp. 435–449, Jan. 2016.

[13] C. Fiandrino, F. Anjomshoa, B. Kantarci, D. Kliazovich, P. Bouvry, and J. N. Matthews, "Sociability-driven framework for data acquisition in mobile crowdsensing over fog computing platforms for smart cities," *IEEE Trans. Sustain. Comput.*, vol. 2, no. 4, pp. 345–358, Oct. 2017.

[14] R. Menchaca-Mendez, B. Luna-Nuñez, R. Menchaca-Mendez, A. Yee-Rendon, R. Quintero, and J. Favela, "Opportunistic mobile sensing in the fog," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–18, Aug. 2018.

[15] M. Silva, G. Signoretti, J. Oliveira, I. Silva, and D. G. Costa, "A crowdsensing platform for monitoring of vehicular emissions: A smart city perspective," *Future Internet*, vol. 11, no. 1, p. 13, Jan. 2019.

[16] D. Costa, A. Damasceno, and I. Silva, "CitySpeed: A crowdsensing-based integrated platform for general-purpose monitoring of vehicular speeds in smart cities," *Smart Cities*, vol. 2, no. 1, pp. 46–65, Feb. 2019.

[17] C. Wang, X. Gaimu, C. Li, H. Zou, and W. Wang, "Smart mobile crowdsensing with urban vehicles: A deep reinforcement learning perspective," *IEEE Access*, vol. 7, pp. 37334–37341, 2019.

[18] C. Maihöfer, T. Leinmüller, and E. Schoch, "Abiding geocast: Time–stable geocast for ad hoc networks," in *Proc. 2nd ACM Int. Workshop Veh. Ad Hoc Netw.*, 2005, pp. 20–29.

[19] I. Goto, D. Nobayashi, K. Tsukamoto, T. Ikenaga, and M. Lee, "Transmission control method to realize efficient data retention in low vehicle density environments," in *Proc. Int. Conf. Intell. Netw. Collaborative Syst.* Cham, Switzerland: Springer, 2019, pp. 390–401.

[20] F. Z. Bousbaa, F. Zhou, N. Lagraa, and M. B. Yagoubi, "Robust geocast routing protocols for safety and comfort applications in VANETs," *Wireless Commun. Mobile Comput.*, vol. 16, no. 10, pp. 1317–1333, Jul. 2016.

[21] F. Zhang, B. Jin, Z. Wang, H. Liu, J. Hu, and L. Zhang, "On geocasting over urban bus-based networks by mining trajectories," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 6, pp. 1734–1747, Jun. 2016.

[22] E. Gurumoorthi and A. Ayyasamy, "Cache agent based location aided routing protocol using direction for performance enhancement in VANET," *Wireless Pers. Commun.*, vol. 109, no. 2, pp. 1195–1216, Nov. 2019.

[23] C. Maihöfer and R. Eberhardt, "Time-stable geocast for ad hoc networks and its application with virtual warning signs," *Comput. Commun.*, vol. 27, no. 11, pp. 1065–1075, Jul. 2004.

[24] S. D. Hermann, C. Michl, and A. Wolisz, "Time-stable geocast in intermittently connected IEEE 802.11 MANETs," in *Proc. IEEE 66th Veh. Technol. Conf.*, Sep. 2007, pp. 1922–1926.

[25] Q. Yu and G. Heijenk, "Abiding geocast for warning message dissemination in vehicular ad hoc networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2008, pp. 400–404.

[26] H. Rahbar, K. Naik, and A. Nayak, "DTSG: Dynamic time-stable geocast routing in vehicular ad hoc networks," in *Proc. 9th IFIP Annu. Medit. Ad Hoc Netw. Workshop (Med-Hoc-Net)*, Jun. 2010, pp. 1–7.

[27] Y. Lim, S. Ahn, and K.-H. Cho, "Abiding geocast for commercial ad dissemination in the vehicular ad hoc network," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2011, pp. 115–116.

[28] O. Kaiwartya, S. Kumar, and R. Kasana, "Traffic light based time stable geocast (T-TSG) routing for urban VANETs," in *Proc. 6th Int. Conf. Contemp. Comput. (IC)*, Aug. 2013, pp. 113–117.

[29] X. Zhang, L. Yan, and W. Li, "Efficient and reliable abiding geocast based on carrier sets for vehicular ad hoc networks," *IEEE Wireless Commun. Lett.*, vol. 5, no. 6, pp. 660–663, Dec. 2016.

[30] I. Leontiadis and C. Mascolo, "Opportunistic spatio-temporal dissemination system for vehicular networks," in *Proc. 1st Int. MobiSys Workshop Mobile Opportunistic Netw. (MobiOpp)*, 2007, pp. 39–46.

[31] I. Leontiadis and C. Mascolo, "GeOpps: Geographical opportunistic routing for vehicular networks," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw.*, Jun. 2007, pp. 1–6.

[32] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke Univ., Durham, NC, USA, Tech. Rep. CS-2000-06, 2000.

[33] *Network Simulator 3 Website*. Accessed: Jan. 7, 2020. [Online]. Available: https://www.nsnam.org/

[34] G. F. Riley and T. R. Henderson, *The ns-3 Network Simulator*. Berlin, Germany: Springer, 2010, pp. 15–34.

[35] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO—Simulation of urban mobility," *Int. J. Adv. Syst. Meas.*, vol. 5, pp. 128–138, Dec. 2012.

[36] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo—Simulation of urban mobility: An overview," in *Proc. 3rd Int. Conf. Adv. Syst. Simulation*, 2011, pp. 1–6.

[37] L. Codeca, R. Frank, S. Faye, and T. Engel, "Luxembourg SUMO traffic (LuST) scenario: Traffic demand evaluation," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 2, pp. 52–63, Apr. 2017.

[38] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM SIGCOMM Workshop Delay-Tolerant Netw. (WDTN)*, 2005, pp. 252–259.

LUIS R. GALLEGO-TERCERO received the B.S. degree in computer science engineering from the Instituto Tecnológico de Tepic, Tepic, Nayarit, in 2011, and the M.S. degree in computer science from the Instituto Politécnico Nacional, Mexico City, Mexico, in 2015, where he is currently pursuing the Ph.D. degree in computer science with the Centro de Investigación en Computación (CIC).

ROLANDO MENCHACA-MENDEZ (Member, IEEE) received the B.S. degree in electronic engineering from the Universidad Autónoma Metropolitana, Mexico City, Mexico, in 1997, the M.S. degree in computer science from the Instituto Politécnico Nacional (IPN), Mexico City, in 1999, and the Ph.D. degree in computer engineering from the University of California Santa Cruz, Santa Cruz, CA, USA, in 2009. He is currently a Professor with the Centro de Investigación en Computación (CIC), IPN.

MARIO E. RIVERO-ANGELES (Member, IEEE) was born in Mexico City, Mexico, in 1976. He received the B.Sc. degree from the Universidad Autónoma Metropolitana (UAM), Mexico, in 1998, and the M.Sc. and Ph.D. degrees in electrical engineering from CINVESTAV-IPN, in 2000 and 2006, respectively. He has been a Professor with the Centro de Investigación en Computación (CIC-IPN), Instituto Politécnico Nacional, Mexico, since 2002. He was a Postdoctoral Fellow at the Dyonisos research project with the Institut National de Recherche en Informatique et en Automatique, Rennes, France, from 2007 to 2010. His research interests include random access protocols and data transmission in cellular networks, P2P networks, and wireless sensor networks.

RICARDO MENCHACA-MENDEZ (Member, IEEE) received the B.S. degree in computer science from the Escuela Superior de Cómputo, Mexico City, Mexico, in 2001, the M.S. degree from the Instituto Politécnico Nacional (IPN), Mexico City, in 2005, and the Ph.D. degree in computer science from the University of California at Santa Cruz, in 2013. He is currently a Professor with the Centro de Investigación en Computación (CIC), IPN.