Hindawi Publishing Corporation
Mobile Information Systems
Volume 2015, Article ID 490191, 16 pages
http://dx.doi.org/10.1155/2015/490191

Research Article

HCBLS: A Hierarchical Cluster-Based Location Service in Urban Environment

Raik Aissaoui,[1,2] Amine Dhraief,[2] Abdelfettah Belghith,[3] Hamid Menouar,[4] Hassan Mathkour,[3] Fethi Filali,[4] and Adnan Abu-Dayya[4]

[1] *College of Engineering, Qatar University, Al Tarfa, Doha 2713, Qatar*
[2] *HANA Research Lab, University of Manouba, 2010 Manouba, Tunisia*
[3] *College of Computer and Information Sciences, King Saud University, Riyadh 12372, Saudi Arabia*
[4] *Qatar Mobility Innovations Center, Qatar Science & Technology Park, Doha 210531, Qatar*

Correspondence should be addressed to Abdelfettah Belghith; abdelfettah.belghith@hotmail.com

Received 28 July 2015; Revised 15 October 2015; Accepted 8 November 2015

Academic Editor: Laurence T. Yang

Vehicle location information is central to many location-based services and applications in VANETs. Tracking vehicles positions and maintaining an accurate up-to-date view of the entire network are not easy due to the high mobility of vehicles and consequently rapid topology changes. The design of a scalable, accurate, and efficient location service is still a very challenging issue. In this paper, we propose a lightweight hierarchical cluster-based location service in city environments (HCBLS). HCBLS integrates a logical clustering based on the city digital map and consequently does not involve extra signaling overhead. An advanced location update aggregation at different levels of the assumed hierarchy is adopted to maintain up-to-date and accurate location information. Simulation results show that HCBLS achieves much better performances than the Efficient Map-Based Location Service (EMBLS) and any regular (non-cluster-based) updating scheme. HCBLS increases the success rate by around 10%, improves the overview of the network by more than 30%, lowers the location update and query costs by more than 7 times, lowers the message delivery latency by around 3 times, and presents around 4 times better localization accuracy.

1. Introduction

The automobile industry is experiencing an unprecedented shift: vehicles are no more seen just as a mean of transportation but they are considered as computers on wheels [1–4]. According to several studies [5, 6], by 2025 every vehicle will be connected through several vehicular wireless communication systems (embedded devices, smartphones, etc.) to both road infrastructures (Vehicle to Infrastructure (V2I)) and other vehicles (Vehicle to Vehicle (V2V)) [7, 8]. Vehicle manufacturers believe that V2V and V2I and their applications will contribute to further improvement of safety in vehicular environments. In order to fully take advantage of these emerging applications, Intelligent Transportation Systems (ITS) require supportive communication technologies and protocols. The IEEE has recently developed the WAVE (Wireless Access in Vehicular Environment) standard, including the IEEE 1609 protocol family and the IEEE 802.11p [9]. The WAVE standard is based on the 802.11 architecture; however, it achieves higher data rates and provides a wider communication range. Similarly, ETSI ITS TC has proposed a new protocol stack based on a variety of existing and new access technologies to enable ITS applications [10].

The ubiquitous connectivity in vehicular environments will lead to the development of a large set of new safety and nonsafety applications [11, 12]. However, an essential condition for the sustainable safety improvement is an accurate vehicle positioning to be shared among key road components. Indeed, real-time vehicle position is a cornerstone in several safety applications such as emergency notifications, crash detection and avoidance systems, and road work alerts [13–15]. Any vehicle can easily acquire its current geographical position through an embedded global navigation satellite system (GNSS) receiver. Information on the positions of other vehicles in the system can be provided by some location servers which maintain an up-to-date snapshot of all vehicles positions [16, 17]. All vehicles are supposed to regularly update their positions at the location servers which are disseminated throughout the city. A sender vehicle queries these servers to accurately retrieve any destination (vehicle) current coordinates. To get accurate coordinates, an efficient location service should reduce the location update costs and achieve a short location response delay.

In this paper, we design an efficient hierarchical cluster-based location service for the urban environment requiring no extra overhead. It relies on a very limited number of location servers to efficiently provide location updates and queries. Clustering is performed based on the road map of the city. Roads are logically divided into fragments of a certain fixed dimension. Vehicles currently located in a given road fragment form a cluster. The dynamically selected cluster head in a given cluster at a given instant is the vehicle nearest to the geographical center of its road segment. As such, the different cluster heads of the different clusters are dynamically selected with no extra signaling cost. Throughout the paper we deliberately use the word cluster to refer either to the vehicles within its corresponding fragment or to the road fragment itself. Dynamically selected cluster heads are responsible of feeding the location servers with vehicles current positions. The new clustering protocol avoids known drawbacks of existing clustering protocols for high mobility networks where maintaining the clusters requires additional signaling messages [18].

In order to evaluate the performance of the proposed solution, we implemented the proposal and conducted extensive simulation tests using the network simulator NS3 version [19] of the simulation platform developed by the European research project iTETRIS [20] alongside with the simulator for urban mobility SUMO [21]. Obtained results show that we have an accurate overview of the vehicles circulating throughout the simulated region, yet the required signaling overhead is very low compared to that needed by the Efficient Map-Based Location Service (EMBLS) [22] and the Hybrid Location Service (HLS). The HLS is a location service mechanism that is based on regular flat location updating without any clustering comparable in these terms to [23]. However, it is leveled up to use the same V2V and V2I technologies as the proposed HCBLS. As such, HLS is a variant of HCBLS where the location update mechanism is performed directly to the nearest Road Side Unit (RSU) without clustering.

The rest of this paper is structured as follows. Section 2 presents a taxonomy of the location service and reviews some relevant related works. Section 3 details the proposed solution. Section 4 describes the simulation studies and discusses the obtained results. Finally, we conclude and highlight some future research orientations.

2. Related Work

We focus on some relevant proposals of a location service suited for an urban environment, and we adopt a taxonomy of existing location service paradigms.

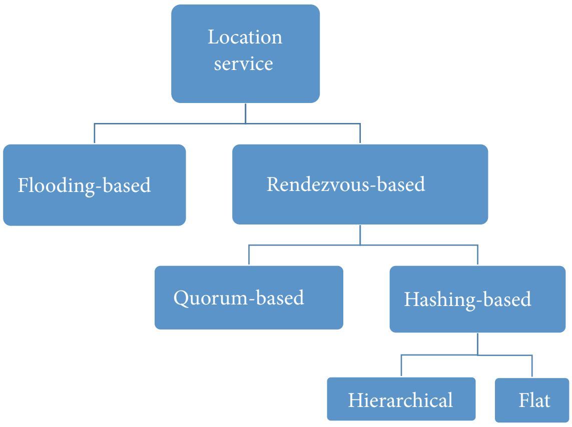


Figure 1: Location service taxonomy.

2.1. Location Service Taxonomy. Location services can be divided into two main categories: flooding-based and rendezvous-based (as shown in Figure 1).

In a flooding-based location service [24], the location updates or queries are flooded into the network. This flooding covers the entire network at the cost of a broadcast storm [25]. Thereby, it causes a huge load on the network which consumes valuable network resources. This location service family is inefficient and has a poor scalability [26]. Since the flooding-based approach severely degrades the network performance, we solely focus on rendezvous-based approaches.

In a rendezvous-based location service, a set of location servers are placed throughout the network and are considered as rendezvous points. According to a defined update strategy, a vehicle updates its location information to a subset of these location servers. The rendezvous-based location service can further be classified into quorum and hashing-based location services.

In a quorum-based location service [27], two overlapping groups of nodes are formed: an update quorum and a search quorum. The update quorum holds all nodes current positions, whereas the search quorum handles nodes queries. A source vehicle asking for a specific destination position sends its requests to the nearest node in the search quorum. This request is forwarded until it reaches a node member of both the search and the update quorum. This solution is proposed for ad hoc and sensor networks and may not be suitable for vehicular networks as forming and maintaining vehicular groups is a tough task due to vehicles' high mobility. Two main examples of a quorum-based location service are quorum-based sink location service for irregular wireless sensor networks [28] and quorum-based location service in Vehicular Sensor Networks [29].

Hashing-based location services rely on a hash function which maps each node to location servers. This hash function takes as input a node identifier while its output is either a location server(s) identifier or its/their coordinates. The hashing-based location service family is divided into two main categories: flat and hierarchical location service. In a flat hashing-based location service [30], the map is divided into several zones without considering any hierarchy between these zones. A hash function is used to map each node to one or many location servers in different zones. In a hierarchical hashing-based location service, location servers are selected at different levels within the city. The location query is forwarded to higher level location servers if the requested information is not available at a certain level.

The essential difference between quorum-based and hashing-based mechanisms has been theoretically analysed and experimentally investigated in [16]. The authors compared three location service protocols: a quorum-based location service (Column-Row Location Service (XYLS)), a hierarchical hashing-based location service (Grid Location Service (GLS)), and a flat hashing-based location service (Geographic Hashing Location Service (GHLS)). They showed that although GLS asymptotically is more scalable than both GHLS and XYLS in terms of signaling load, GHLS is far simpler and transmits fewer packets than GLS in dense networks. Similarly, although XYLS scales worse asymptotically than GLS, it transmits fewer control packets and delivers more data packets than GLS in large mobile networks.

2.2. Location Service in the Urban Environment. Saleet et al. proposed in [31] a Region-Based Location Service Management Protocol (RLSMP). This protocol targets improving the scalability of location service in VANETs. The entire urban area is divided into cells. In each cell, a cluster head aggregates location information from vehicles within its cell and sends the information to a Location Service Cell (LSC). A vehicle querying for destination coordinates sends its location query to its LSC. If the requested information is not found in this LSC, the query is forwarded to the neighboring LSCs following a spiral shape until the coordinates of the destination are found. The RLSMP generates, however, a significant signaling load as the same query may visit several LSCs before reaching the one holding the information if any.

A Map-Based Location Service is proposed in [32, 33]. The entire urban area is recursively divided into hierarchical squares wherein a “waypoint” is selected to be the position of the location server. In the Map-Based Location Service (MBLS) [32], waypoints are road intersections. MBLS uses a predefined hash function to map each node identifier to a corresponding waypoint at each level of the hierarchy. The closest node to the selected waypoints acts therefore as a location server. When a node crosses a new waypoint, it sends a location update to the new location server including its position, the road segment along which it is driving, and its direction and its speed. In Density-Aware Map-Based Location Service (DMBLS) [33], the server selection is based on vehicular density at a particular waypoint. Location servers are vehicles located near some of the road intersections with a high traffic density. Waypoints are also used as reference points for sending location updates. A query is forwarded to higher levels until the required information is found and replied back to the client vehicle. In these two Map-Based Location Service solutions [32, 33], the consistency of the location server (waypoints) selection is highly affected by the mobility of the nodes. Indeed, when a given node moves into a new square, its previous location information stored in the location servers becomes outdated and obsolete. In addition, the movement of the nodes acting as location servers should be treated accordingly to ensure having an available location server.

In [34], a Vehicle Location Service Protocol (VLS) is proposed to support V2V communication in urban environment. The network is divided into $(M \times N)$ grids. Each grid contains a location server. The coordinate of the location server of a vehicle is derived from the vehicle’s identifier and a hash function. Hence, a vehicle can send its location update message using the coordinates of the nearest location server. This location update is then forwarded to the remaining location servers along the minimum spanning tree. A vehicle can query another vehicle’s location by sending the location query message to the coordinates of the nearest location server of the queried vehicle. Those coordinates are derived from the queried vehicle’s ID and the hash function used. This solution suffers from a high query response time. In fact, a query may be forwarded to all the grids in the tree. Furthermore, when a location server leaves its position, it has to transfer its location information database to a new location server. This process incurs a high signaling load due to the high mobility of the vehicles. Finally, if there is no forwarder, the spanning tree becomes useless.

In [35], Hsu and Wu proposed an Efficient Cost-Based (ECB) Location Service Protocol for VANETs. A set of cost functions are defined to evaluate the performance of the location service protocol. The map is divided into several level-1 grids. Nine level-1 grids are grouped into a level-2 grid. Each level-2 grid holds a location server. A location server has a dual role: it serves as a local location server of the local vehicles and a dedicated location server of a subset of the remaining vehicles. The location update cost of the ECB scheme is lower than that of VLS because the location update message only needs to be sent to the local location server and the dedicated location servers. The location query cost of the ECB scheme is higher than that of VLS because only a subset of the location servers has the location information of the queried vehicle and hence the location query message needs to travel a longer distance.

In [22], Ashok et al. proposed an adaptive location update policy called Efficient Map-Based Location Service (EMBLS). The location service uses a vehicle density-aware server selection policy to select servers at high density regions of the urban area. The location servers are vehicles. This solution considered two hierarchy levels. The entire urban area is divided into four squares. At level 1 (within each square), a number of intersections with higher densities are selected for hosting Intersection Leaders (IL). Vehicles send their location updates to the nearest IL. At level 2 (nearest to the center of the urban area), at a high density intersection, Location Servers (LS) are selected. A localized query answering strategy is used for replying to location queries where a server with the latest location information replies to the query. The request is forwarded hop by hop until it eventually reaches one of the location servers of the destination which replies back with the position information. This approach is advantageous when the source and the destination are close enough to be in the same square and thus the request does not need to traverse long distances. In this solution, location servers are elected and this amounts to an additional signaling overhead. Each vehicle is supposed to know the square in which it currently resides and the intersections in the square and their densities. Each vehicle receives a periodic update of this information within a specific traffic information message. This periodical updating consumes valuable network resources and induces a high signaling load.

In summary, the different existing location service protocols for the urban environment are incurring either high location updating costs like flooding-based location services and VLS or a long location query delay like ECB and EMBLS. To improve the existing work, we propose a novel location service protocol which minimizes both the total location updates and the location query cost and therefore the total signaling load.

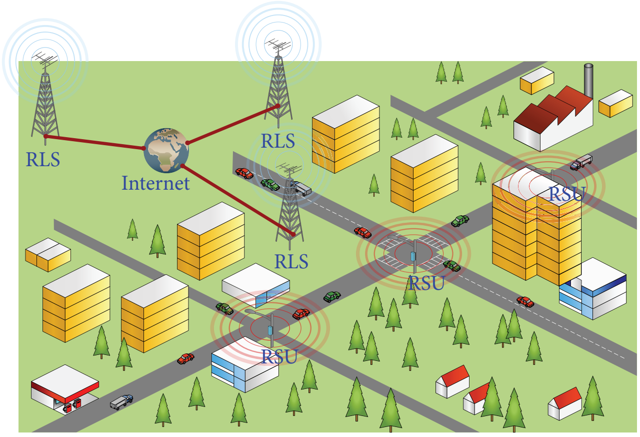


Figure 2: Location servers hierarchy.

3. A Hierarchical Cluster-Based Location Service in Urban Environment

Any positioning system should reduce as much as possible the control or signaling load while preserving the validity and the accuracy of the vehicles' positions. The location service periodically gathers vehicle positions through a location update mechanism. An update mechanism based on flooding for instance will rapidly exhaust the available network resources. A cluster-based strategy can minimize the impact of location updates on the available network resources. However, as it has been previously stated in Section 2, regular clustering techniques are not suitable for vehicular environments. We propose in this work an original and lightweight cluster-based location update mechanism adapted to high mobility nodes and providing an accurate real-time positioning of vehicles within a certain area of interest.

A location service should also provide an accurate response to a location service request. This response usually includes the target vehicle's identifier, its coordinates, and a timestamp. To ensure a low query latency, our solution relies on a hierarchy, a regional location server layer and a layer of RSUs, to spread the location service query quickly for long distances and large areas. We divide the urban area into a two-level hierarchy (Figure 2). At level 1, the RSUs are deployed at some intersections or ultimately at every road intersection. The impact of the progressive RSU market penetration will be investigated in Section 4.2. Cluster head vehicles send their location updates to the nearest Road Side Unit (RSU). At level 2, we deploy one regional location server (RLS) near the geographical center of the urban area. RSUs periodically send their location tables to the corresponding RLS. The RLS is the unique entity that holds the complete and comprehensive view of the vehicles currently circulating in the city.

3.1. Technologies. Each vehicle is equipped with an embedded computer, an on-board wireless network interface (supporting 802.11p [36]), and a global navigation satellite system (GNSS) receiver (such as a Global Positioning System (GPS) receiver (http://www.gps.gov/)). A digital map is preinstalled on the embedded computer. It enables the association of the vehicle's geographic position to its current localization within urban roads. Each vehicle periodically sends beacon messages to inform nodes in its vicinity of its current position, direction, and velocity. Road Side Units (RSUs) use 802.11p communication technology for V2I communication and LTE for infrastructure to infrastructure (I2I) communication. Regional location servers are equipped with LTE technology for I2I communication and a wired technology to get connected with other regional location servers. Figure 2 shows the different required technologies used in our proposal.

Recently, some studies proposed heterogeneous or hybrid network architectures to combine the advantages of both infrastructure-based and infrastructure-less ad hoc networking architectures. In [37], authors proposed a theoretical framework which compares the basic patterns of both technologies in the context of safety-of-life vehicular scenarios. They presented a mathematical model for the evaluation of the considered protocols essentially in terms of the probability of a successful beacon delivery. Authors concluded that the abilities of LTE to support beaconing for vehicular safety applications are rather poor. The network becomes easily overloaded even under idealistic assumptions. Moreover, cellular networks are not at no cost for this kind of operation.

In [38], authors put in evidence how LTE is expected to play a critical role to overcome sparse network situations where 802.11p-equipped vehicles are not within the transmission range. Moreover, LTE can be particularly helpful at intersections by enabling the reliable exchange of cross-traffic assistance applications when 802.11p communications are hindered by nonline of sight conditions due to buildings. However, the LTE centralized architecture remains the main concern. It does not natively support V2V communications as it requires passing via infrastructure nodes of the core network that should intercept uplink traffic before being redistributed to concerned vehicles.

In [39], authors argued the use of IEEE 802.11p and LTE in VANETs and presented the pros and cons of the use of each of these two technologies. Using 802.11p, a vehicle periodically exchanges beacon messages with other vehicles either directly or via RSUs in an ad hoc manner. In fact, the IEEE 802.11p defines a way to exchange data without the need to establish a basic service set (BSS) (there is no authentication or association). In contrast, in LTE this exchange is carried out through the base station node (eNB in LTE) of the cellular network. In this latter case, all the beacons that are received at the eNB have to traverse the entire core network (i.e., the Evolved Packet Core) before they can be disseminated to the rest of the vehicles within the network. Authors concluded that LTE meets most of VANET application requirements; however, its performance remains very sensitive to the sustained overhead as well as to the number of cellular network users.

Multihop Cluster-Based IEEE 802.11p and LTE Hybrid Architecture for VANET Safety Message Dissemination, namely, VMaSC-LTE, was proposed in [40]. Authors explained that, in networks deploying only LTE, the delay and delivery ratio of safety message dissemination are degraded due to the broadcast storm and disconnected network problems at both high and low vehicle densities. They further mentioned that a pure cellular based VANET communication is not feasible due to the high cost of communication between vehicles and the base stations. Besides, the high mobility of vehicles gives rise to a high number of hand-off occurrences at the base stations. In consequence, authors proposed a hybrid architecture, combining IEEE 802.11p based multihop clustering and LTE, with the goal of achieving at the same time a high data packet delivery ratio and a low delay.

HCBLS is a Hybrid Location Service (Figure 2). Vehicles coordinates are provided by GPS and are transferred among vehicles and to Road Side Units using 802.11p. LTE technology is used between Road Side Units and regional location servers. These later are connected through the Internet infrastructure.

3.2. Location Table Data Structure. The ETSI TC ITS architecture assumes that each vehicle maintains a data structure holding information about vehicles in its vicinity called the location table [10]. An entry in the location table is allocated to a vehicle's neighbor and contains the following fields: (i) a vehicle identification (ID), (ii) a timestamp of the geographical position, and (iii) a position vector (latitude, longitude, velocity, and the heading (direction)). The location table is periodically updated based on the information included in the received beacons. An entry is considered fresh as long as a new message is received from the vehicle's neighbor in the last N seconds. N is a lifetime timer related to each entry in the location table. When N expires, the entry is considered as deprecated and is consequently deleted.

3.3. The Clustering Algorithm. The objective is to design a clustering algorithm based on a predefined road map and enabling lightweight location updating. Roads are logically regarded as divided into consecutive fixed-sized fragments as shown in Figure 3. Vehicles which are currently within a given road fragment form a cluster. We interchangeably use the word fragment or cluster to refer to either the road fragment or the vehicles that are currently located in it. The length or the size of a cluster is predefined and the width is set to that of the road. For simplicity, we assume that an RSU is placed at each intersection (i.e., at the road section boundary); the penetration rate of RSUs in the urban area will also be investigated. We define a *road segment* as the distance between two given intersections equipped with RSUs. According to the urban area map, a road segment might be covered by single or several clusters. Furthermore, each vehicle retrieves its coordinates and speed using an embedded GNSS receiver (such as the Global Positioning System (GPS)). Each vehicle is equipped with a digital map augmented with the list of the urban area cluster centers.

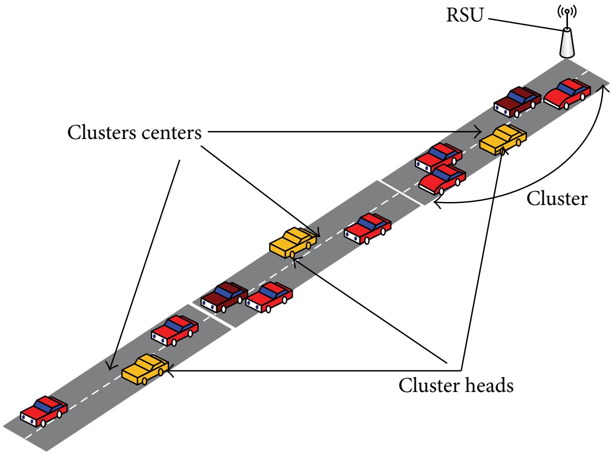


Figure 3: Clustering architecture.

The current nearest vehicle to the cluster center is dynamically self-appointed as the current cluster head and becomes responsible for collecting and transmitting location update messages to its nearest RSU. As such, the selection (or more exactly the self-appointment) of cluster heads is totally dynamic and distributed with no signaling overhead whatsoever. In a dense cluster, more than one cluster head could be self-appointed. This redundancy is beneficial in a wireless environment and does not impact negatively the operation and the efficiency of the updating mechanism as will be investigated and showed later throughout the performance section.

Cluster heads in the range of their nearest RSU directly send their update messages. However, for a long road segment covered by multiple clusters, a given cluster head may be out of the reach of its corresponding RSU. In this case, we propose to use a greedy forwarding mechanism to send location update messages from this cluster head to its RSU via one or several forwarders [41]. A forwarder is a neighbor vehicle that is the nearest to the RSU. It is not mandatory that the forwarder is itself a cluster head. As greedy forwarding is applied on road segments between 2 intersections, we do not need any additional complex mechanism like the left-hand rule to reach the RSU (Figure 5).

Our solution does not require any additional signaling message to designate cluster heads. The cluster head selection process in the proposed clustering protocol has 5 steps as

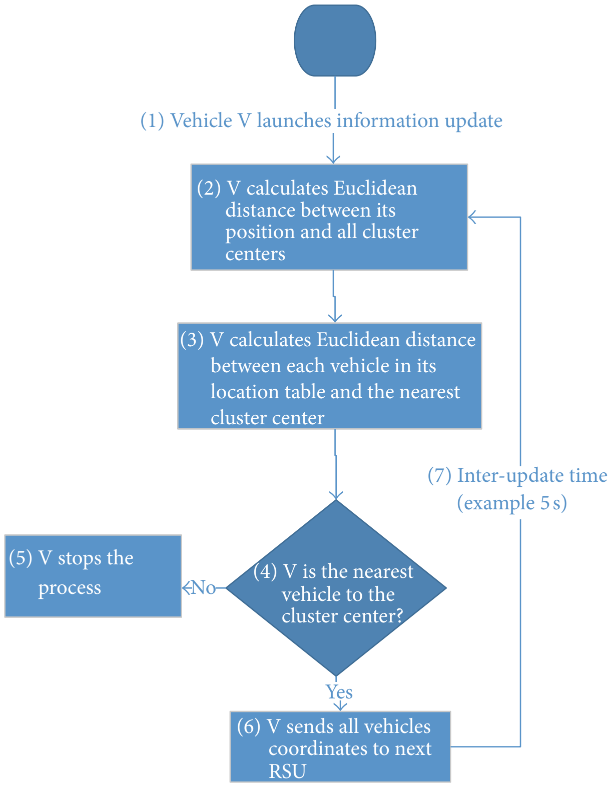

(1) Vehicle V launches information update
(2) V calculates Euclidean distance between its position and all cluster centers
(3) V calculates Euclidean distance between each vehicle in its location table and the nearest cluster center
(4) V is the nearest vehicle to the cluster center?
No
(5) V stops the process
Yes
(6) V sends all vehicles coordinates to next RSU
(7) Inter-update time (example 5 s)

Figure 4: Clustering protocol.

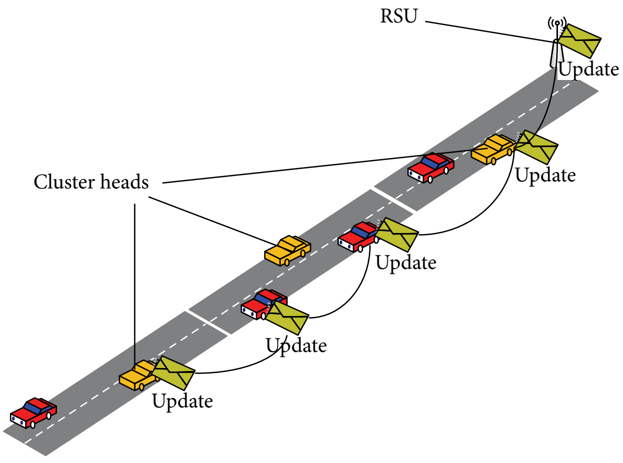


Figure 5: Greedy forwarding.

sketched on Figure 4: (1) every inter-update time (the delay in sending two consecutive information updates to the RSU) each vehicle starts the update procedure. (2) It checks if it is the nearest vehicle to the cluster center to which it belongs. It gets the nearest cluster center by calculating the Euclidean distance between its position and the positions of its surrounding cluster centers provided by its digital map. (3) Then, it calculates the Euclidean distance between each vehicle in its location table (including itself) and this nearest cluster center. (4) If and only if it is the nearest vehicle, (5) it becomes a cluster head and then sends to its nearest RSU an updating message which contains its own location information as well as the locations of all its neighbors registered in its location table.

3.4. Location Update Mechanism. A vehicle obtains its location information using a GNSS device. It broadcasts periodically a beacon for 1 hop. The designated cluster head sends periodically all coordinates of the vehicles in its location table to the nearest RSU. Finally each RSU sends the received data to the regional location server (RLS).

3.5. Location Query Mechanism. A location service query is composed of a location service request (LS request) and a location service reply (LS reply). An LS request is sent by a source vehicle to inquire for destination coordinates. An LS reply is then sent in response to this LS request. Depending on the positions of source and destination vehicles, we can distinguish three different use cases. Figure 6 summarizes these use cases.

(a) Use Case 1: Source and Destination Vehicles Belong to the Same Cluster. If source has in its location table the destination coordinates, the LS request needs not to be sent and the source vehicle can directly communicate with the destination.

(b) Use Case 2: Source and Destination Vehicles Are Not in the Same Cluster but in the Same Region. The source vehicle starts the location service procedure. (1) It sends an LS request to its cluster head. (2) The LS request is forwarded in a greedy way until it reaches the next RSU. (3) If the RSU location table includes the destination coordinates, (4) then the RSU directly sends the LS reply to the source vehicle. (5) Otherwise, the RSU forwards the LS request to the regional location server. (7) As condition 6 is already verified in this use case, the latter sends an LS reply using the reverse path towards the source vehicle.

(c) Use Case 3: Source and Destination Vehicles Are in Different Regions. In this use case, the regional location server does not include the destination coordinates. (8) Thus, it inquires other RLSs through the wired backbone connecting the different regions. (10) An LS reply will be forwarded to the source vehicle through its regional location server along the reverse path. (11) If the destination vehicle information does not exist in any RLS, the LS request is simply deleted.

3.6. Disconnected Networks. Disconnected networks are not a rare use case in vehicular networks. The density of vehicles changes significantly during the day and depends on many factors. As a result, some clusters might become totally empty. Moreover, only a proportion of circulating vehicles are indeed equipped with wireless communication devices. Though it is expected that this proportion gets much bigger in the near future, we need to take care of disconnected networks and investigate the impact of the penetration rate of vehicles equipped with wireless devices on the performance of our proposed location service mechanism.

To the best of our knowledge, proposed location service architectures, such as the ones presented in the related work

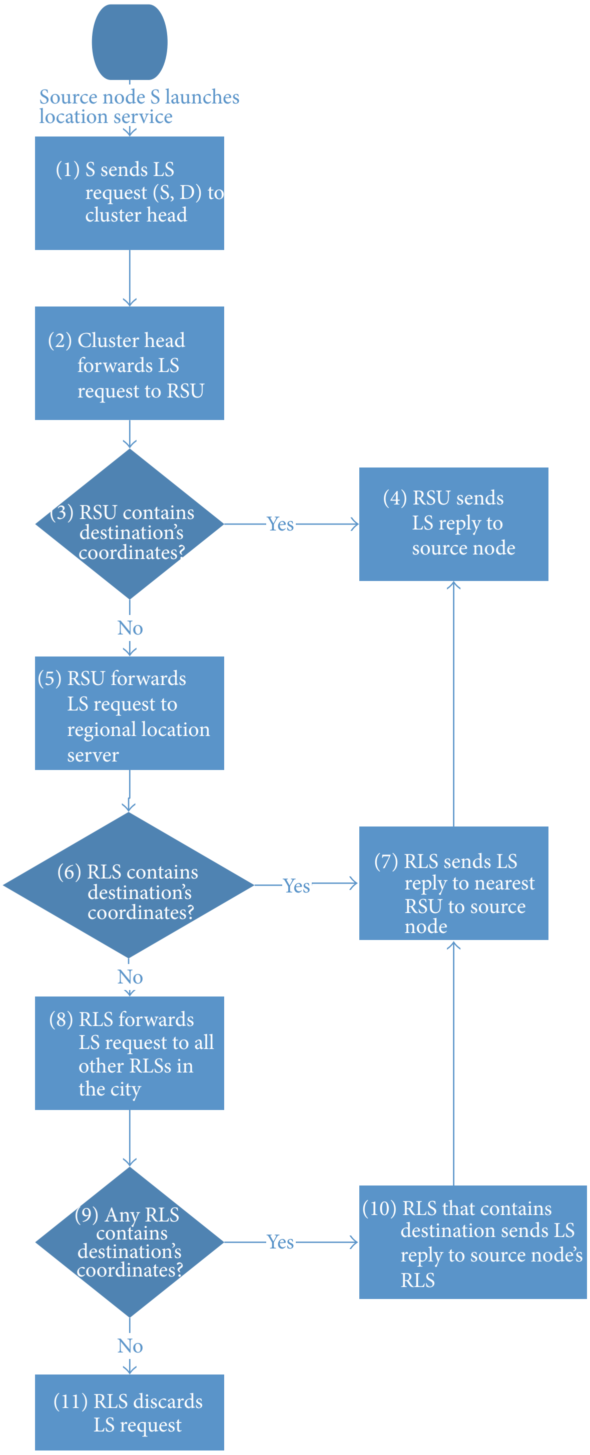

Source node S launches location service
(1) S sends LS request (S, D) to cluster head
(2) Cluster head forwards LS request to RSU
(3) RSU contains destination's coordinates?
Yes
(4) RSU sends LS reply to source node
No
(5) RSU forwards LS request to regional location server
(6) RLS contains destination's coordinates?
Yes
(7) RLS sends LS reply to nearest RSU to source node
No
(8) RLS forwards LS request to all other RLSs in the city
(9) Any RLS contains destination's coordinates?
Yes
(10) RLS that contains destination sends LS reply to source node's RLS
No
(11) RLS discards LS request

Figure 6: Location service request.

in Section 2, do not support disconnected networks. Vehicles simply drop their messages if they do not find any forwarder. Our solution still works even in the case of disconnected networks. When a vehicle crosses an empty cluster, firstly it becomes a forwarder from the greedy perspective and secondly it stores the messages until it reaches another forwarder or the destination RSU itself.

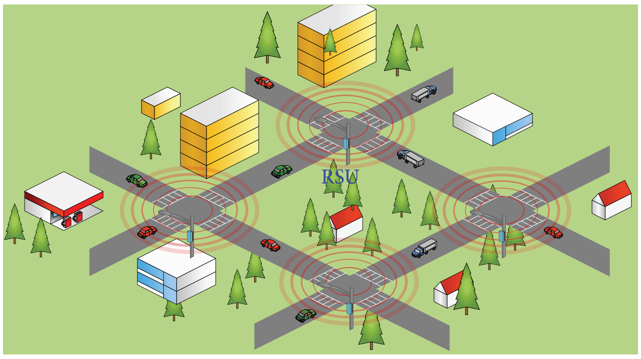


Figure 7: Map.

4. Performance Evaluation

In order to assess the performance of our solution, first of all we evaluate the clustering algorithm parameters: the cluster size and the interlocation update time. Then, we focus on the performance of our solution under a reduced number of RSUs (i.e., not all the intersections are equipped with RSUs). This study targets highlighting the suitability of our solution for a low cost deployment, and/or a stepwise penetration ratio of the wave technology. Finally, we evaluate the whole solution under a real city scenario.

4.1. Clustering Algorithm Parameters. We here target identifying an adequate cluster size and inter-update time. These two parameters are fundamental to the remaining performance evaluation of our location service solution.

4.1.1. Simulation Setup. The region is 3 Km × 3 Km. The subsegment size is 1 Km, resulting in a total of 24 subsegments with a total of 12 Km of two-way roads (1 lane per direction) (as shown in Figure 7). There are a total of 4 intersections. We set vehicle's speed randomly between 10 and 20 m/s. We choose the vehicle interspace randomly between 50 m and 125 m. We set the vehicle wireless communication ange to 400 m. The propagation loss model is "LogDistancePropagationLossModel". We set the beacon frequency at 10 Hz. We use 1 RSU at each intersection. The simulation duration is 120 seconds. In each scenario, we run as many trials as needed to reach a 95% confidence interval at $\varepsilon = 1\%$ of the average value. In our simulation, the update starts from the 5th second. Table 1 summarizes the different simulation parameters.

4.1.2. Metrics. In the following simulations, we consider the following metrics:

(i) The fraction of vehicles saved in the RSUs: it is the number of vehicles saved in the location tables of all RSUs divided by the real number of vehicles in the map.

(ii) The overhead: it is the average size of updating messages transmitted by vehicles.

(iii) The success rate: it is the average number of messages that have been successfully received at RSUs divided

Table 1: Simulation parameters.

Map dimensions	3 Km × 3 Km
Road width	4 m (1 lane per direction)
Speed	Randomly between 10 and 20 m/s
Vehicle interspace	Randomly between 50 m and 125 m
Vehicle radio range	400 m
Propagation loss model	LogDistancePropagationLossModel ReferenceLoss: 46.8588 Exponent: 3
Road Side Unit position	4 RSUs at intersections
Simulation duration	120 s
Confidence interval	95%
Beacon frequency	10 Hz
Market penetration rate	100%

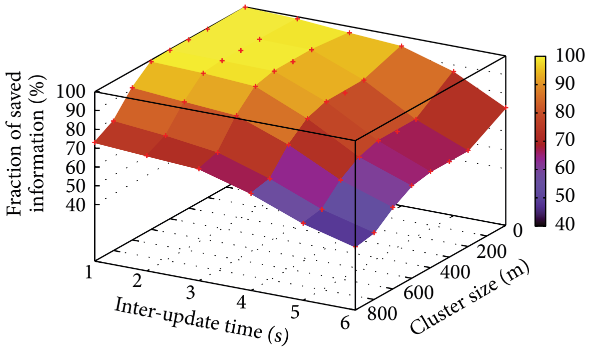


Figure 8: Fraction of vehicles saved in the RSUs (%).

by the number of messages that have been initially triggered from the source.

(iv) Localization error: it is the difference between the vehicle position saved in RSU and its real position at an instant t (in meters).

4.1.3. Results. We performed several simulations by varying the cluster dimension from 200 m to 800 m and the inter-update time from 1 s to 6 s. We target getting the best overview (meaning the highest fraction possible where ultimately all nodes in the road are observed by the RSUs) with the lowest overhead possible. We have to find two operational parameters (cluster size and inter-update time) to fulfill our objective. We compare our contribution to HLS with a regular updating [23] where all vehicles are updated directly and individually to the nearest RSU (i.e., without clustering). Our aim here is twofold. First, we evaluate the improvement brought by clustering. Second, we ascertain the values of our two operational parameters: the cluster size and the inter-update time.

Figure 8 portrays a three-dimensional plot involving the inter-update time, the fraction of the vehicles saved in RSUs (percentage), and the cluster size. A cluster size equal to zero refers to the regular updating that is without clustering. We observe that the fraction of the vehicles saved in RSUs decreases as we increase the inter-update time and the cluster size. We notice that, for an inter-update time ranging from 1 s to 3 s, the fraction of the vehicles saved in RSUs is almost 100% for a cluster size ranging from 200 m to 500 m (details are gathered in Table 2).

Table 2: Fraction (%).

Cluster size (m)	Inter-update time (s)		
	1	3	6
0	98.9078	93.7049	66.5243
200	98.4847	95.9146	55.2631
400	98.5066	95.8128	55.1184
800	72.084	66.4478	37.6608

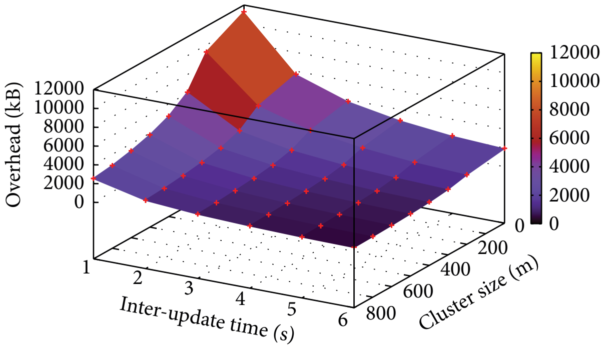


Figure 9: Signaling overhead.

Figure 9 portrays a three-dimensional plot involving the induced overhead (i.e., the signaling overhead), the inter-update time, and the cluster size. We observe that the overhead decreases whenever we increase both the inter-update time and the cluster size. For a small inter-update time less than or equal to 2 s, we observe a signaling storm (about 9200 KB for an inter-update equal to 1 s and a cluster size equal to 200 m). However, for an inter-update time equal to 3 s, the overhead experiences a significant decrease (overhead is only about 1500 KB for a cluster size equal to 400 m).

Back to Figure 8, we see that the best fraction of the vehicles saved in RSUs is obtained when using an inter-update time equal to 1 s, unfortunately at the expenses of a very high overhead as shown in Figure 9. For 3 s, the fraction of the vehicles saved in RSUs is still near 100% (about 96%) while the overhead as provided by Figure 9 is very acceptable. This gives advantage to an inter-update time equal to 3 s as we get less signaling overhead and we still have a high fraction of the vehicles saved in RSUs. We may conclude that, for such an environment, an inter-update time equal to 3 s is a good tradeoff between the fraction of the vehicles saved in RSUs and the resulting overhead. Starting from an inter-update time equal to 4 s, we observe a sharp drop in the surface on Figure 8 even for a small cluster size.

An inter-update time equal to 3 s (see Figure 8 and Table 2) and a cluster size of 400 m provide an accurate and precise overview of the current traffic (a fraction equals 95.81%) which is better than that of the regular updating (93.70%). Although the difference between these two values

Table 3: Comparison of success rate.

IUT (s)	CS (m)							
	0	200	300	400	500	600	700	800
1	80.96	95.17	97.13	98.16	99.17	99.14	99.14	99.2
2	81.57	95.28	96.75	98.34	99.21	99.23	99.23	99.28
3	83.23	95.3	97.33	98.78	99.57	99.45	99.38	99.54
4	85.49	95.45	96.13	99.94	99.59	99.47	99.59	99.78
5	87.90	95.75	97.39	99.54	99.65	99.49	99.62	99.89
6	89.59	95.95	96.75	99.89	99.78	99.50	99.8	99.99

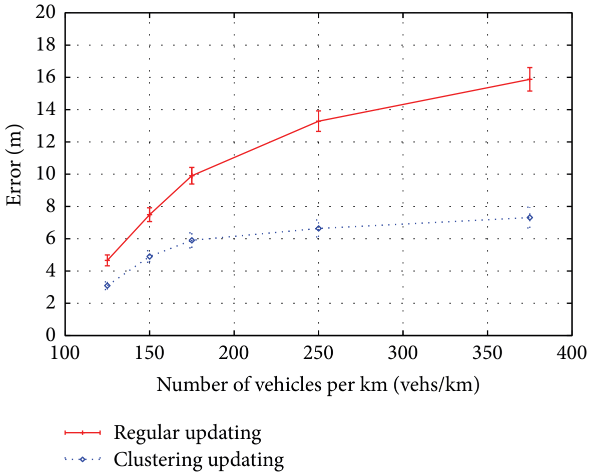


Figure 10: Vehicles localization error for inter-update time equal to 3 s.

is not that much, the overhead of our proposed strategy is 4 times less than that of the regular strategy, which is indeed an important improvement.

Table 3 shows the success rate values obtained when varying the inter-update time (IUT) and the cluster size (CS). The first column of this table refers to the regular updating as the cluster size is put to zero. The proposed updating solution using an inter-update time of 3 s and a cluster size of 400 m yields a success rate of about 99% which is much greater than that obtained using the regular updating (81%). Indeed as explicated in Table 3, the regular updating is much less efficient than our proposed updating scheme for all studied values of the inter-update time and the cluster size.

Now, we turn to the localization error using the simulation parameters reported in Table 1 and our proposed inter-update time of 3 s and cluster size of 400 m. Figure 10 plots the localization error as a function of the network density (the number of vehicles per km) for both of our proposed updating and the regular updating. We observe that our proposal provides a much smaller localization error, almost half of that of the regular updating. For instance, for a density of 400 vehicles per km, our proposal yields an error just underneath the 8 meters while that of the regular updating reaches 16 meters. Besides, the slope of the curve relating to our proposal is smaller which indicates a better scalability for higher densities.

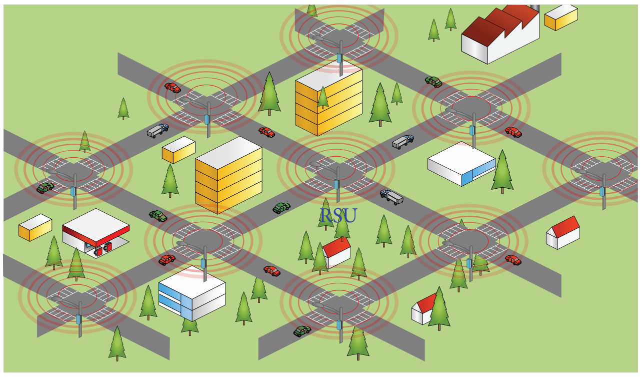


Figure 11: Map.

4.2. RSU Penetration Rate Impact. Our solution relies on two levels of localization update aggregations: RSUs and RLSs. In the best-case scenario, an RSU is deployed at every road intersection. However, to reduce the deployment costs of our solution and to consider the current state of RSU penetration even in different smart cities around the world, we can envisage that only a subset of road intersections are equipped with RSUs. In the following, we investigate the impact of the RSU penetration rate on the overhead, the success rate, the fraction of vehicles saved in location servers, and the vehicle localization error. We use the same simulation parameters described above in Section 4.1.1 but with a map of around 4 Km × 4 Km comprising rather 10 intersections, as shown in Figure 11, and 150 vehicles circulating at 20 m/s. The adopted values of the inter-updating time and the cluster size are those found earlier, that is, 3 s and 400 m, respectively.

4.2.1. Metrics. In the following simulations, we consider the following metrics:

(i) Number of hops: it is the mean number of hops crossed by the updating messages from the sender vehicle to its corresponding RSU.

(ii) The normalized overhead: it is the measured overhead (the cumulative size of all updating messages emanating from all cluster heads) divided by that of the best-case scenario overhead (i.e., 100% of road intersection equipped with RSUs).

(iii) The success rate: the average number of updating messages that have been successfully received at all RSUs over the number of updating messages that have been initially triggered from the source.

(iv) The fraction of vehicles saved in the RSUs: it is the number of vehicles saved in the location tables of all RSUs divided by the real number of vehicles in the map.

(v) Localization error: the difference between the vehicle position saved in RSU and its real position at instant t in meters.

4.2.2. Number of Hops. Figure 12 portrays the average number of hops crossed by the updating messages from the

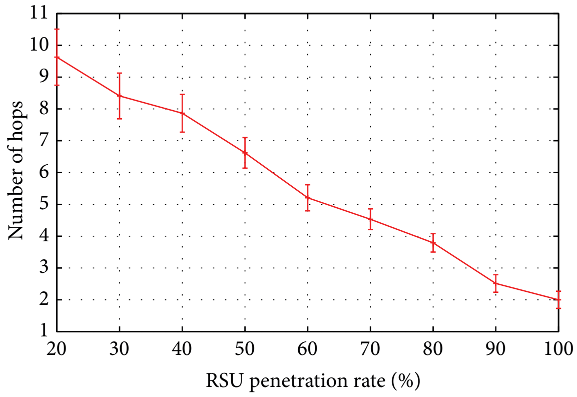


Figure 12: Number of hops.

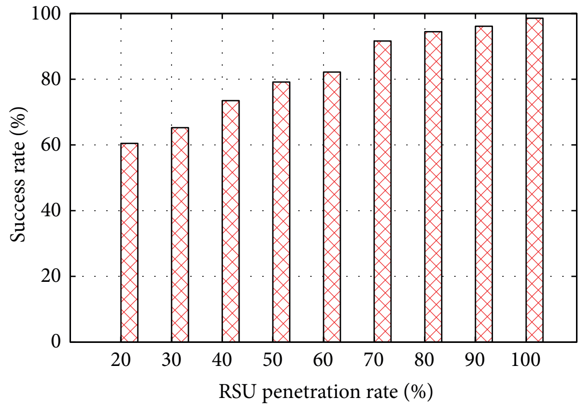


Figure 14: Success rate.

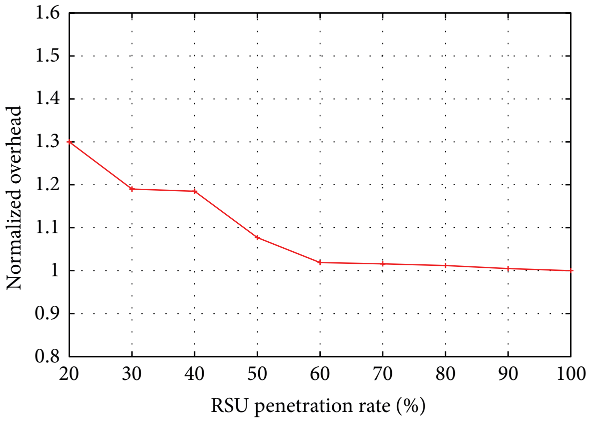


Figure 13: Normalized overhead.

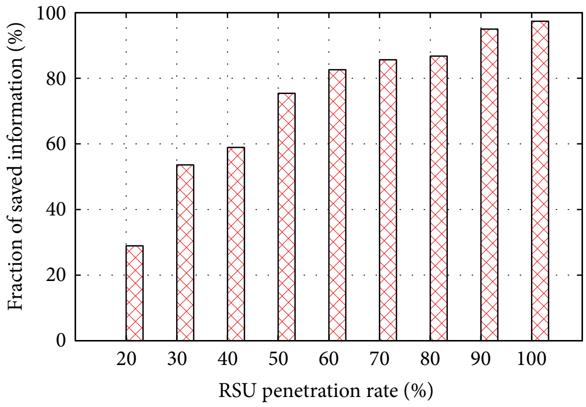


Figure 15: Fraction of vehicles information saved in the RSUs (%).

sender vehicle to its corresponding RSU as a function of the RSU penetration rate that varies from 20% to 100%. For a penetration rate of 100%, the average number of hops is equal to 2 hops. For the lowest penetration rate of 20%, the average number of hops increases to 10 hops.

4.2.3. Normalized Overhead. Figure 13 depicts the normalized overhead evolution as a function of the RSUs penetration rate. We observe that the normalized overhead approaches its minimum value as the penetration rate gets larger than 60%. For a penetration rate equal to 30% the normalized overhead is 1.19, and for the lowest penetration rate (20%), the normalized overhead reaches 1.3. For low penetration rates (from 20% to 50%), localization update messages need to be forwarded through more hops to reach their corresponding RSUs (as showed in Section 4.2.2).

4.2.4. Success Rate. Figure 14 portrays the success rate of updating messages as a function of the RSU penetration rate. For a full penetration rate of 100%, the success rate is equal to 99%. For a penetration rate higher than 60%, the measured success rate still exceeds 82%. For the lowest penetration rate (20%), the measured success rate decreases to 60%. The decrease in the success rate for low penetration rates from 50% to 20% emanates essentially from the increase in the overhead (Figure 13) and the number of hops (Figure 12) to be traversed.

4.2.5. Fraction of Vehicles Saved in the RSUs. Figure 15 shows the fraction of vehicles observed (saved by) by the RSUs while varying the RSU penetration rate from 20% to 100%. For a penetration rate equal to 100%, 97% of vehicles coordinates are known. For a penetration rate higher than 60%, the measured fraction still exceeds 82%. However, when we decrease the penetration rate to 30%, the fraction of vehicles saved in the RSUs becomes 54%, and it falls to 29% for the lowest penetration rate of 20%. This is mainly caused by the tight relationship between the fraction of vehicles saved in the location server and the success rate of updating message. As explained in Section 4.2.4, the success rate of updating messages drops while decreasing the RSU penetration rate.

4.2.6. Localization Error. Figure 16 portrays the localization error as a function of the penetration rate of RSUs. For a full RSU penetration rate (100%), the measured localization error is around 5 m. For the lowest penetration rate (20%), it increases to 17 m. The localization error stays less than 8 m for the entire range of penetration rates varying from 60% to 100%. For low penetration rate, the localization error increase

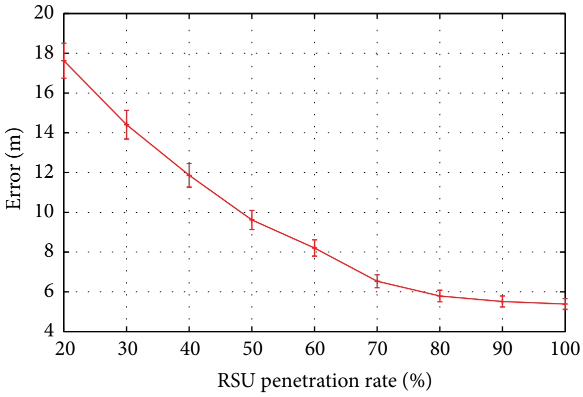


Figure 16: Localization error.

is the direct result of the absence of updating information. An outdated information leads to a high localization error.

4.2.7. Conclusion. The above analysis of the penetration rate of RSUs and its impact on the considered performance metrics clearly show that our proposed Hybrid Cluster-Based Location Service (HCBLS) delivers good results even when the penetration rate decreases to 60%. Indeed, for a penetration rate higher than 60%, the average overhead is almost equal to that obtained by a full penetration rate, the measured success rate exceeds 82%, the percentage of vehicles stored in the RSUs exceeds 82%, and the localization error is lower than 8 m.

4.3. Doha Scenario. We now evaluate our proposed HCBLS in a real city environment using the clustering parameters assessed in Section 4.1 (cluster size = 400 m and inter-update time = 3 s). We compare (HCBLS) with the Efficient Map-Based Location Service (EMBLS) [22] and the Hybrid Location Service (HLS).

Recall that HLS is a variant of our solution where the location update mechanism is performed without clustering. We use IEEE 802.11p and LTE as communication technologies. IEEE 802.11p is used in both V2V and V2I communications, while LTE is used between the RSUs and the regional location server. Recall that EMBLS uses only 802.11p for V2V, V2I, and I2I as its RLS is just one of the vehicles in the system.

We use a portion of 5 Km × 5 Km from the map of Doha city (the capital of Qatar (http://en.wikipedia.org/wiki/Doha)). All streets are two ways with one lane per direction. Vehicle speeds are randomly set between 10 and 35 m/s. We vary the number of simulated vehicles between 150 and 600. We set the vehicle radio range to 400 m and the beacon frequency to 10 Hz (10 beacons per second). We deploy 13 RSUs (1 RSU at each intersection or roundabout) and 1 regional location server (in the middle of the map) as shown in Figure 17 and its SUMO graph in Figure 18.

The simulation duration is 120 seconds. We run as many simulation trials as needed to reach a 95% confidence interval at $\varepsilon = 1\%$ of the average value. In our simulation, the location update starts from the 5th second. Table 4 summarizes the different simulation parameters.

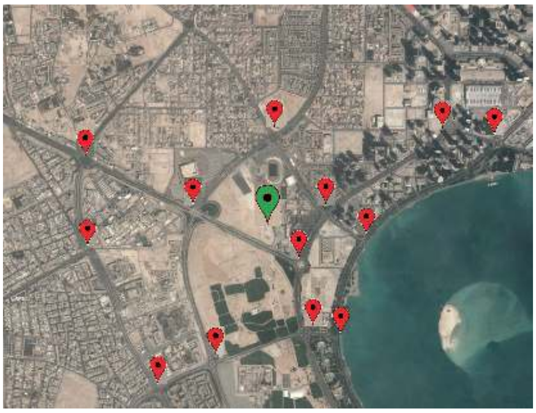


Figure 17: Doha Map.

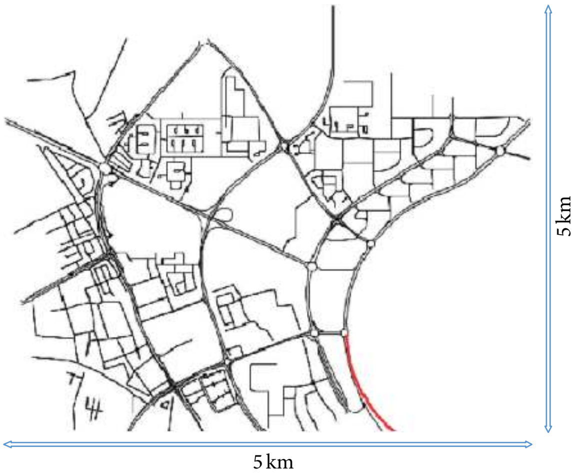


Figure 18: Doha Map with SUMO.

Metrics. In the following simulations, we focus on the following metrics:

(i) Success rate: it is the average number of messages that have been successfully received at all RSUs and the location server over the number of messages that have been initially sent from sources.

(ii) Fraction of vehicles saved in the location server: it is the number of vehicles saved in the location table of the location server divided by the real number of vehicles in the map.

(iii) Overhead: it is the total size of messages generated by both the location update and the location query processes in KB.

(iv) Message delivery latency: it is the average delay spent by location queries and geo-unicast messages to reach their destinations in ms.

(v) Localization error: it is the difference between the vehicle position saved in the location server and its real position at an instant t in meters.

Table 4: Simulation parameters.

Simulation time (s)	120 s
Map dimensions	5 Km × 5 Km
Road width	1 lane (4 m) in both directions
Number of vehicles	150, 200, 300, and 600
Speed (m/s)	Random 10, 15, 20, 25, 30, and 35
Vehicle radio range	400 m
Propagation loss model	LogDistancePropagationLossModel ReferenceLoss: 46.8588 Exponent: 3
Number of RSUs	13 RSUs
Number of location servers	1 location server
Confidence interval	95%
Beacon frequency	10 Hz
MAC layer protocol	IEEE 802.11p and LTE
Market penetration rate	100%

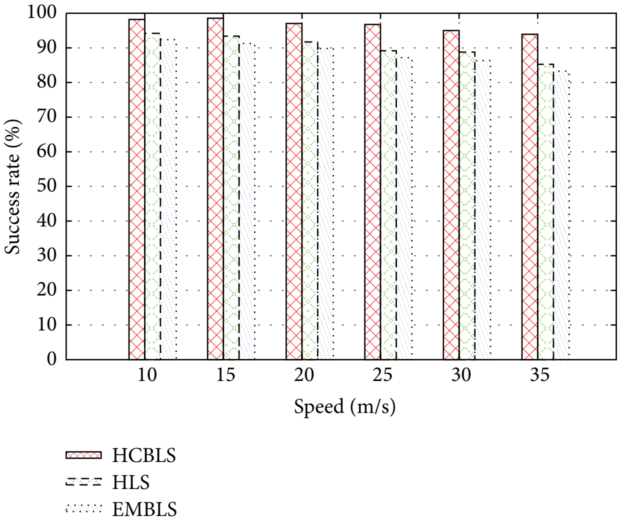


Figure 19: Comparison of success rate for 150 simulated vehicles.

4.3.1. Success Rate. Figure 19 portrays the success rate as a function of the vehicle speed for the three different considered location service mechanisms, namely, HCBLS, HLS, and EMBLS. For the highest simulated speed (35 m/s), the success rate of HCBLS is equal to 94%, while the success rate of both HLS and EMBLS is, respectively, equal to 85% and 83%. For the lowest simulated speed (10 m/s), the success rate of HCBLS is equal to 98%, while the success rate of both HLS and EMBLS is, respectively, equal to 94% and 92%. We can make here three observations. Firstly, the decrease in the success rate of all three mechanisms as the speed gets higher is very moderate, which is due essentially to the adopted frequency of beaconing of 10 beacons per second. Secondly, the success rate of HCBLS outperforms those of HLS and EMBLS. Thirdly, the success rate provided by our proposed HCBLS for the largest speed (94%) is even greater than those provided by EMBLS (92%) and HLS (94%) for the lowest speed. This later observation states that our proposed HCBLS is efficient even for high speed.

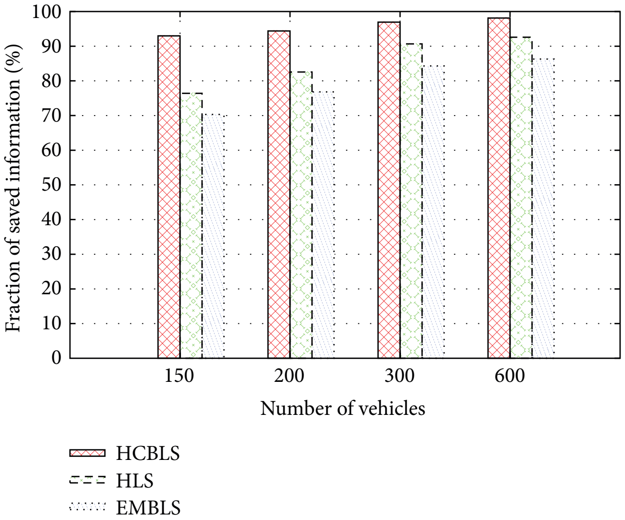


Figure 20: Fraction of vehicles information saved in the regional location server (%).

Recall that HCBLS is a cluster-based location service while HLS is not, but both use two distinct access technologies to send location updates. The superiority of HCBLS emanates from using clustering. For the highest simulated speed, the clustering increases the success rate by 11%, whereas it increases by 4% for the lowest simulated speed. The effect and benefit of clustering increase with the speed. On the other hand, we observe from Figure 19 that HLS outperforms EMBLS for all considered speeds. Recall that the latter uses only one access technology, the IEEE 802.11p. Using two appropriate access technologies (HCBLS and HLS) outperforms the use of a single technology which might not be appropriate for I2I communications.

4.3.2. Fraction of Vehicles Saved in the Regional Location Server. Figure 20 plots the fraction of the vehicles saved in the regional location server (percentage) as a function of the number of circulating vehicles, for the three mechanisms. We first observe that this fraction increases as the number of simulated vehicles increases. HCBLS has the highest fraction and therefore gives always the best overview of the circulating vehicles in the map.

For the lowest simulated number of vehicles (150 vehicles), the measured fraction for HCBLS is equal to 93%, while for both HLS and EMBLS it is, respectively, equal to 76% and 70%. Our HCBLS achieves then an improvement of 22% compared to HLS and 33% compared to EMBLS. Recall here that HCBLS takes into account the case of disconnected networks in its message forwarding. For the highest simulated number of vehicles (600 vehicles), the measured fraction for HCBLS is equal to 98%, while that of HLS and EMBLS is, respectively, equal to 93% and 86%. Our HCBLS achieves then an improvement of 5% compared to HLS and 14% compared to EMBLS.

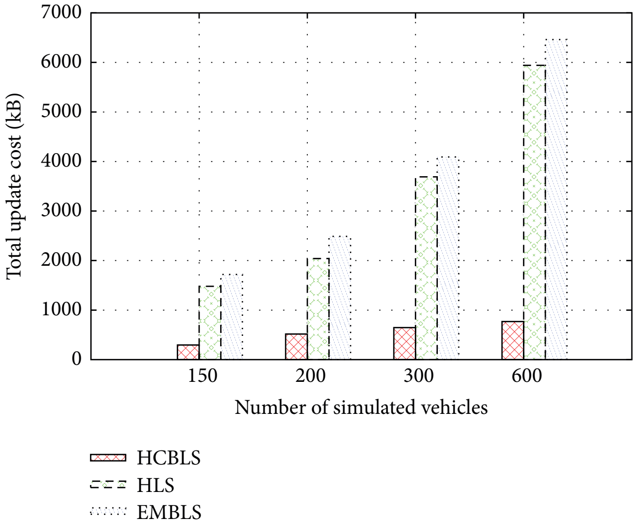


FIGURE 21: Location update cost (Kbytes).

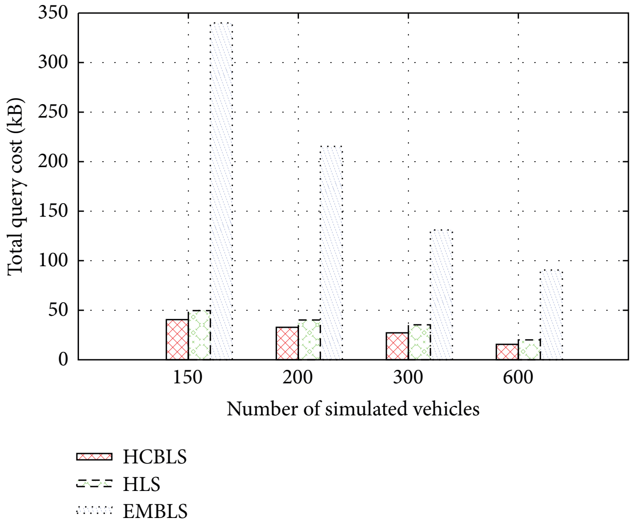


FIGURE 22: Location query cost (Kbytes).

4.3.3. Overhead. The overhead is composed of the cumulative control or signaling messages used by the location service mechanism to update the locations of the circulating vehicles and to query about vehicles locations.

(a) Location Update Cost. Figure 21 depicts the total cost of locations update for different numbers of simulated vehicles. As the number of vehicles increases, the total cost of the location update also increases. Regardless of the simulated number of vehicles, the location update cost of the EMBLS is the highest, followed by that of the HLS and then that of HCBLS.

For the lowest number of simulated vehicles (150), the measured HCBLS location update load is 296 KB, while those of HLS and EMBLS are, respectively, 1480 and 1721 KB. The improvement achieved by HCBLS is very clear as the location update load of HLS and EMBLS is, respectively, 5 times and 6 times higher than that of HCBLS. For the highest number of simulated vehicles (600), the measured HCBLS location update load is 769 KB, while those of HLS and EMBLS are, respectively, 5942 and 6461 KB. The improvement here is also very clear since the location update load of HLS and EMBLS is, respectively, 8 times and 9 times higher than HCBLS.

EMBLS location update load is always the highest as location update messages need to be sent to the location server hop by hop through intermediate vehicles. HLS has a lower location update cost than EMBLS as the location update messages are sent rather by RSUs to the regional location server through LTE. The location update load of HCBLS is the lowest one as only cluster heads update the RSUs with vehicles data.

(b) Location Query Cost. Figure 22 shows the total cost of location query for different numbers of simulated vehicles. The total cost of location queries is a decreasing function of the number of simulated vehicles. This is true since as vehicle density gets higher, the opportunities to get the location information of the queried vehicle from other vehicles get higher. The location query cost of HCBLS is the lowest. While the location query cost of HLS is close to that of HCBLS, and that of EMBLS is far much higher.

For the lowest number of simulated vehicles (150), the measured HCBLS location query load amounts to 40 KB, while those of HLS and EMBLS are, respectively, 50 and 340 KB. The improvement achieved by our proposed HCBLS is very clear as the location query load of EMBLS is roughly 9 times higher. For the highest number of simulated vehicles (600), the measured HCBLS location query load amounts to 15 KB, while those of HLS and EMBLS are, respectively, 21 and 91 KB. The improvement achieved by our proposed HCBLS is also very clear as the location query load of EMBLS is 6 times higher.

For the location query, HLS achieves a slightly larger cost than HCBLS. Vehicles generate more location queries in HLS to get the location information of the queried vehicles because essentially of higher packet loss. The location query cost of the EMBLS scheme is much larger because location query messages need to travel a longer distance (in terms of number of hops). In HCBLS and HLS, we use LTE for the communication between the RSUs and the location server. In EMBLS, this communication is compensated through additional hops to deliver location query messages.

(c) Total Cost of Location Service. Figure 23 portrays the total cost as a function of the number of simulated vehicles. For the lowest number of simulated vehicles (150), the HCBLS total location service load amounts to 337 KB, while those for HLS and EMBLS are, respectively, 1530 and 2061 KB. We clearly observe the improvement achieved by HCBLS as the location service load of HLS and EMBLS is, respectively, 5 times and 7 times higher. For the highest number of simulated vehicles (600), the HCBLS total location service load is only 785 KB, while those for HLS and EMBLS are, respectively, 5962 and 6551 KB. The improvement achieved by HCBLS is very clear as the location service load of HLS and EMBLS is, respectively, 8 times and 9 times higher.

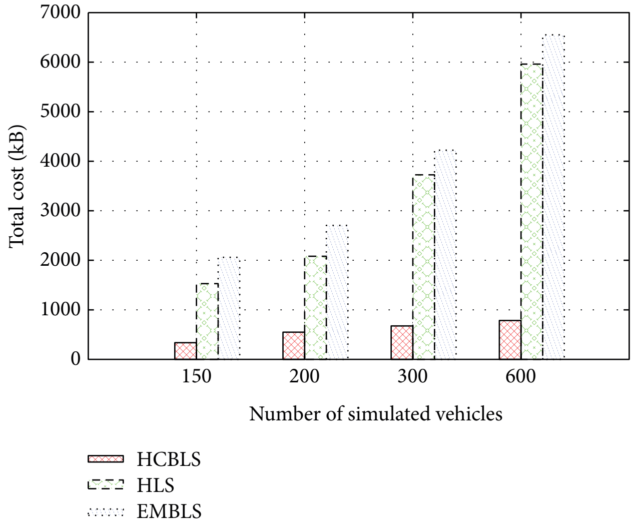


Figure 23: Total cost of location service (Kbytes).

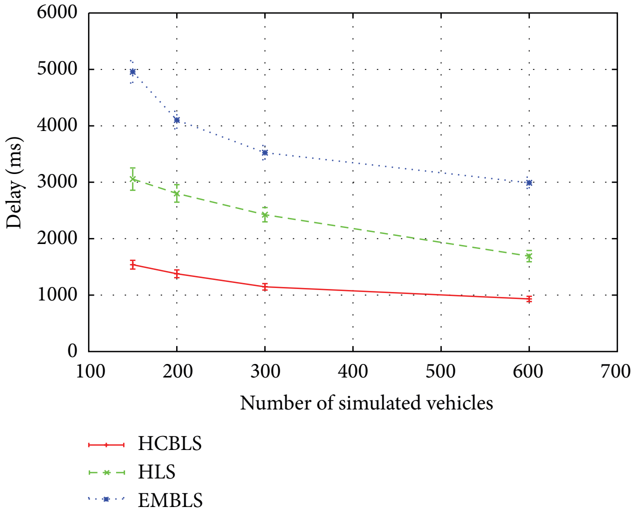


Figure 25: Message delivery latency.

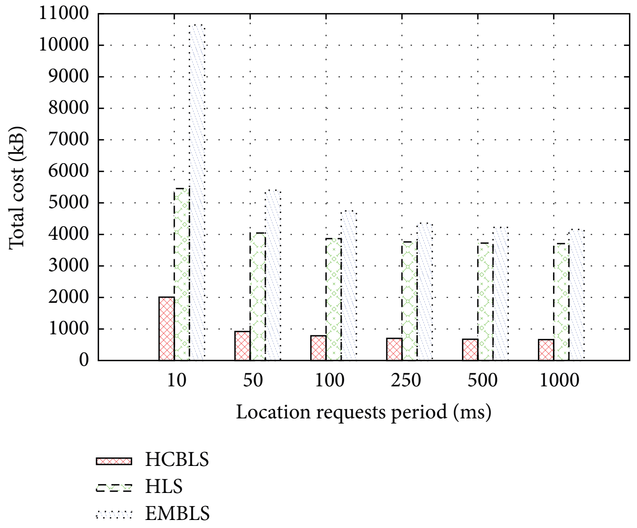


Figure 24: Total signaling cost of location service (Kbytes) as a function of the location requests period τ (ms).

To investigate the efficiency of our proposed HCBLS when several vehicles request locations, we plot on Figure 24 the total signaling cost of location services as a function of the location requests frequency and for the case of 300 vehicles. Let τ be the period in milliseconds of location requests. As such, the number (namely, the frequency) of location requests per second is just $1/\tau$, and these requests emanate from different randomly chosen vehicles and destined each to a randomly chosen destination. For τ equal to 10 ms, the total signaling cost of HCBLS is just 2000 KB while that of EMBLS amounts to 10600 KB which is five times larger. Despite this high frequency of location queries (100 queries per second), HCBLS generates a reasonable overhead compared to EMBLS. More importantly, Figure 24 shows that our proposed HCBLS is scalable as it sustains much better the increase in location requests per second. Indeed when the frequency of location requests increases from one request per second ($\tau = 1000$ milliseconds) to 100 requests per second ($\tau = 10$ milliseconds), the signaling cost of HCBLS increases from around 800 KB to just 2000 KB.

4.3.4. Message Delivery Latency. The message delivery latency is one of the most critical metrics that highlights the accuracy and responsiveness of a location service. A high latency greatly impacts the accuracy of the vehicle localization as received information is outdated, and consequent geo-unicast messages using this outdated information will not reach their final destinations.

Figure 25 portrays the message delivery latency (in ms) as a function of the number of simulated vehicles. Our proposed HCBLS exhibits the lowest message delivery latency for all numbers of simulated vehicles. For the lowest number of simulated vehicles (150), HCBLS message delivery latency amounts to only 1539 ms, while those of HLS and EMBLS are equal to 3056 ms and 4956 ms, respectively. For the highest number of simulated vehicles (600), HCBLS has a message delivery latency equal to 931 ms, while those of HLS and EMBLS are 1690 ms and 2990 ms, respectively. In fact, EMBLS has the highest location query load as investigated in Section 4.3.3(b). In EMBLS, a source vehicle generates more location queries to get a destination vehicle's coordinates.

4.3.5. Localization Error. We finalize our study by showing the efficiency of our proposed HCBLS in terms of its achieved localization error as compared to those of HLS and EMBLS. Figure 26 portrays the localization error as a function of the speed. HCBLS indeed outperforms both HLS and EMBLS over all considered speeds. For the lowest considered speed of 10 m/s, HCBLS achieves a very good localization error of just 6 m while those of HLS and EMBLS amount to 20 m and 24 m, respectively. For the high considered speed of 35 m/s, the localization error achieved by HCBLS is just around 15 m,

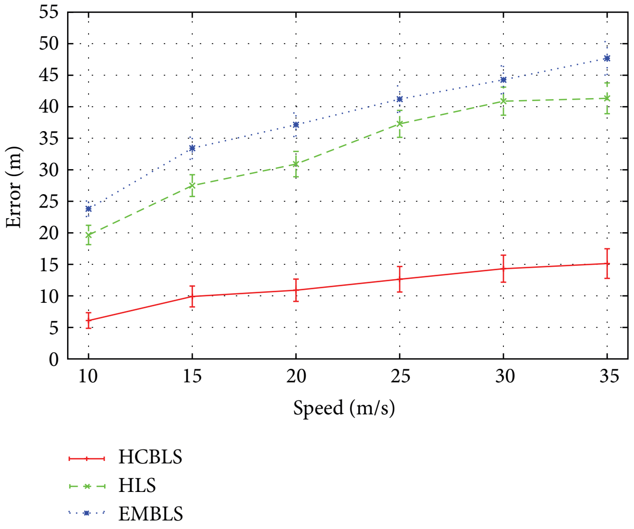


FIGURE 26: Localization error for 150 simulated vehicles.

while those of HLS and EMBLS are, respectively, 42 m and 48 m. Overall, the improvement achieved by our proposed HCBLS is around 4 times better than HLS and EMBLS. Besides, the slope of the curve relating to our proposal is smaller which indicates a much better scalability for higher speeds.

5. Conclusion

The paper presented a lightweight Hierarchical Cluster-Based Location Service (HCBLS). Clustering in HCBLS is performed with no extra signaling overhead as it is based on a logical decomposition of the network based on a preloaded digital map. Location updating is achieved through aggregation at different levels of the assumed hierarchy.

Extensive simulations using different scenarios were conducted on the NS3 version developed by the European research project iTETRIS coupled with the traffic simulator SUMO. Results clearly exhibited the good efficiency of HCBLS as compared to the Enhanced Map-Based Location Service (EMBLS) and any regular (non-cluster-based) updating scheme. HCBLS increased the success rate, improved the overview of the network, lowered the location update and query costs, reduced the message delivery latency, and presented a much greater localization accuracy. Furthermore, HCBLS presented a better scalability which is required for dense networks of city environments. Further investigations are undertaken to enhance HCBLS in terms of reliability and security. Besides, we intend to adapt HCBLS mechanism to get a new geographic routing protocol and compare its performance with existing routing protocols.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through Research Group no. RGP-1436-002.

References

[1] K. Aboltins and B. Rivza, "The car aftersales market development trends in the new economy," *Procedia—Social and Behavioral Sciences*, vol. 110, pp. 341–352, 2014, the 2-dn International Scientific Conference Contemporary Issues in Business, Management and Education 2013.

[2] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular Ad Hoc network," *Journal of Network and Computer Applications*, vol. 37, no. 1, pp. 380–392, 2014.

[3] J. J. P. C. Rodrigues, R. Hsing, M. Chen, B. Jiao, and B. Vaidya, "Guest editorial on vehicular networking protocols," *Journal of Network and Computer Applications*, vol. 36, no. 3, pp. 959–960, 2013.

[4] M. S. Kakkasageri and S. S. Manvi, "Information management in vehicular ad hoc networks: a review," *Journal of Network and Computer Applications*, vol. 39, no. 1, pp. 334–350, 2014.

[5] K. G. Automotive Leaders, "KMPG's global automotive executive survey 2014," 2014, http://goo.gl/8Qpdnj.

[6] G. mAutomotive, "Connected car forecast: global connected car market to grow threefold within five years," 2014, http://goo.gl/mOjrln.

[7] H. Menouar, R. Aissaoui, F. Filali, A. Abu-Dayya, and A. Belghith, "Cellular-based location service to support GeoUnicast communications in vehicular wireless networks," in *Proceedings of the 20th International Conference on Telecommunications (ICT '13)*, pp. 1–5, IEEE, Casablanca, Morocco, May 2013.

[8] R. Aissaoui, H. Menouar, A. Dhraief, F. Filali, A. Belghith, and A. Abu-Dayya, "Advanced real-time traffic monitoring system based on V2X communications," in *Proceedings of the IEEE International Conference on Communication (ICC '14) and Mobile and Wireless Networking Symposium (MWN '14)*, Sydney, Australia, June 2014.

[9] Y. Hu, W. Zhang, Y. Li, and P. Xiong, "The research of WAVE architecture based vehicles to vehicles communication technology of Intelligent Transport System," in *Proceedings of the 2nd International Conference Power Electronics and Intelligent Transportation System (PEITS '09)*, vol. 3, pp. 134–139, Shenzhen, China, December 2009.

[10] ETSI, "Intelligent transport systems (its); vehicular communications; geonetworking; part 4: geographical addressing and forwarding for point-to-point and point-to-multipoint communications; sub-part 1: media-independent functionality," ETSI TS 102 636-4-1 V1.1.1 (2011-06), 2011, http://goo.gl/eyhFUl.

[11] T. Gazdar, A. Benslimane, and A. Belghith, "Secure clustering scheme based keys management in VANETs," in *Proceedings of the IEEE 73rd Vehicular Technology Conference (VTC '11)*, pp. 1–5, Yokohama, Japan, May 2011.

[12] T. Gazdar, A. Rachedi, A. Benslimane, and A. Belghith, "A distributed advanced analytical trust model for VANETs," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '12)*, pp. 201–206, IEEE, Anaheim, Calif, USA, December 2012.

[13] U. D. Gandhi, A. Singh, A. Mukherjee, and A. Chandak, "Smart vehicle connectivity for safety applications," in *Proceedings of*

the International Conference on Reliability, Optimization and Information Technology (ICROIT '14), pp. 262–266, Faridabad, India, February 2014.

[14] M. Bakhouya, J. Gaber, and P. Lorenz, "An adaptive approach for information dissemination in vehicular Ad hoc networks," *Journal of Network and Computer Applications*, vol. 34, no. 6, pp. 1971–1978, 2011.

[15] S. M. Bilal, C. J. Bernardos, and C. Guerrero, "Position-based routing in vehicular networks: a survey," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 685–697, 2013.

[16] S. M. Das, H. Pucha, and Y. C. Hu, "Performance comparison of scalable location services for geographic ad hoc routing," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, vol. 2, pp. 1228–1239, IEEE, Miami, Fla, USA, March 2005.

[17] T. Camp, J. Boleng, and L. Wilcox, "Location information services in mobile ad hoc networks," in *Proceedings of the International Conference on Communications (ICC '02)*, vol. 5, pp. 3318–3324, May 2002.

[18] J. Y. Yu and P. H. J. Chong, "A survey of clustering schemes for mobile ad hoc networks," *IEEE Communications Surveys and Tutorials*, vol. 7, no. 1, pp. 32–47, 2005.

[19] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator," in *Proceedings of the SIGCOMM Demonstration Conference*, Seattle, Wash, USA, August 2008.

[20] V. Kumar, L. Lin, D. Krajzewicz et al., "iTETRIS: adaptation of its technologies for large scale integrated simulation," in *Proceedings of the IEEE 71st Vehicular Technology Conference (VTC Spring '10)*, pp. 1–5, Taipei, Taiwan, May 2010.

[21] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "Sumo (simulation of urban mobility)," in *Proceedings of the 4th Middle East Symposium on Simulation and Modelling*, pp. 183–187, 2002.

[22] D. M. Ashok, M. M. M. Pai, and J. Mouzna, "Efficient map based location service for VANETs," in *Proceedings of the 11th International Conference on ITS Telecommunications (ITST '11)*, pp. 387–392, St. Petersburg, Russia, August 2011.

[23] F. Zhang, J. Hao, and S. Li, "Traffic information aggregation and propagation scheme for VANET in city environment," in *Proceedings of the 3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT '10)*, pp. 619–623, IEEE, Beijing, China, October 2010.

[24] Y. Bae Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," in *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom '98)*, Dallas, Tex, USA, 1998.

[25] S. Oh, J. Kang, and M. Gruteser, "Location-based flooding techniques for vehicular emergency messaging," in *Proceedings of the 3rd Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous '06)*, pp. 1–9, San Jose, Calif, USA, July 2006.

[26] W. Peng and X.-C. Lu, "On the reduction of broadcast redundancy in mobile ad hoc networks," in *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc '00)*, pp. 129–130, IEEE Press, Boston, Mass, USA, August 2000.

[27] D. Liu, I. Stojmenović, and X. Jia, "A scalable quorum based location service in ad hoc and sensor networks," in *Proceedings of the IEEE International Conference on Mobile Ad Hoc and Sensor Sysetems (MASS '06)*, pp. 489–492, Vancouver, Canada, October 2006.

[28] F. Yu, G. Hu, S. Park, E. Lee, and S.-H. Kim, "Quorum based sink location service for irregular wireless sensor networks," *Computer Communications*, vol. 35, no. 12, pp. 1422–1432, 2012.

[29] E. Lee, H. Choe, P. Thirumurthi, M. Gerla, and S.-H. Kim, "Quorum-based location service in vehicular sensor networks," in *Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC '13)*, pp. 1744–1749, IEEE, Cagliari, Sardinia, July 2013.

[30] S.-C. M. Woo and S. Singh, "Scalable routing protocol for ad hoc networks," *Wireless Networks*, vol. 7, no. 5, pp. 513–529, 2001.

[31] H. Saleet, O. Basir, R. Langar, and R. Boutaba, "Region-based location-service-management protocol for VANETs," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 2, pp. 917–931, 2010.

[32] M. Boussedjra, J. Mouzna, P. Bangera, and M. M. M. Pai, "Map-based location service for VANET," in *Proceedings of the International Conference on Ultra Modern Telecommunications & Workshops (ICUMT '09)*, pp. 1–6, IEEE, St. Petersburg, Russia, October 2009.

[33] N. Brahmi, M. Boussedjra, J. Mouzna, A. K. V. Cornelio, and M. M. Manohara, "An improved map-based location service for vehicular ad hoc networks," in *Proceedings of the 6th Annual IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob '10)*, pp. 21–26, Ontario, Canada, October 2010.

[34] X.-Y. Bai, X.-M. Ye, J. Li, and H. Jiang, "VLS: a map-based vehicle location service for city environments," in *Proceedings of the IEEE International Conference on Communications (ICC '09)*, pp. 1–5, Dresden, Germany, June 2009.

[35] C.-S. Hsu and S.-W. Wu, "An efficient cost-based location service protocol for vehicular ad hoc networks," in *Proceedings of the IEEE International Conference on Communication, Networks and Satellite (COMNETSAT '12)*, pp. 93–97, Bali, Indonesia, July 2012.

[36] A. Ibanez, C. Flores, P. Reyes, A. Barba, and A. Reyes, "A performance study of the 802.11p standard for vehicular applications," in *Proceedings of the 7th International Conference on Intelligent Environments (IE '11)*, pp. 165–170, IEEE, Nottingham, UK, July 2011.

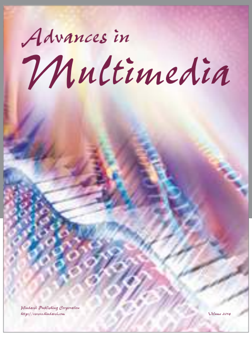
[37] A. Vinel, "3GPP LTE versus IEEE 802.11p/WAVE: which technology is able to support cooperative vehicular safety applications?" *IEEE Wireless Communications Letters*, vol. 1, no. 2, pp. 125–128, 2012.

[38] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "LTE for vehicular networking: a survey," *IEEE Communications Magazine*, vol. 51, no. 5, pp. 148–157, 2013.

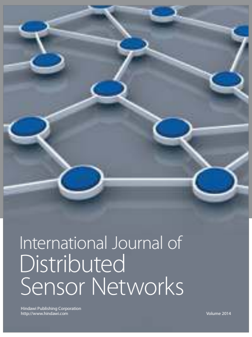
[39] Z. Hameed Mir and F. Filali, "LTE and IEEE 802.11p for vehicular networking: a performance evaluation," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, article 89, 2014.

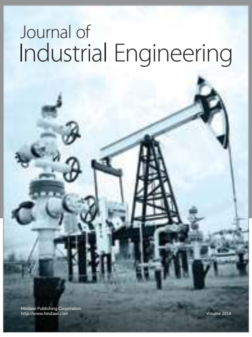
[40] S. Ucar, S. Coleri Ergen, and O. Ozkasap, "Multi-hop cluster based IEEE 802.11p and LTE hybrid architecture for VANET safety message dissemination," *IEEE Transactions on Vehicular Technology*, 2015.

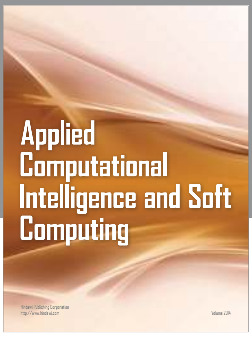
[41] C. Lochert, M. Mauve, H. Füßler, and H. Hartenstein, "Geographic routing in city scenarios," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 1, pp. 69–72, 2005.

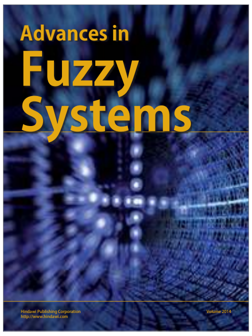

Advances in
Multimedia

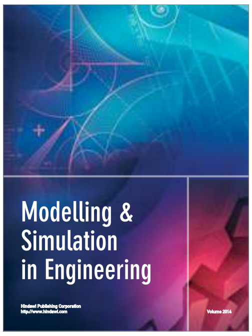
The Scientific
World Journal

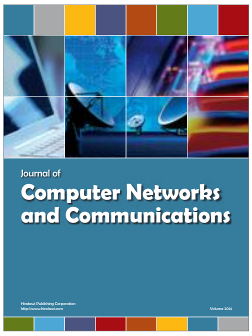

International Journal of
Distributed
Sensor Networks

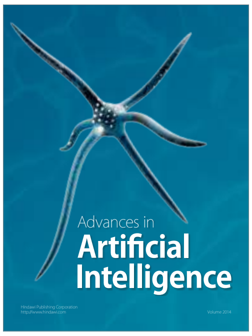

Journal of
Industrial Engineering

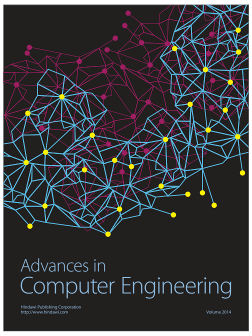

Applied
Computational
Intelligence and Soft
Computing


Advances in
Fuzzy
Systems

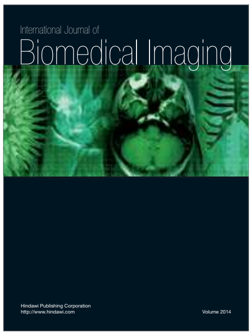

Modelling &
Simulation
in Engineering

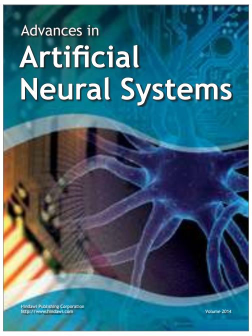

Journal of
Computer Networks
and Communications

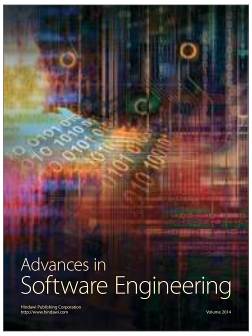

Advances in
Artificial
Intelligence


Advances in
Computer Engineering

International Journal of
Computer Games
Technology

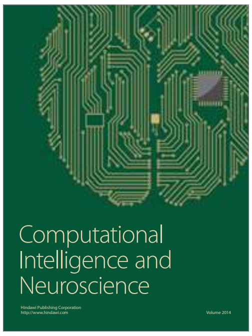

International Journal of
Biomedical Imaging


Advances in
Artificial
Neural Systems


Advances in
Software Engineering

Journal of
Robotics

Advances in
Human-Computer
Interaction


Computational
Intelligence and
Neuroscience

International Journal of
Reconfigurable
Computing

Journal of
Electrical and Computer
Engineering