Region-Based Location-Service-Management Protocol for VANETs

Hanan Saleet, *Student Member, IEEE*, Otman Basir, *Member, IEEE*, Rami Langar, *Member, IEEE*, and Raouf Boutaba, *Senior Member, IEEE*

***Abstract*—The efficiency by which a node of a vehicular ad hoc network (VANET) can route messages to destinations heavily depends on the VANET's ability to keep track of the locations of its nodes (vehicles). Current location-management schemes lack scalability and, hence, are proven unable to work in large-scale networks. Therefore, location management in VANETs remains a major challenge. In this paper, we propose a new region-based location-service-management protocol (RLSMP) that uses mobility patterns as means to synthesize node movement and, thus, can be used in large VANET applications. The protocol attempts to relax the scalability issue suffered by other protocols by employing message aggregation in location updating and in querying. Furthermore, due to the protocol's intrinsic locality awareness, it achieves minimum control overhead. To evaluate the efficiency of the protocol, we study its performance analytically and by using simulation for a 2-D random-walk model, as well as on real mobility patterns. The performance of the protocol is compared with that of other prominent location-management protocols.**

***Index Terms*—Communication overhead, location service, mobility management, performance analysis, vehicular ad hoc networks (VANETs).**

I. Introduction

Vehicular ad hoc networks (VANETs) represent a rapidly emerging and challenging class of mobile ad hoc networks (MANETs). In such networks, each node operates not only as a host but also as a router, forwarding packets for other mobile nodes [1]–[3]. Vehicles form a decentralized communication network by means of wireless multihop routing and forwarding protocols. A wide range of applications can be enabled in VANETs (to name a few, we have emergency message dissemination, real-time traffic condition monitoring, and collision avoidance and safety), where communications are exchanged to improve the driver's responsiveness and safety in case of road incidents [4], [5]. VANETs not only can enhance traffic safety but can also enable infotainment applications via multihop communications between vehicles [6], [7]. As such, to support the different types of applications, the network must be able to efficiently locate its nodes. While moving, mobile nodes send location updates to nodes that are located in specific regions in the network called home regions. These nodes (which are referred to as location servers) are responsible for replying to location queries. An efficient location-service-management protocol is, thus, needed to track the locations of mobile nodes (location updates) and reply to location queries with minimum overhead [8], [9].

Numerous approaches for location service management in MANETs have been reported in the literature [10]–[21]. These approaches fall into two categories, namely, flooding-based and quorum-based approaches. Flooding-based approaches involve global network flooding [14], [15]. Such flooding requires all the mobile nodes of the network to send periodic updates or queries to all the other nodes, which results in unbounded control overhead. As such, these approaches are not suitable in large-size ad hoc networks such as VANETs, and their use is limited to small-size networks with slow-moving nodes [16]. In quorum-based approaches, location servers, which are also called quorums, are assigned with their role by mapping the geographical information of the nodes to quorums in a random or static way [17]–[21]. Quorum-based approaches assume that each node knows its own location using, for example, the Global Positioning System (GPS). They are relatively more scalable than the flooding-based approaches, particularly in highly dynamic ad hoc networks [22]. However, their performance is dependent on the existence of an efficient location-service-management scheme.

In this paper, we capitalize on quorum-based approaches to achieve efficient location service management by means of node clustering and message aggregation. Thus, the updates and the queries of nearby nodes are aggregated in one control message. Distinct groups of nodes called clusters are formed based on their geographical locations (i.e., nodes that are located in one geographical area are grouped in one cluster). In addition, since the communication patterns in VANETs are considered, to a great extent, to be local (i.e., vehicles inside one geographical cluster are more likely to communicate with each other), the proposed service-management protocol should be locality-aware. Consequently, location servers are assigned while considering the local traffic patterns. We formulate this strategy in the proposed region-based location-service-management protocol (RLSMP).

To study the effectiveness of the proposed protocol, we first derive analytical expressions for costs due to location

Manuscript received February 18, 2009; revised June 18, 2009. First published September 29, 2009; current version published February 19, 2010. The review of this paper was coordinated by Dr. P. Lin.

H. Saleet is with the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: hsaleet@uwaterloo.ca).

O. Basir is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: obasir@uwaterloo.ca).

R. Langar is with the Computer Science Laboratory of Paris 6 (LIP6), University of Pierre and Marie Curie—Paris Universitas, 75016 Paris, France (e-mail: rami.langar@lip6.fr).

R. Boutaba is with the Computer Science Department, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: rboutaba@uwaterloo.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVT.2009.2033079

updates and queries in the context of a general 2-D random walk mobility model. Furthermore, simulations are conducted using real mobility patterns to evaluate the performance of the proposed scheme in real mobility situations. Second, we address the total control overhead as an optimization problem. Numerical and simulation results show that the proposed location-service-management protocol can significantly reduce the location update cost and yields low querying overhead when compared with existing prominent schemes (the scalable update-based routing protocol (SLURP) [17], the XY-location service (XYLS) [20], and the hierarchical location service (HLS) [21]) under various scenarios.

The remainder of this paper is organized as follows. Section II discusses the related work reported in the literature. A description of the proposed protocol (i.e., RLSMP) is given in Section III. Section IV presents the analytical framework used to evaluate the communication overhead. In Section VI, a comparison between the proposed protocol and existing protocols is presented based on analytical and simulation results. Finally, Section VII provides our concluding remarks.

II. Related Work

Location-service-management protocols that are designed for MANETs suffer from two main shortcomings that make them unsuitable for VANETs: 1) high control overhead and 2) lack of locality awareness. In what follows, we review some of the existing location-service-management techniques and, when applicable, highlight their shortcomings.

Camp *et al.* [14] have proposed a flooding-based location protocol, namely, the dream location service. It is assumed that each node in the network sends location messages to update the location tables of all the other nodes. These messages are sent to nearby nodes more frequently than to faraway nodes. This results in high overhead, particularly in dense networks. Due to bandwidth limitations, the scalability of this protocol is questionable, as the number and the mobility of vehicles increase.

A reactive location-service protocol has been proposed by Ksemann *et al.* [15]. When a node wants to communicate with a destination, it first checks its location table. If the location information is not available or is expired, then the source node floods a location-request packet in the entire network. In this case, the delay in receiving the location-reply message increases as the number of nodes in the network increases. This affects the scalability of the protocol, making it less suitable for large and dense urban scenarios [23].

A SLURP has been proposed in [17]. This protocol divides the area covered by the network into rectangular regions. For a given node D, one specific region (called the home region) is selected by means of a hash function. As D changes its position, it transmits the corresponding updates to its home region. If another node wants to determine the position of D, it uses the same hash function to determine the region that may hold information about the new position of D. A potential drawback of the SLURP is the lack of locality awareness since it assumes that any two nodes are equally likely to communicate with each other. As the home region can be far away from both the source and destination nodes, the total path length of both updating and querying messages can be excessive.

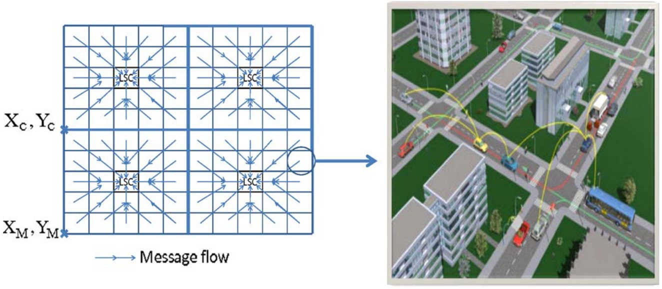
Flury and Wattenhofer [18] assume that a node can start a communication session with any other node without knowing its current location. They use a hashing function that maps node IDs to certain locations called pointers. Each node updates its current location information on a hierarchical data structure formed by these pointers. The querying messages are then routed to the destination's pointers to retrieve the location information. Location information updating frequency to a hierarchy of pointers is proportional to the vehicles' velocity. However, in a bandwidth-restricted environment, such as VANETs, such updates may cause channel congestion.

As in the SLURP, the geographic hashing location service (GHLS), which is a quorum-based protocol, uses a hashing function that maps the node ID to a region called the home region [19]. It assumes that the node closest to the center of this home region is the location server. Nevertheless, the GHLS attempts to solve the locality awareness problem of the SLURP by generating a location-service region near the center of the whole network. Although the querying overhead is lower than that of the SLURP, the updating overhead remains high due to the randomness of the hash function.

In [20], the XYLS assumes that the network space consists of vertical and horizontal strips. For each destination node, the nodes that are located along the north–south direction form the location servers (called the updating quorum), whereas for each source node, the nodes that are located along the east–west direction form the query quorum. Thus, location-updating information is disseminated in a direction such that a query can intersect an update quorum. The XYLS further assumes that the traffic pattern in the network may be random, i.e., any node can randomly initiate a communication session to any other node, and that the same node will frequently traverse the entire network, which is a rare case in VANETs. Indeed, in such ad hoc networks, the traffic pattern is mainly local, where vehicles often communicate in a local zone.

In [21], the HLS divides the network into a hierarchy of regions. For a given node D, a hash function is used to assign one specific cell on each level of the hierarchy to be responsible for keeping the location information of D. If a source node wants to communicate with D, it uses the same hash function to determine the responsible cells of D. Note that the HLS requires each node to update its current position on a hierarchy of responsible cells each time it crosses the boundary of one level. In some cases, the target node may be oscillating between two points that are located in two different levels, which results in increasing the signaling traffic in the network.

The aforementioned location-management protocols have mainly focused on reducing the signaling overhead. Achieving high scalability in VANET applications remains a major challenge. As such, we propose in this paper a scalable and locality-aware RLSMP that improves both network performance and scalability. Furthermore, we derive explicit analytical models of both location updates and queries cost and formulate the total control overhead as an optimization problem. We demonstrate that the proposed scheme enables high scalability as well as signaling cost savings compared with existing solutions.



(a) (b)

Fig. 1. System description. (a) LSC updating process. (b) Vehicle-to-vehicle communication in one cell.

III. Region-Based Location Service Management

Here, we introduce the RLSMP. We envision a VANET environment that consists of roads with intersections, which is a typical scenario in urban areas.

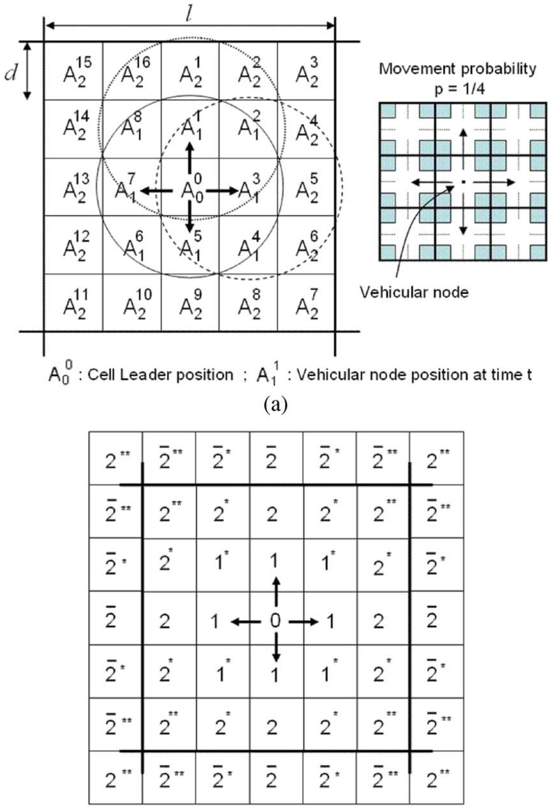
The RLSMP uses the location of nodes as a criterion for building geographical clusters. That is, each vehicle automatically determines its geographical cluster while moving (as will be explained below), with no additional communication or delay. Furthermore, the geographical clustering suggests the possibility of message aggregation, which is essential in suppressing the number of control signals in the network.

To better understand the functionalities of our protocol, we consider the following network structure as shown in Fig. 1. The vehicles in this network are considered as nodes of an ad hoc network, partitioned into virtual cells that form a virtual infrastructure. The nodes' mobility space is viewed as a grid (see Fig. 1). Each node is aware of the location of the grid origin (X_M, Y_M) (zero longitude and zero latitude). Also, each cell has an origin (X_c, Y_c) with respect to the grid origin. The origin of each cell gives the cell a unique identifier (ID) that identifies its location with respect to the grid origin. Each cell has a particular node called a cell leader (CL) that is responsible for aggregating the location information about all nodes within the cell. Furthermore, the grid is divided into segments; each segment contains a number of cells. The nodes inside one segment construct one geographical cluster.

Fig. 1 shows a grid of four clusters, and each cluster consists of 25 cells. Each node is aware of the size of the grid and the clusters, as well as the size of each cell. Therefore, a node can determine which cell and cluster it currently resides in by mapping its GPS location into coordinates in the grid.

A key property of the RLSMP is that nodes are grouped into geographical clusters. Consequently, the location information is restricted in a geographical cluster in the network. This explains the strategy of the protocol in denoting the central cell of each cluster as a home region or a location-service-management entity. Thus, nodes that are physically located in the central cell of one cluster are responsible for storing current location information about all nodes that belong to that cluster. This central cell is called the location service cell (LSC). Thus, the location information is locally kept inside one cluster.

The CL aggregates the location information about all the nodes in its cell and forwards the aggregated control message to the LSC of its cluster. Specifically, each CL stores detailed information about the mobile nodes that it manages. This information contains the node ID, the X–Y coordinates of the node location, the time of the last update, and the velocity and direction of the node movement. Then, an aggregated location message is forwarded to the LSC containing the summarized information about the nodes in the cell. We maintain that intelligently filtered or summarized information about the location of nodes in the network is sufficient. Therefore, it is not necessary to send the detailed information to the LSC since the desired level of details decreases with the increase in distance and time.


l
d
A_2^{15} A_2^{16} A_2^{1} A_2^{2} A_2^{3}
A_2^{14} A_1^{8} A_1^{1} A_1^{2} A_2^{4}
A_2^{13} A_1^{7} A_0^{0} A_1^{3} A_2^{5}
A_2^{12} A_1^{6} A_1^{5} A_1^{4} A_2^{6}
A_2^{11} A_2^{10} A_2^{9} A_2^{8} A_2^{7}
Movement probability
p = 1/4
Vehicular node
A_0^0 : Cell Leader position ; A_1^1 : Vehicular node position at time t
2^{**} $\bar{2}^{**}$ $\bar{2}^{*}$ $\bar{2}$ $\bar{2}^{*}$ $\bar{2}^{**}$ 2^{**}
$\bar{2}^{**}$ 2^{**} 2^{*} 2 2^{*} 2^{**} $\bar{2}^{**}$
$\bar{2}^{*}$ 2^{*} 1^{*} 1 1^{*} 2^{*} $\bar{2}^{*}$
$\bar{2}$ 2 1 0 1 2 $\bar{2}$
$\bar{2}^{*}$ 2^{*} 1^{*} 1 1^{*} 2^{*} $\bar{2}^{*}$
$\bar{2}^{**}$ 2^{**} 2^{*} 2 2^{*} 2^{**} $\bar{2}^{**}$
2^{**} $\bar{2}^{**}$ $\bar{2}^{*}$ $\bar{2}$ $\bar{2}^{*}$ $\bar{2}^{**}$ 2^{**}

(a)

(b)

Fig. 2. Square-cell modeling. (a) Cell model with $r = 2$ rings. (b) Corresponding states aggregation.

Hence, the cost of location updating involves two terms. The first term is due to the messages that are sent by the mobile nodes to the CL to update their locations within the cell. The second term is related to the LSC updates that are sent from the CLs to the LSC to update the location information of the nodes that reside in that cluster. We refer to the first term as the *CL updating cost* and to the second term as the *LSC updating cost*. These terms will be thoroughly discussed in the following sections.

A. CL Updating in the RLSMP

As stated before, the CL tracks the mobility of nodes within the cell and keeps the mobile node location information up-to-date. For each movement, a mobile node updates its location to the CL as follows. The decision of CL updating is based on the mobile node's location relative to the center of the cell. By analyzing this information, the mobile node can make a decision without consulting other nodes, which minimizes the overhead. Specifically, the mobile node sends its current location information to the central cell element using the geographical routing each time it crosses one cell element [see Fig. 2(a)]. In this figure, we consider one cell with a side length of l divided into 25 cell elements. Thus, the CL-updating procedure is performed whenever the mobile node moves a distance d from its current location, where d depends on the transmission range T_r of the mobile node. All nodes residing in the central cell element will then receive these messages and can act as CLs. Note that in geographical routing, each node must periodically transmit HELLO messages to its one-hop neighbors to allow nodes to know the position of their neighbors.

It is important to note that the assumption of having at least one node in the central cell element is suitable for urban dense areas, which is the case considered in this paper, since the road density in cities is relatively high. Note also that this assumption can be relaxed when using the road infrastructure. Indeed, the roadside unit (RSU) can act as a fixed CL, where we store the detailed information about the mobile nodes in that cell.

In addition, each mobile node is aware of the current cell boundary. Therefore, each time the mobile node crosses the boundary of the current cell, it informs the old CL about its movement. At the same time, it announces itself to the new CL by sending an update message to the center of the new cell. In our protocol, the boundary of one cell is estimated by the number of rings r around the central cell element. r is translated into an upper bound that triggers the renewal process, as shown in Section IV.

It is worth noting that when the CL is about to leave a central cell element, it looks into its local table and chooses a mobile node that is closer to the center of the cell with minimum velocity V. This node is selected as the new CL and, hence, will receive from the old CL all the stored information about the mobile nodes located in the cell. This information is, in fact, piggybacked to the new CL by means of the HELLO message that is sent from the old CL to its one-hop neighbors (i.e., beaconing mechanism). This mechanism is employed by all geographical routing-based protocols (e.g., SLURP, XYLS, and HLS) since each node must periodically transmit HELLO messages to its one-hop neighbors to allow the nodes know the position of their neighbors. Hence, the proposed scheme does not introduce additional overhead from the CL changing process.

B. LSC Updating in the RLSMP

The LSC is defined as the central cell of a cluster whose member nodes are responsible for keeping track of all the mobile nodes that are located in the cluster. The RLSMP relies on aggregating and forwarding the location updating messages. This process is achieved by all CLs residing in the cluster and must be synchronized among them. Indeed, a time schedule, which is denoted by $Time\_to\_Send$, is used in each CL to know when to begin sending the aggregated message. Recall that in each cell, the CL is responsible for forwarding the aggregated packets of all mobile nodes residing in the cell. Each CL stores detailed information about the mobile nodes that it manages. This information contains the node ID, the X–Y coordinates of the node location, the time of the last update, and the velocity and direction of the node movement. At the same time, the CL forwards summarized information (node ID, cell ID, and time stamp) about those nodes to the LSC of its cluster. The forwarding zone is defined by the tree structure (shown in Fig. 1), which visits the CLs. In this figure, the arrows represent flows of message aggregation, and each square represents a cell with one CL. When a CL receives location information messages from another CL in its subtrees, it collects and combines them into one aggregated message, which is forwarded to its parent until it reaches the LSC of the cluster. Such message aggregation process effectively suppresses the number of itinerant messages in the whole network. The LSC updating algorithm is described by the pseudocode in Algorithm 1.

Algorithm 1 Location Information Updating Algorithm

```
1: In one Cluster do:
2: if (Cell Leader) then
3:   Save detailed information (nodes_ID, X, Y, V, Dir) in
     local_table;
4:   Aggregate summarized information (nodes_ID,
     Cell_ID, Time Stamp);
5:   if (packet_size $\geq$ packet_size_limit) then
6:     Go to step 13;
7:   else
8:     Continue aggregation in the same packet;
9:     if (Time_to_Send) then
10:      Go to step 13;
11:    end if
12:  end if
13:  Send the aggregated message to the next Cell Leader
     in the downstream direction toward the LSC;
14:  if (next Cell Leader is the LSC) then
15:    Stop;
16:  end if
17: end if
```

Using this strategy, nodes located in the center of the LSC can act as a CL of that LSC. This "special" CL has detailed information of the mobile nodes of that LSC, as well as summarized information of all nodes belonging to the corresponding cluster. Hence, LSC renewal depends on the renewal of this "special" CL. This latter operation follows the same procedure stated above in Section III-A, i.e., using the beaconing mechanism.

C. Location-Information Retrieval in the RLSMP

The RLSMP is the first protocol that uses message aggregation in location querying. The steps involved in location querying are as follows. When a vehicle wants to communicate with another one, it forwards a query to the CL, which aggregates the querying messages and forwards them to the location servers, i.e., the nodes that are located in the local LSC. If the queries are answered by the local LSC, i.e., the destinations are registered in the same cluster as the source nodes, we call

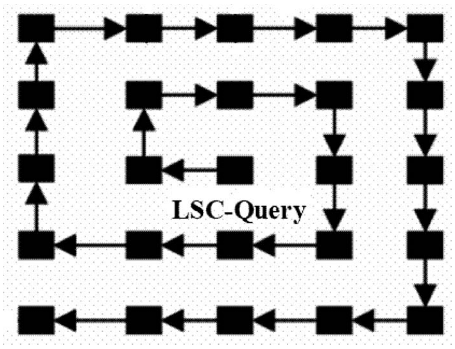


Fig. 3. Spiral shape of the location-information retrieval.

this query a *local query*. Otherwise, the query is called a *global query*, where the destinations are located in a cluster other than the local cluster of the source nodes. In this case, the local LSC does not directly forward global queries; instead, they are delayed for a prespecified time. This delay is essential for aggregating queries that are sent by the vehicles residing in that cluster. The forwarded aggregated queries pass through the different LSCs, as shown in Fig. 3. They are forwarded in a spiral shape around the local LSC, where this spiral shape visits all surrounding LSCs until it finds information about the destinations' location. The nodes inside the visited LSCs will make use of the information stored in their own tables to determine the destinations' IDs.

The use of the spiral shape is motivated by the fact that any location service protocol has to account for the locality awareness property of VANETs [23], [24]. Indeed, the communication patterns in VANETs are almost local (i.e., within the same geographical area) [23], [24]. In the proposed scheme, this is achieved by exploring the source nodes' vicinity since the destination search begins in clusters surrounding the local cluster of the source nodes. In addition, as opposed to broadcasting or multicasting trees, the spiral shape minimizes the communication overhead and saves bandwidth. This querying algorithm of the RLSMP is described by the pseudocode in Algorithm 2.

Algorithm 2 Querying Algorithm

```
 1: In the network do:
 2: if (Cell Leader) then
 3:   Save detailed information (srcs_ID, X, Y, V, Dir,
      Dst_ID) in local_table;
 4:   Aggregate summarized information (srcs_ID, Cell_ID,
      Dst_ID, Time Stamp);
 5:   if (packet_size ≥ packet_size_limit) then
 6:     Go to step 13;
 7:   else
 8:   Continue aggregation in the same packet;
 9:     if (Time_to_Send) then
10:       Go to step 13;
11:     end if
12:   end if
13:   Send the aggregated message to the next Cell Leader
      in the downstream direction toward the LSC;
14:   if (Dst_ID is found) then
15:     Go to step 29;
16:   else
17:     if (Next Cell Leader is LSC) then
18:       Look up in the surrounding 8 LSCs;
19:       if (Dst_ID is found) then
20:         Go to step 29;
21:       else
22:         Continue Lookup by enlarging the spiral around the
            source LSC to cover all network;
23:         if (Dst_ID is found) then
24:           Go to step 29;
25:         end if
26:       end if
27:     end if
28:   end if
29:   Send a reply packet to the source node;
30:   if (the source node receives the reply packet)} then
31:     Stop;
32:   end if
33: end if
```

It is worth noting that, for both the SLURP [17] and the HLS [21], since the structure of these two protocols assumes that each mobile node in the network has its own home region, it is difficult to aggregate the messages and then send them to the different home regions. Thus, in this case, updates and queries are sent individually to the corresponding home regions. On the other hand, the XYLS [20] can support aggregation since update and query messages can be sent in the whole column or row. However, this process is not considered in the XYLS [20]. Indeed, according to [20], each node will individually send its own updates and queries since the mobile nodes are free to move and traverse the network frequently.

In Section IV, we present a framework for analyzing the behavior of the location-management scheme of the RLSMP. We develop Markovian models to estimate the communication overhead. The obtained results will be used, in a latter stage, to derive the protocol performance metrics such as the cell-boundary crossing rate, the location update cost, and the location query cost.

IV. Framework for Location-Management Analysis

Here, we derive analytical expressions for both location updating and querying costs of the RLSMP. Table I describes the parameters that are used in the analysis.

A. Location-Updating Cost

As stated before, the location-updating cost in the RLSMP involves two terms. The first term is pertaining to the messages that are sent by the mobile nodes to the CL to update their locations within the cell (i.e., CL updating cost). The second term is related to the updates that are sent from the CLs to the local LSC to update the location information of the nodes that reside in that cluster (i.e., LSC updating cost).

1) CL Location-Updating Cost: In this paper, we consider the square-based 2-D model. As opposed to [25], where the

TABLE I
LIST OF PARAMETERS

Parameter	Description
A	network area
u_1	the content of the updating packet that CLs send to the local LSC
u_2	the content of the query packet
d	side length of one cell element
l	cell side length
n	cell side length multiplicity
N	number of nodes in the network
p	probability of movement to the neighboring cells
p_b	probability of boundary crossing
p_l	probability of local querying
r	number of rings of cell elements in one cell
R	number of rings of cells in one cluster
R_c	number of rings of clusters in the network
f_u	updating frequency
f_q	Querying frequency
T_r	transmission range
z	average progress in one hop
γ	node density

focus is on n-subarea cells that form a diamond-shape cluster, our focus here is on the cell itself. Typically, a cell with a side length of l is divided into $n \times n$ square cell elements that form a grid (see Fig. 2). We call n the cell-side-length multiplicity. The side length d of each cell element is determined such that each node D_i in the cell element i can directly communicate (i.e., in one hop) with a node D_j that is located in the adjacent cell element j. In this case, $d = T_r/\sqrt{5}$.

In this paper, we consider first a general 2-D random-walk model as in [25]–[28]. Consequently, under this model, the mobile node can move to one of the neighboring cell elements with equal probability p ($p = 1/4$). However, the general methodology we present applies to other mobility models as well, specifically, models that have approximately exponential distribution times. These include many popular mobility models like random direction and random waypoint. In addition, by using different values of the probability p_k such that $\sum_{k=1}^{4} p_k = 1$, different nonrandom mobility patterns can be generated. For example, for the Manhattan mobility model [29], the probability of moving right or left is 0.25, and the probability that the mobile node moves in the same direction is 0.5.

Fig. 2(a) represents an example of the cell model used in this paper when $n = 5$. The cell contains the CL's area surrounded by $r = \lfloor n/2 \rfloor = 2$ rings of cell elements. Each element is referenced by the ring label and its position inside that ring, which determines the mobile node's position with respect to the CL. For example, cell elements belonging to ring 1 are referenced by A_1^j, $1 \leq j \leq 8$, those belonging to ring 2 are referenced by A_2^j, $1 \leq j \leq 16$, and so on. To generalize, let $i = 0, 1, \ldots, r$ designate the ith ring away from the CL. The CL is denoted by A_0^0. Cell elements belonging to ring i are referenced by A_i^j, $1 \leq j \leq 8i$.

Let $X(t)$ be the mobile node's location within the cell at time t. The sojourn time of a mobile node in each element A_i^j is assumed to be exponentially distributed with mean $1/\mu$. $\{X(t), t \geq 0\}$ is, therefore, a Markov process with continuous time and finite state space $E = \{A_i^j | 0 \leq i \leq r, 1 \leq j \leq 8i\}$. Recall that our main objective is to determine the mobile node's position within the cell to predict its evolution. According to its next location, we can compute the CL location-update cost as well as the cell-boundary crossing rate corresponding to the RLSMP.

The resolution of the Markovian chain $X(t)$, as defined above, is time consuming. Moreover, this chain suffers from the state space explosion problem mainly when the number of rings is high. To avoid this issue, we extract a new chain $Y(t)$ from $X(t)$ by aggregating its states. In other words, all the states where the mobile node exhibits exactly the same behavior will be aggregated. Hence, the size of the state space E will be reduced. To achieve this, we make use of the symmetric property of the 2-D model. The algorithm to perform the state aggregation is described as follows.

1) Let A_i^j denote the cell element that contains the mobile node. As presented in Fig. 2(a), state A_i^1 is chosen to be the one at the top of state A_{i-1}^1. Subsequently, each ring i consists of $8i$ elements labeled in a clockwise direction as $A_i^1, \ldots, A_i^{8i}$. Let A_i^{j*} denote the new aggregated state of the cell, where i always designates the ring reference, and $j*$ is the state label inside the ring. Since all cells of the cluster have the same size, the aggregated states A_{r+1}^{j*} located at the ring $r+1$ represent the boundary states of the cell under consideration. These states will be denoted by $\bar{A}_r^{j*}$.

2)

```
Start with $i = 1$;
until $(i = r)$
Repeat {
  set $A_i^{0*} = A_i = \bigcup_{0 \leq j \leq 3} A_i^{2ij+1}$
  For $m = 1$ to $m = i$
    set $A_i^{m*} = \bigcup_{0 \leq j \leq 3} A_i^{2ij+m+1} + \bigcup_{1 \leq j \leq 4} A_i^{2ij-m+1}$
  $i = i + 1$;
}
set $\bar{A}_r^{0*} = \bar{A}_r = \bigcup_{0 \leq j \leq 3} A_{r+1}^{2j(r+1)+1}$
For $m = 1$ to $m = r$
  set $\bar{A}_r^{m*} = \bigcup_{0 \leq j \leq 3} A_{r+1}^{2j(r+1)+m+1} + \bigcup_{1 \leq j \leq 4} A_{r+1}^{2j(r+1)-m+1}$
```

For instance, for $r = 2$, we obtain the following aggregated states:

$$
\begin{aligned}
A_0^{0*} &= A_0 = \{A_0^0\} \\
A_1^{0*} &= A_1 = \{A_1^1, A_1^3, A_1^5, A_1^7\} \\
A_1^{1*} &= \{A_1^2, A_1^4, A_1^6, A_1^8\} \\
A_2^{0*} &= A_2 = \{A_2^1, A_2^5, A_2^9, A_2^{13}\} \\
A_2^{1*} &= \{A_2^2, A_2^4, A_2^6, A_2^8, A_2^{10}, A_2^{12}, A_2^{14}, A_2^{16}\} \\
A_2^{2*} &= \{A_2^3, A_2^7, A_2^{11}, A_2^{15}\} \\
\bar{A}_2^{0*} &= \bar{A}_2 = \{A_3^1, A_3^7, A_3^{13}, A_3^{19}\} \\
\bar{A}_2^{1*} &= \{A_3^2, A_3^6, A_3^8, A_3^{12}, A_3^{14}, A_3^{18}, A_3^{20}, A_3^{24}\} \\
\bar{A}_2^{2*} &= \{A_3^3, A_3^5, A_3^9, A_3^{11}, A_3^{15}, A_3^{17}, A_3^{21}, A_3^{23}\}.
\end{aligned}
$$

For ease of use, the aggregate states were assigned numbers as follows [see Fig. 2(b)]:

$$
A_0 \rightarrow 0, A_1 \rightarrow 1, A_1^{1*} \rightarrow 1^*, A_2 \rightarrow 2, A_2^{1*} \rightarrow 2^*
$$
$$
A_2^{2*} \rightarrow 2^{**}, \bar{A}_2 \rightarrow \bar{2}, \bar{A}_2^{1*} \rightarrow \bar{2}^*, \bar{A}_2^{2*} \rightarrow \bar{2}^{**}.
$$

Theorem: Let $F = \{A_0, A_1, A_1^{1*}, \ldots, A_i, A_i^{1*}, \ldots, A_i^{i*}, \ldots, A_r, A_r^{1*}, \ldots, A_r^{r*}, \bar{A}_r, \bar{A}_r^{1*}, \ldots, \bar{A}_r^{r*}\}$ designate the state space of the new chain $Y(t)$ obtained from the aggregation of the initial Markovian chain $X(t)$. The resulting aggregated process $Y(t)$ is also Markovian.

Proof: For convenience, we denote by F_i each state of the set F and by $M = (r+1) \times (r+4)/2$ the set size. Let Q_{init} designate the generator matrix of the initial Markov chain $X(t)$. We arrange the states of E according to the space F partitions (i.e., $E = \{\underbrace{A_0^0}_{F_1}, \underbrace{A_1^1, A_1^3, A_1^5, A_1^7}_{F_2}, \underbrace{A_1^2, A_1^4, A_1^6, A_1^8}_{F_3}, \underbrace{A_2^1, A_2^5, A_2^9, A_2^{13}}_{F_4}$ $\underbrace{A_2^2, A_2^4, A_2^6, A_2^8, A_2^{10}, A_2^{12}, A_2^{14}, A_2^{16}}_{F_5}, \underbrace{A_2^3, A_2^7, A_2^{11}, A_2^{15}}_{F_6}, \ldots\}$).

In this case, the infinitesimal matrix Q_{init} can be written as an $S \times S$ matrix $(S = 1 + (\sum_{i=1}^{r+1} 8i) - 4)$ and has the following form:

$$Q_{\text{init}} = (B_{ms})_{1 \le m,s \le M}$$

where B_{ms} is the block matrix corresponding to the transition probabilities between each element of the set F_m and the set F_s. In addition, these blocks verify the constant-row sum property [30]. In other words, we have

$$\forall\, i, \sum_j (B_{ms})_{ij} \text{ is a constant, denoted by } c_{ms}\ (c_{ms} \ge 0).$$

Thus, according to [30], the resulting aggregated process is Markovian. To illustrate this result, let us revisit the example of Fig. 2(b) where $r = 2$. As discussed before, and according to the state-aggregation algorithm, the state space E of the initial Markov chain $X(t)$ can be arranged with respect to the partition F. The generator matrix Q_{init} of the process $X(t)$ can be written as in (1), shown on the bottom of the next page, where $a = p\mu$, and $b = -4p\mu$. It is obvious to see that each block matrix justifies the constant-row sum property. The new generator matrix Q_{aggr} of the aggregated Markov chain $Y(t)$ is, therefore, given by

$$Q_{\text{aggr}} = \left[\begin{array}{ccccccccc}
A_0 & A_1 & A_1^{1*} & A_2 & A_2^{1*} & A_2^{2*} & \bar{A}_2 & \bar{A}_2^{1*} & \bar{A}_2^{2*} \\
b & 4a & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a & b & 2a & a & 0 & 0 & 0 & 0 & 0 \\
0 & 2a & b & 0 & 2a & 0 & 0 & 0 & 0 \\
0 & a & 0 & b & 2a & 0 & a & 0 & 0 \\
0 & 0 & a & a & b & a & 0 & a & 0 \\
0 & 0 & 0 & 0 & 2a & b & 0 & 0 & 2a \\
0 & a & 0 & 0 & 2a & 0 & a+b & 0 & 0 \\
0 & 0 & a & a & 0 & a & 0 & a+b & 0 \\
0 & 0 & 0 & 0 & 2a & 0 & 0 & 0 & 2a+b
\end{array}\right]. \tag{2}$$

The associated state transition diagram (i.e., for $r = 2$) is depicted in Fig. 4.

It is important to note that the proposed analytical model uses a new feature to reduce constraints on mobile movements and provide a more-realistic roaming scenario with minimal assumptions compared with that reported in [25]. Indeed, in [25], all boundary cells are aggregated into one state called the absorbing state. In our model, we use rather a set of aggregate states (i.e., original states) to trace user movement within one cell and then use a set of special (bar) states when crossing the boundary of this cell (i.e., for the boundary cells). Consequently, we model the mobile node's movement to enter the original states again from the special states once the mobile node starts moving within the new cell.

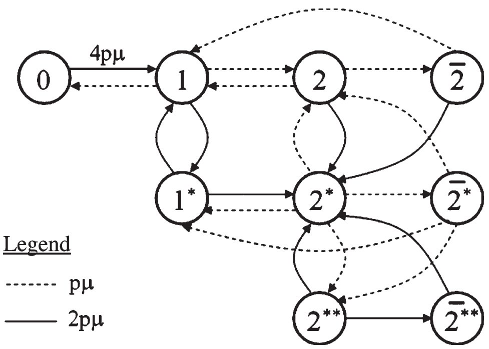


Fig. 4. State transition diagram of the aggregated Markov chain for $r = 2$.

To illustrate this, let us consider again the example of Fig. 2(b). In this figure, as long as the mobile node moves within the same cell, it is in one of the main aggregate states, i.e., 1, 1^*, 2, 2^*, or 2^{**}. The transitions to the bar states indicate that the mobile node moves to a boundary state in an adjacent cell. When the mobile node starts moving in the new cell, for example, when it moves from state $\bar{2}$ to $\bar{2}^*$ in the new cell, our proposed model binds to the normal states (i.e., original states), indicating that the mobile node has now started moving in cell elements belonging to the same new cell. This is shown by the transitions from the bar states (i.e., $\bar{2}$, $\bar{2}^*$, and $\bar{2}^{**}$) to the normal states in Fig. 4.

Steady-State Probabilities: Based on the state transition diagram of the aggregated Markov chain (see Fig. 4 where $r = 2$), we can obtain the steady-state probability for state F_i $(i = 1, \ldots, M)$. Denote by Π_i and $\Pi_i^{(m)}$ $[i = (0, 1, \ldots, r)$ and $m = (1, \ldots, i)]$ the stationary probability of the system for aggregated states A_i and A_i^{m*}, respectively. Denote also by $\Pi_{\bar{r}}$ and $\Pi_{\bar{r}}^{(m)}$, $m = (1, \ldots, r)$ the stationary probability of the system for the boundary states. The balance equations for the aggregated Markov chain are recursively obtained $\forall r \ge 2$ as follows:

$$\begin{cases}
\Pi_0 = p\Pi_1 \\
\Pi_1 = 4p\Pi_0 + 2p\Pi_1^{(1)} + p\Pi_2 + \alpha p\Pi_{\bar{2}} \\
\forall\, 2 \le i \le r-1 \\
\Pi_i = p\Pi_{i-1} + p\Pi_{i+1} + p\Pi_i^{(1)} + \beta p\Pi_{i\bar{+}1} \\
\Pi_r = p\Pi_{r-1} + p\Pi_r^{(1)} + p\Pi_{\bar{r}}^{(1)}
\end{cases} \tag{3}$$

$$\begin{cases}
\Pi_1^{(1)} = 2p\Pi_1 + p\Pi_2^{(1)} + \alpha p\Pi_{\bar{2}}^{(1)} \\
\forall\, 2 \le i \le r-1 \\
\Pi_i^{(i)} = p\Pi_i^{(i-1)} + p\Pi_{i+1}^{(i)} + \beta p\Pi_{i\bar{+}1}^{(i)} \\
\Pi_r^{(r)} = p\Pi_r^{(r-1)} + p\Pi_{\bar{r}}^{(r-1)}
\end{cases} \tag{4}$$

$$\begin{cases}
\forall\, 2 \le i \le r-1, \qquad j = 1 \\
\Pi_i^{(1)} = 2p\Pi_i + p\delta\Pi_{i-1}^{(1)} + p\delta\Pi_i^{(2)} + p\Pi_{i+1}^{(1)} + \beta p\Pi_{i\bar{+}1}^{(1)} \\
\forall\, 2 \le j \le r-2 \text{ and } j+1 \le i \le r-1 \\
\Pi_i^{(j)} = p\Pi_i^{(j-1)} + p\delta\Pi_{i-1}^{(j)} + p\delta\Pi_i^{(j+1)} \\
\qquad + p\Pi_{i+1}^{(j)} + \beta p\Pi_{i\bar{+}1}^{(j)}
\end{cases} \tag{5}$$

$$\begin{cases} \Pi_r^{(1)} = 2p\Pi_r + p\Pi_{r-1}^{(1)} + p\Pi_r^{(2)} + 2p\Pi_{\bar{r}} + p\Pi_{\bar{r}}^{(2)} \\ \forall\ 2 \le j \le r-1 \\ \Pi_r^{(j)} = p\Pi_r^{(j-1)} + p\theta\Pi_{r-1}^{(j)} + p\theta\Pi_r^{(j+1)} \\ \qquad\qquad + p\theta\Pi_{\bar{r}}^{(j+1)} + p\Pi_{\bar{r}}^{(j-1)} \end{cases} \tag{6}$$

$$\begin{cases} 3p\Pi_{\bar{r}} = p\Pi_r \\ \forall\ 1 \le j \le r-1, \quad 3p\Pi_{\bar{r}}^{(j)} = p\Pi_r^{(j)} \\ 2p\Pi_{\bar{r}}^{(r)} = 2p\Pi_r^{(r)} \end{cases} \tag{7}$$

where

$$\alpha = \begin{cases} 1, & \text{if } r = 2 \\ 0, & \text{otherwise} \end{cases} \quad ; \quad \beta = \begin{cases} 1, & \text{if } i = r-1 \\ 0, & \text{otherwise} \end{cases}$$

$$\delta = \begin{cases} 2, & \text{if } i = j+1 \\ 1, & \text{otherwise} \end{cases} \quad ; \quad \theta = \begin{cases} 2, & \text{if } j = r-1 \\ 1, & \text{otherwise} \end{cases}$$

$$\sum_{i=0}^{r} \Pi_i + \sum_{i=1}^{r} \sum_{j=1}^{i} \Pi_i^{(j)} + \Pi_{\bar{r}} + \sum_{j=1}^{r} \Pi_{\bar{r}}^{(j)} = 1. \tag{8}$$

$$Q_{init} = \left[\begin{array}{c|cccc|cccc|cccc|cccccccc|cccc|cccc|cccccccc|cccccccc}
A_0^0 & A_1^1 & {}_1^3 & {}_1^5 & {}_1^7 & A_1^2 & {}_1^4 & {}_1^6 & {}_1^8 & A_2^1 & {}_2^5 & {}_2^9 & {}_2^{13} & A_2^2 & {}_2^4 & {}_2^6 & {}_2^8 & {}_2^{10} & {}_2^{12} & {}_2^{14} & {}_2^{16} & A_2^3 & {}_2^7 & {}_2^{11} & {}_2^{15} & A_3^1 & {}_3^7 & {}_3^{13} & {}_3^{19} & A_3^2 & {}_3^6 & {}_3^8 & {}_3^{12} & {}_3^{14} & {}_3^{18} & {}_3^{20} & {}_3^{24} & A_3^3 & {}_3^5 & {}_3^9 & {}_3^{11} & {}_3^{15} & {}_3^{17} & {}_3^{21} & {}_3^{23} \\
b & a&a&a&a & &&& & &&& & &&&&&&& & &&& & &&& & &&&&&&& & &&&&&&& \\
\hline
a & b&&& & a&&&a & a&&& & &&&&&&& & &&& & &&& & &&&&&&& & &&&&&&& \\
a & &b&& & a&a&& & &a&& & &&&&&&& & &&& & &&& & &&&&&&& & &&&&&&& \\
a & &&b& & &a&a& & &&a& & &&&&&&& & &&& & &&& & &&&&&&& & &&&&&&& \\
a & &&&b & &&a&a & &&&a & &&&&&&& & &&& & &&& & &&&&&&& & &&&&&&& \\
\hline
 & a&a&& & b&&& & &&& & a&a&&&&&& & &&& & &&& & &&&&&&& & &&&&&&& \\
 & &a&a& & &b&& & &&& & &&a&a&&&& & &&& & &&& & &&&&&&& & &&&&&&& \\
 & &&a&a & &&b& & &&& & &&&&a&a&& & &&& & &&& & &&&&&&& & &&&&&&& \\
 & a&&&a & &&&b & &&& & &&&&&&a&a & &&& & &&& & &&&&&&& & &&&&&&& \\
\hline
 & a&&& & &&& & b&&& & a&&&&&&&a & &&& & a&&& & &&&&&&& & &&&&&&& \\
 & &a&& & &&& & &b&& & &a&a&&&&& & &&& & &a&& & &&&&&&& & &&&&&&& \\
 & &&a& & &&& & &&b& & &&&a&a&&& & &&& & &&a& & &&&&&&& & &&&&&&& \\
 & &&&a & &&& & &&&b & &&&&&a&a& & &&& & &&&a & &&&&&&& & &&&&&&& \\
\hline
 & &&& & a&&& & a&&& & b&&&&&&& & a&&& & &&& & a&&&&&&& & &&&&&&& \\
 & &&& & a&&& & &a&& & &b&&&&&& & a&&& & &&& & &a&&&&&& & &&&&&&& \\
 & &&& & &a&& & &a&& & &&b&&&&& & &a&& & &&& & &&a&&&&& & &&&&&&& \\
 & &&& & &a&& & &&a& & &&&b&&&& & &a&& & &&& & &&&a&&&& & &&&&&&& \\
 & &&& & &&a& & &&a& & &&&&b&&& & &&a& & &&& & &&&&a&&& & &&&&&&& \\
 & &&& & &&a& & &&&a & &&&&&b&& & &&a& & &&& & &&&&&a&& & &&&&&&& \\
 & &&& & &&&a & &&&a & &&&&&&b& & &&&a & &&& & &&&&&&a& & &&&&&&& \\
 & &&& & &&&a & a&&& & &&&&&&&b & &&&a & &&& & &&&&&&&a & &&&&&&& \\
\hline
 & &&& & &&& & &&& & a&a&&&&&& & b&&& & &&& & &&&&&&& & a&a&&&&&& \\
 & &&& & &&& & &&& & &&a&a&&&& & &b&& & &&& & &&&&&&& & &&a&a&&&& \\
 & &&& & &&& & &&& & &&&&a&a&& & &&b& & &&& & &&&&&&& & &&&&a&a&& \\
 & &&& & &&& & &&& & &&&&&&a&a & &&&b & &&& & &&&&&&& & &&&&&&a&a \\
\hline
 & &&a& & &&& & &&& & &&&a&a&&& & &&& & b&&a& & &&&&&&& & &&&&&&& \\
 & &&&a & &&& & &&& & &&&&&a&a& & &&& & &b&&a & &&&&&&& & &&&&&&& \\
 & a&&& & &&& & &&& & a&&&&&&&a & &&& & a&&b& & &&&&&&& & &&&&&&& \\
 & &a&& & &&& & &&& & &a&a&&&&& & &&& & &a&&b & &&&&&&& & &&&&&&& \\
\hline
 & &&& & &a&& & &&a& & &&&&&&& & &a&& & &&& & b&&&a&&&& & &&&&&&& \\
 & &&& & &&&a & &&&a & &&&&&&& & &&&a & &&& & &b&&&&&a& & &&&&&&& \\
 & &&& & &&a& & &&&a & &&&&&&& & &&a& & &&& & &&b&&&a&& & &&&&&&& \\
 & &&& & a&&& & a&&& & &&&&&&& & a&&& & &&& & a&&&b&&&& & &&&&&&& \\
 & &&& & &&&a & a&&& & &&&&&&& & &&&a & &&& & &&&&b&&&a & &&&&&&& \\
 & &&& & &a&& & &a&& & &&&&&&& & &a&& & &&& & &&a&&&b&& & &&&&&&& \\
 & &&& & a&&& & &a&& & &&&&&&& & a&&& & &&& & &a&&&&&b& & &&&&&&& \\
 & &&& & &&a& & &&a& & &&&&&&& & &&a& & &&& & &&&&a&&&b & &&&&&&& \\
\hline
 & &&& & &&& & &&& & &&a&a&&&& & &&& & &&& & &&&&&&& & b&&a&a&&&& \\
 & &&& & &&& & &&& & &&&&&&a&a & &&& & &&& & &&&&&&& & &b&&&&&a&a \\
 & &&& & &&& & &&& & &&&&a&a&& & &&& & &&& & &&&&&&& & &&b&&a&a&& \\
 & &&& & &&& & &&& & a&a&&&&&& & &&& & &&& & &&&&&&& & a&a&&b&&&& \\
 & &&& & &&& & &&& & &&&&&&a&a & &&& & &&& & &&&&&&& & &&&&b&&a&a \\
 & &&& & &&& & &&& & &&a&a&&&& & &&& & &&& & &&&&&&& & &&a&a&&b&& \\
 & &&& & &&& & &&& & a&a&&&&&& & &&& & &&& & &&&&&&& & a&a&&&&&b& \\
 & &&& & &&& & &&& & &&&&a&a&& & &&& & &&& & &&&&&&& & &&&&a&a&&b
\end{array}\right] \tag{1}$$

Given the balance equations (3)–(7) and the normalization equation (8), the steady-state probabilities of the aggregated Markov chain can be derived. In the following, the obtained results will be used to derive the cell-boundary crossing rate and the CL location updating cost.

Cell-boundary crossing rate: Let P_b denote the probability that a mobile node crosses the cell boundary when moving within the cell. Such a situation happens when the mobile node is located either at the ring r of the cell or at the boundary states, and it moves in the direction that increases the number of rings with respect to the current cell. Based on the above analysis, this probability can be given by

$$P_b = p \times \left(\Pi_r + \Pi_{\bar{r}} + \Pi_r^{(r)} + \Pi_{\bar{r}}^{(r)} + \sum_{j=1}^{r} \left(\Pi_r^{(j)} + \Pi_{\bar{r}}^{(j)} \right) \right). \tag{9}$$

CL location update cost: Let CL_{updates} denote the cost of CL location updates when the mobile node moves within the cell. According to the node mobility, this cost can be written as follows:

$$C_1 = \text{Cost}_{\text{intra}} + \text{Cost}_{\text{inter}} \tag{10}$$

where $\text{Cost}_{\text{intra}}$ and $\text{Cost}_{\text{inter}}$, respectively, denote the signaling cost of location updates when the mobile node moves within the same cell (i.e., intracell movement) and when the mobile node crosses the cell boundary (i.e., intercell movement). Using the results of Section IV, the expressions of $\text{Cost}_{\text{intra}}$ and $\text{Cost}_{\text{inter}}$ are given as

$$\begin{aligned}\text{Cost}_{\text{intra}} =& \sum_{i=1}^{r-1} 2p_i(2i+1)\Pi_i + \sum_{i=1}^{r-1}\sum_{j=1}^{i} 4p(i+j)\Pi_i^{(j)} \\ &+ p(3r+1)\Pi_r + 2p(2r-1)\Pi_r^{(r)} \\ &+ \sum_{j=1}^{r-1} p(3r+3j-1)\Pi_r^{(j)} \qquad (11)\\ \text{Cost}_{\text{inter}} =& \, p(2r+1)(\Pi_r+\Pi_{\bar{r}}) + p(4r+1)\left(\Pi_r^{(r)}+\Pi_{\bar{r}}^{(r)}\right) \\ &+ \sum_{j=1}^{r} p(2r+2j+1)\left(\Pi_r^{(j)}+\Pi_{\bar{r}}^{(j)}\right) \qquad (12)\end{aligned}$$

where $r = 0.5(n-1)$.

2) LSC Updating Cost: To evaluate the LSC updating cost, we define a new term: the number of cell rings R. From Fig. 5, we distinguish between the number of rings of cell elements r and the number of cell rings R. For example, the first eight cells surrounding the local LSC are called the first cell ring, the next 16 surrounding cells are called the second cell ring, and so on. In this figure, the number of cell rings R is equal to 2, and the number of rings of cell elements r is also equal to 2.

To calculate the LSC updating cost, we consider three main factors as follows.

Node density: The LSC updating cost is proportional to the number of nodes in one cell. Recall that in each cell, the CL is responsible for forwarding the aggregated packets of all mobile nodes residing in that cell. Therefore, the LSC updating cost (in terms of bytes) is proportional to $\gamma(nd)^2 u_1$, where u_1 is the number of bytes that represent the data for one specific node (see Table II), and γ is the vehicle density measured in nodes per square kilometer.

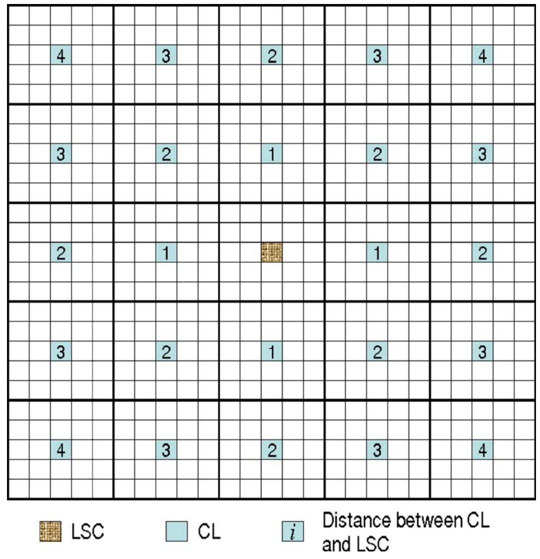


Fig. 5. Distance between the CLs and their LSC in one cluster.

TABLE II
DETAILS ABOUT u_1 AND u_2 IN THE NUMBER OF BYTES [32]

Field type	u_1	u_2
$Node\ Identifier(ID)$	8	8
$Destination\ Identifier(ID)$	-	8
$Cell\ Identifier(ID)$	8	8
$Time\ stamp$	2	2

Cluster size: The average distance of the different CLs from the local LSC depends on the cell-side-length multiplicity n and the number of cell rings R around the LSC. In Fig. 5, the number assigned to the CL of each cell is the distance between that CL and the local LSC in terms of the cell side length l. Recall that $l = nd$. Therefore, to calculate the average distance, we use the power series as follows:

$$\begin{aligned} D_{\text{ave}} &= \frac{nd}{(2R+1)^2}\left(\sum_{i=1}^{R} 4i\left((2R+1)-i\right) + \sum_{i=1}^{R} 4i^2\right) \\ &= \frac{nd}{(2R+1)^2}\sum_{i=1}^{R} 4i(2R+1) \\ &= \frac{nd}{(2R+1)} 2R(R+1). \end{aligned} \tag{13}$$

Updating frequency: Recall that the aggregating and forwarding processes in the RLSMP are synchronized using the time schedule $Time\_to\_Send$. This time schedule represents the LSC updating frequency.

By considering the three factors mentioned above, we can formulate the LSC updating cost (in terms of bytes × hops / second) as follows:

$$C_2 = f_u \frac{nd}{(2R+1)} \frac{2R(R+1)}{z} \gamma(nd)^2 u_1 \tag{14}$$

where f_u denotes the frequency of sending the aggregated updates, and z denotes the average forward progress made toward a destination in the course of one transmission [17]. Note that z depends on the transmission range T_r and the average node density γ. In this paper, we assume that both T_r and γ are constant; hence, z is constant.

B. Location-Querying Cost

In the RLSMP, we distinguish between local and global queries. Local queries correspond to those answered by the local LSC inside one cluster (i.e., both the source and destination nodes are registered in the same cluster). On the other hand, in global queries, other LSCs will be involved in the location query process to find the destinations' IDs.

1) Local Query: When a vehicle wants to communicate with another node, it forwards a query to the CL of the cell at which it resides. The simple case happens when both source and destination nodes reside in the same cell. In this case, the CL directly informs the source with the destination location. Otherwise, a timer is triggered by the CL to begin aggregating the querying messages and forwarding them to the local LSC. Following the same steps as above, the overhead introduced by this kind of query can be determined as follows:

$$C_3 = p_l f_q \frac{nd}{(2R+1)} \frac{2R(R+1)}{z} \gamma (nd)^2 u_2 \tag{15}$$

where f_q and u_2 denote the frequency of sending the queries and the average query packet size, respectively. Recall that in VANETs, the traffic pattern is assumed to be local. Therefore, we assume in our analysis that the probability of initiating a query is exponentially decaying, as shown by the following:

$$\sum_{i=1}^{2} p_i = p_l + \frac{p_l}{2} = 1. \tag{16}$$

This means that the probability of local querying (i.e., p_l) is higher than that of global querying (i.e., $1 - p_l$).

2) Global Query: If the destination is not located in the local cluster where the source node resides, the query is called a *global query*. For simplicity, let us refer to the first eight clusters surrounding the local LSC as the first cluster ring. The next 16 surrounding clusters correspond to the second cluster ring, and so on. The global querying cost is affected by the following three factors: node density, network size, and global querying probability.

Node density: The global querying cost is proportional to the number of nodes in one cluster. Therefore, the global querying cost (in terms of bytes) is proportional to $\gamma(2R+1)^2(nd)^2u_2$.

Network size A: The distance traveled from the local LSC to the first LSC, which is located in the first cluster ring, is equal to $(nd)(2R+1)$. Thus, the distance traveled to visit the ith cluster ring $(i = 0, \ldots, Rc)$ is $8i(nd)(2R+1)$, where R_c denotes the number of rings of clusters in the network. This parameter can be given as

$$R_c = \frac{\sqrt{A}}{2(2R+1)(nd)} - \frac{1}{2}. \tag{17}$$

Querying frequency: Recall that f_q designates the frequency by which the nodes send the queries, and $1 - p_l$ corresponds to the probability of a global query. The cost of sending the queries from the local LSC to the surrounding LSCs can be given as follows:

$$\begin{aligned}\text{Cost}_q &= (1-p_l) f_q \frac{8u_2\gamma(2R+1)^3(nd)^3}{z} \sum_{i=1}^{R_c} i \\ &= (1-p_l) f_q \frac{8u_2\gamma(2R+1)^3(nd)^3}{2z} R_c(R_c+1). \end{aligned} \tag{18}$$

Hence, the global querying cost C_4 is expressed as

$$C_4 = \frac{(1-p_l)C_3}{p_l \gamma (nd)^2} + \frac{\text{Cost}_q}{\gamma(2R+1)^2(nd)^2} \tag{19}$$

where the first term corresponds to the normalized cost related to the queries sent from the CLs to the local LSC, and the second term is the normalized cost of forwarding the queries from the local LSC to the surrounding ones.

V. Formulating the Total Control Overhead as an Optimization Problem

Here, we address the optimal values of the cell-side-length multiplicity n and the number of rings of cells R in one cluster that minimizes the total control overhead.

Based on the above analysis, this cost is defined as the sum of the total updating cost, which includes both the CL updates (i.e., C_1) and the LSC updates (i.e., C_2), as well as the total querying cost that comprises the local queries (i.e., C_3) and the global queries (i.e., C_4).

Consequently, we formulate the problem as a simple optimization problem with the following objective function:

$$\min_{n,R} \sum_{i=1}^{4} C_i \tag{20}$$

subject to

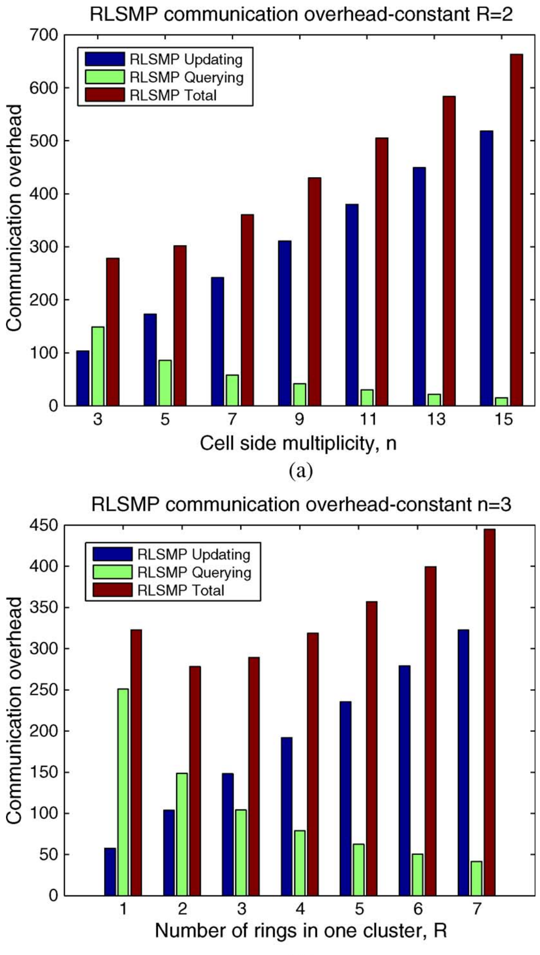
$$n \in \left[1, 2, \ldots, \left\lfloor \frac{\sqrt{A}}{d} \right\rfloor\right] \tag{21}$$

$$R \in \left[0, 1, 2, \ldots, \left\lfloor \sqrt{\frac{A}{(2R+1)^2(nd)^2}} - 1 \right\rfloor\right]. \tag{22}$$

The rational behind this is that, given a network of area A, the network designer can compute the optimal values of n and R by solving the above problem using total enumeration (see Fig. 6). Thus, the total control overhead will be minimized, and an optimal network structure will be formed.

VI. Numerical and Simulation Results

Here, we compare the RLSMP with respect to the SLURP [17], the XYLS [20], and the HLS [21] through both simulations and analytical approaches. To evaluate the cost of location updates and location queries by simulations, we developed a discrete-event simulator.



(a)

(b)

Fig. 6. Optimal communication cost. (a) Constant R. (b) Constant n.

The simulation environment consists of a large-scale wireless ad hoc network. A number of nodes N are randomly generated within the network. The resulting network is considered only if all nodes maintain connectivity between them, i.e., there is at least a path that connects each pair of nodes.

In our experiments, the density γ of the nodes in the network is kept constant and is equal to 8 nodes/km^2. In this case, N is varied between 160 and 4000 nodes. The mobility of nodes is simulated using either a random-walk model or real mobility patterns. Note that in the latter case, we used the mobility traces of taxicabs in San Francisco, CA, provided by Dartmouth University. These traces record the identity, the physical location (i.e., X, Y locations), and the time that a specific car has sent its updates for about 540 vehicles over a period of 30 days [31]. The parameter settings in our experiments are listed in Table III, where t_s denotes the simulation time, and V is the mobile node's velocity. According to each location-management policy, the location information updating and querying costs are measured.

In this paper, we considered several scenarios by varying the cell size (Figs. 7 and 8) and the network size (Figs. 9–15). In all figures, we can see that the analytical and simulation curves with respect to the 2-D random-walk model for the RLSMP almost coincide, which illustrates the accuracy of our models.

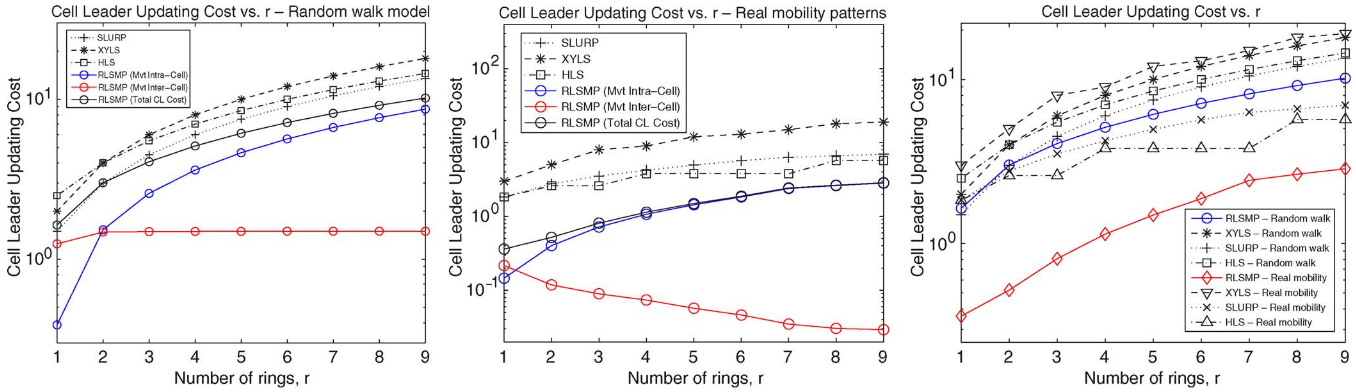
TABLE III
PARAMETER SETTINGS

Parameter	Value	Parameter	Value
t_s	1000 sec	$Time\_to\_Send$	30 $\sim$ 180 sec
T_r	750 m	γ	8 nodes/km^2
V	10 m/s	N	160 $\sim$ 4000
f_u	1/60	A	20 $\sim$ 590 km^2
f_q	1/60	r	1 $\sim$ 9

Fig. 7 plots the different CL update cost as a function of the number of cell element rings r under the 2-D random-walk model and real mobility patterns. In this case, r is varied between 1 and 9 to represent different cell sizes (i.e., 1×1 km, $\ldots, 6 \times 6$ km). Recall that the cell-side-length multiplicity n is equal to $(2r+1)$, and the cell side length is equal to nd. We can observe in Fig. 7 that the CL updates for all protocols increase with r since the cell size increases. The optimal cost of the CL updates depends on the CL position. Specifically, in the SLURP and the HLS, the CL is randomly chosen inside one cell, whereas the CLs of the XYLS are chosen to be all nodes that are located along the cell column. Hence, the CL updates are not optimal for these three protocols, as shown in Fig. 7. The RLSMP, on the other hand, reduces the update cost since the CL is located in the center of the cell. It is worth noting that the cost of CL updates in the RLSMP is equivalent to the cost of intracell movement when r is large, as it is a dominant cost. This is clearly shown in Fig. 7(b) when real mobility patterns are used. Indeed, the probability of crossing the cell boundary by a mobile node decreases with r, as shown in Fig. 8, which implies that the cost of intercell movements becomes nondominant compared with that of intracell movements for both mobility traces.

In addition, we can notice that the communication cost for all protocols (except for the XYLS) using real mobility patterns is lower than that for the random-walk model, as shown in Fig. 7(c). Moreover, from Fig. 8, we can see that the probability of crossing the cell boundary for the RLSMP is higher under the random-walk model than under real mobility patterns. This means that the random walk represents the worst-case scenario, as observed in [29], which illustrates the accuracy of our analytical model.

Fig. 9 depicts the LSC update cost (in terms of bytes) of all underlying protocols as a function of the network area A under the 2-D random-walk model. In this experiment, the network size is varied from 20 to 590 km^2. Like the CL update cost, the LSC update cost using the SLURP, the HLS, and the XYLS is higher than that of the RLSMP. The reason is that, in the SLURP, the HLS, and the XYLS, the distance between the mobile node and its location servers dramatically grows with the network size since the location servers are randomly chosen in the network for the SLURP and HLS cases and along the network column for the XYLS case. This allows a longer forwarding path to be formed and more byte transmissions. Note that for the HLS, the updating cost is less than that of the SLURP and the XYLS since it does not require the mobile node to update the faraway home regions unless it crosses the higher level boundary. On the other hand, when considering the RLSMP, the optimal values of the side length multiplicity n and the number of cells per cluster R are used for each value



(a) (b) (c)

Fig. 7. CL updating cost. (a) Random-walk mobility model. (b) Real mobility patterns. (c) Comparison between both mobility models.

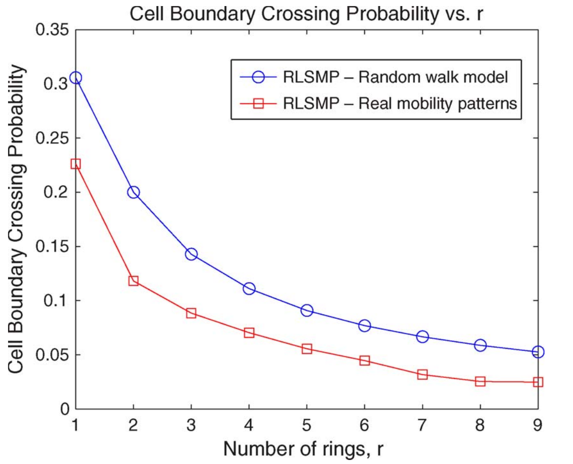


Fig. 8. Cell-boundary crossing probability.

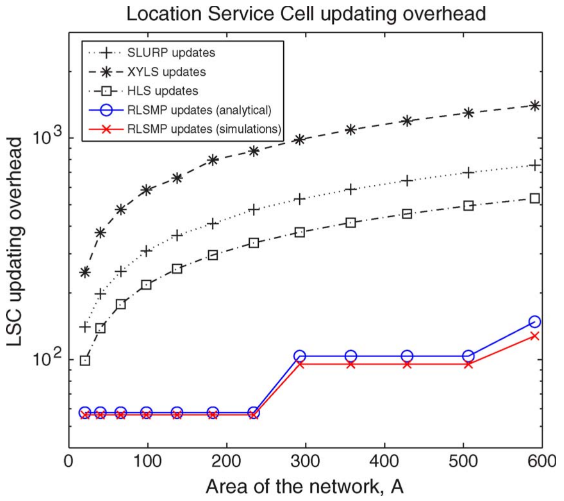


Fig. 9. LSC updating cost.

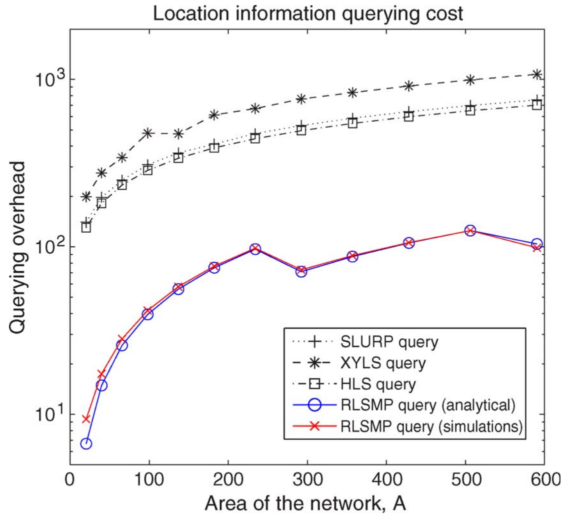


Fig. 10. Location-information query cost.

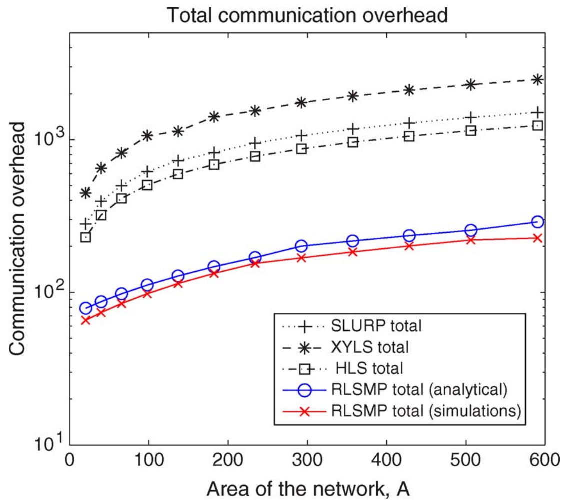


Fig. 11. Total communication overhead with optimal values of n and R.

of A. In this regard, the optimal solution for the optimization problem described in Section V that minimizes the total control overhead also minimizes the LSC updating cost term that is included therein.

Fig. 10 presents the location-information query of all protocols as a function of the network size under the 2-D random-walk model. In this experiment, we also varied the network size A from 20 to 590 km^2. For each value of A, we used the same values for n and R that minimize the total overhead. We can see in Fig. 10 that the RLSMP also reduces the location query cost since it incorporates the locality awareness. Indeed, the queries are first forwarded to the LSCs in the mobile node's

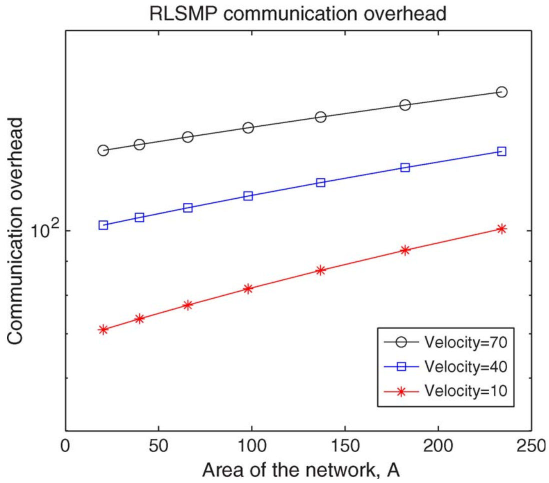


Fig. 12. Communication overhead with different nodes' velocity.

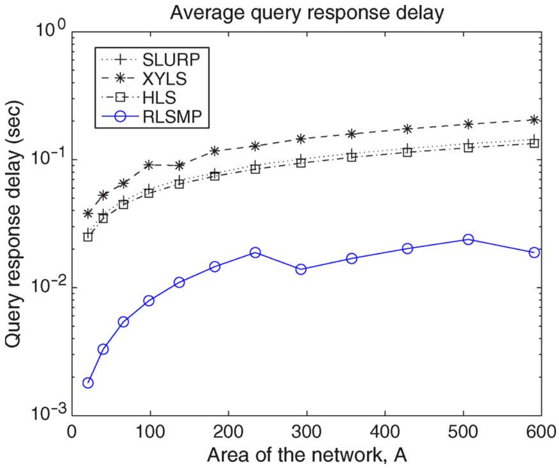


Fig. 13. Average query response delay.

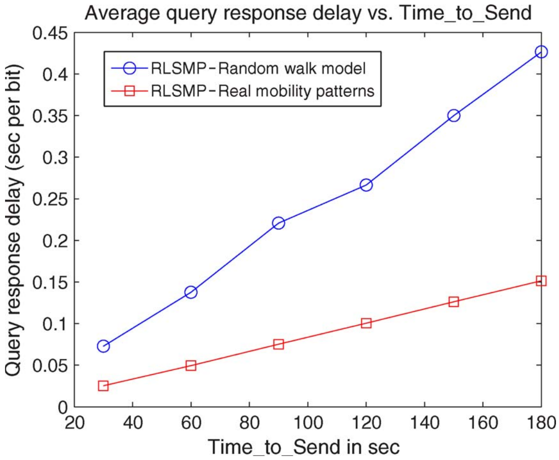


Fig. 14. Impact of $Time\_to\_Send$ on the average query response delay for the RLSMP.

vicinity. In the SLURP, the distance between the source and the destination's location server increases with the network size since it is assumed that any two nodes in the network are equally likely to communicate with each other. Considering the XYLS scheme, the node can frequently move in the whole network. This results in an increase in the distance between the mobile node and the quorum (i.e., the cell where updating and querying intersect) when the network size increases. Considering the HLS, it is worth noting that although the source and destination nodes are geographically close to each other, they are assumed to be located in different levels due to the virtual boundaries of the grid. This forces the query to travel to the destination pointers that are randomly located in the higher levels. As a result, the querying cost increases, as shown in Fig. 10.

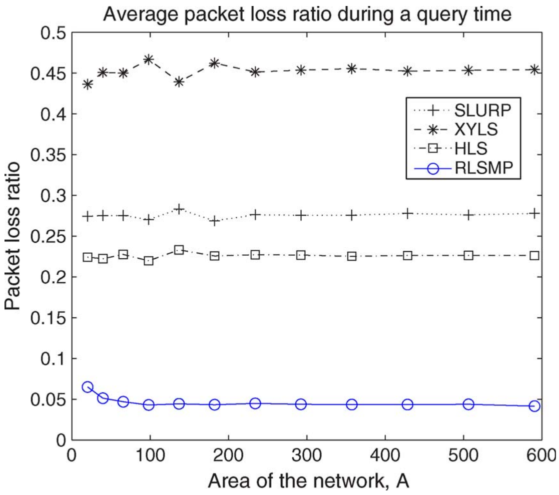


Fig. 15. Average packet loss ratio during a query time.

Fig. 11 compares the total overhead of all protocols when using our optimal RLSMP configuration. It confirms that the RLSMP stands out as the best solution from the communication overhead perspective. Fig. 12 shows the impact of the nodes' velocity V on the total communication overhead. We note that the increase in the nodes' velocity mainly affects the CL updating cost inside one cell since, in this case, the nodes will send more frequent updates to the CL. It is worth noting that, in our simulations, the nodes move with velocity that does not exceed 70 km/h in the urban areas, and then, the reelection of the CL occurs after 30 s, which is a reasonable time to transfer the location information to the new CL.

Fig. 13 shows the average query response delay of all protocols as a function of the network size under the 2-D random-walk model. In this experiment, $Time\_to\_Send$ is set equal to 60 s. Each mobile node randomly sends one query per $Time\_to\_Send$ interval time. We can see in Fig. 13 that the average response time per query for the RLSMP is lower than that of the remaining schemes since the CL will receive the response of all aggregated queries at the same time.

Fig. 14 shows the impact of the time schedule $Time\_to\_Send$ on the average query response delay for the RLSMP under both mobility models (i.e., the 2-D random-walk model and real mobility patterns). In this experiment, $Time\_to\_Send$ is varied between 30 and 180 s. From this figure, we can notice that the average query response delay increases with the increase in $Time\_to\_Send$ for both mobility models since the queries will be more delayed at the level of the CL node to perform message aggregation. In addition, we can notice again that the random-walk mobility model represents the worst-case scenario, as observed in [29].

Finally, Fig. 15 shows the average packet loss ratio during a query time using the optimal values of n and R. We can observe that the RLSMP yields the best performance compared with the other schemes. This is related to the aggregation process in the RLSMP since only a few aggregated packets will contend for the channel access at the media-access-control layer compared with the case in the SLURP, the XYLS, and the HLS. In addition, we can see that the packet loss ratio remains almost constant for all schemes. The reason is that, in our analysis, the network density is assumed to be constant.

VII. Conclusion

In this paper, we have introduced the RLSMP, which is a new location-service-management protocol that supports minimum overhead and locality awareness in VANETs. The RLSMP uses message aggregation that is enhanced by geographical clustering to reduce signaling overhead. It also resolves the localization of a destination node by using local search, which begins by exploring the vicinity of the source node. Thus, we avoid the relatively long distance signaling incurred in other protocols in both location updating and querying processes. Using both analytical and simulation approaches, we have compared the RLSMP scheme with existing solutions (the SLURP, the HLS, and the XYLS). To achieve this, first, we have developed analytical models to evaluate both the location updates and queries for a general 2-D random-walk model. In addition, simulations have been conducted using real mobility patterns to evaluate the performance of the protocol in real mobility situations. Second, we have investigated the optimal configuration of the RLSMP that minimizes the total signaling cost. It has been concluded that the RLSMP achieves substantial communication overhead reduction and improves the locality awareness when increasing the cell size as well as the network size. As such, the RLSMP stands out as a promising candidate for large-scale wireless ad hoc networks such as VANETs.

References

[1] J. Zhao and G. Cao, "VADD: Vehicle-assisted data delivery in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 57, no. 3, pp. 1910–1922, May 2008.

[2] T. Arnold, W. Lloyd, J. Zhao, and G. Cao, "IP address passing for VANETs," in *Proc. IEEE Int. Conf. PerCom*, Mar. 2008, pp. 70–79.

[3] A. Skordylis and N. Trigoni, "Delay-bounded routing in vehicular ad-hoc networks," in *Proc. ACM MOBIHOC*, 2008, pp. 341–350.

[4] S. Panichpapiboon and W. Pattara-atikom, "Connectivity requirements for self-organizing traffic information systems," *IEEE Trans. Veh. Technol.*, vol. 57, no. 6, pp. 3333–3340, Nov. 2008.

[5] J. Zhao, Y. Zhang, and G. Cao, "Data pouring and buffering on the road: A new data dissemination paradigm for vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3266–3277, Nov. 2007.

[6] Y. Toor, P. Muhlethaler, A. Laouiti, and A. Fortelle, "Vehicular ad hoc networks: Applications and related technical issues," *Commun. Surveys Tuts.*, vol. 10, no. 3, pp. 74–88, 3rd Quarter 2008.

[7] B. Ducourthial, Y. Khaled, and M. Shawky, "Conditional transmissions: Performance study of a new communication strategy in VANET," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3348–3357, Nov. 2007.

[8] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proc. ACM MOBICOM*, Rome, Italy, Jul. 2001, pp. 61–69.

[9] C. Perkins, *Ad Hoc Networking*. Reading, MA: Addison-Wesley, 2001.

[10] W. Kiess and M. Mauve, "A survey on real-world implementations of mobile ad-hoc networks," *Ad Hoc Netw.*, vol. 5, no. 3, pp. 324–339, Apr. 2007.

[11] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*. Berlin, Germany: Springer-Verlag, 1996.

[12] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward, "A distance routing effect algorithm for mobility (DREAM)," in *Proc. ACM MOBICOM*, 1998, pp. 76–84.

[13] C. Cheng, H. Lemberg, S. Philip, E. Berg, and T. Zhang, "SLALoM: A scalable location management scheme for large mobile ad-hoc networks," in *Proc. IEEE WCNC*, 2002, pp. 574–578.

[14] T. Camp, J. Boleng, and L. Wilcox, "Location information services in mobile ad hoc networks," in *Proc. IEEE ICC*, 2001, pp. 3318–3324.

[15] M. Ksemann, H. Fler, H. Hartenstein, and M. Mauve, "A reactive location service for mobile ad hoc networks," Dept. Sci., Univ. Mannheim, Mannheim, Germany, Tech. Rep. TR-02-014, Nov. 2002.

[16] W. Wang, F. Xie, and M. Chatterjee, "TOPO: Routing in large scale vehicular networks," in *Proc. IEEE VTC—Fall*, 2007, pp. 2106–2110.

[17] S. Woo and S. Singh, "Scalable routing protocol for ad hoc networks," in *Wireless Networks*, vol. 7. Norwell, MA: Kluwer, 2001, pp. 513–529.

[18] R. Flury and R. Wattenhofer, "MLS: An efficient location service for mobile ad hoc networks," in *Proc. ACM MOBIHOC*, Florence, Italy, May 2006, pp. 226–237.

[19] S. Das, H. Pucha, and Y. Hu, "Performance comparison of scalable location services for geographic ad hoc routing," in *Proc. IEEE INFOCOM*, Miami, FL, 2005, pp. 1228–1239.

[20] D. Liu, I. Stojmenovic, and X. Jia, "A scalable quorum based location service in ad hoc and sensor networks," in *Proc. IEEE MASS*, 2006, pp. 489–492.

[21] W. Kiess, H. Fssler, J. Widmer, and M. Mauve, "Hierarchical location service for mobile ad-hoc networks," *Comput. Commun. Rev.*, vol. 8, no. 4, pp. 47–58, Oct. 2004.

[22] B. N. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. ACM MOBICOM*, Boston, MA, Aug. 2000, pp. 243–254.

[23] H. Hartenstein and K. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 6, pp. 164–171, Jun. 2008.

[24] Y. Zang, L. Stibor, B. Walke, H. Reumerman, and A. Barroso, "Towards broadband vehicular ad-hoc networks—The vehicular mesh network (VMESH) MAC protocol," in *Proc. IEEE WCNC*, Hong Kong, Mar. 2007, pp. 417–422.

[25] I. F. Akyildiz, Y.-B. Lin, W.-R. Lai, and R.-J. Chen, "A new random walk model for PCS networks," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 7, pp. 1254–1260, Jul. 2000.

[26] K.-H. Chiang and N. Shenoy, "A 2-D random-walk mobility model for location-management studies in wireless networks," *IEEE Trans. Veh. Technol.*, vol. 53, no. 2, pp. 413–424, Mar. 2004.

[27] D. N. Alparslan and K. Sohraby, "Two-dimensional modeling and analysis of generalized random mobility models for wireless ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 15, no. 3, pp. 616–629, Jun. 2007.

[28] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: The single-copy case," *IEEE/ACM Trans. Netw.*, vol. 16, no. 1, pp. 63–76, Feb. 2008.

[29] F. Bai, N. Sadagopan, and A. Helmy, "The IMPORTANT framework for analyzing the impact of mobility on performance of routing protocols for adhoc networks," *Ad Hoc Netw.*, vol. 1, no. 4, pp. 383–403, Nov. 2003.

[30] G. Rubino and B. Sericola, "A finite characterization of weak lumpable Markov processes. Part II: The continuous time case," *Stoch. Process. Appl.*, vol. 45, pp. 115–125, 1993.

[31] M. Piorkowski, N. Sarafijanovoc-Djukic, and M. Grossglauser, "A parsimonious model of mobile partitioned networks with clustering," in *Proc. IEEE/ACM COMSNETS*, Bangalore, India, Jan. 2009.

[32] J. A. Barria and R. M. Lent, "MANET route discovery using residual lifetime estimation," in *Proc. IEEE ISWPC*, 2006, pp. 1–4.

Hanan Saleet (S'09) received the B.S. and M.S. degrees in industrial engineering from the University of Jordan, Amman, Jordan. She is currently working toward the Ph.D. degree with the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON, Canada.

Her research interests include performance evaluation and quality-of-service support in vehicular ad hoc networks and network design and optimization in wireless communications.

Otman Basir (M'09) received the B.Sc. degree in computer engineering from Al-Fateh University, Tripoli, Libya, the M.Sc. degree in electrical engineering from Queens University, Kingston, ON, Canada, and the Ph.D. degree in systems design engineering from the University of Waterloo, Waterloo, ON.

He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Waterloo. He is also the Associate Director of the Pattern Analysis and Machine Intelligence Laboratory. His research interests include intelligent transportation systems, embedded real-time systems, sensor networks, sensor and decision fusion, and biologically inspired intelligence.

Rami Langar (M'09) received the B.S. degree in telecommunications engineering from the Ecole Superieure des Communications, Tunis, Tunisia, in 2001, the M.S. degree in network and computer science from the University of Pierre and Marie Curie—Paris 6, Paris, France, in 2002, and the Ph.D. degree in network and computer science from the Ecole Nationale Superieure des Telecommunications de Paris in 2006.

In 2007 and 2008, he was with the School of Computer Science, University of Waterloo, Waterloo, ON, Canada, as a Postdoctoral Research Fellow. He is currently an Associate Professor of computer science with the University of Pierre and Marie Curie—Paris 6. His research interests include post-internet protocol architecture, mobility and resource management in wireless mesh and vehicular ad-hoc networks, performance evaluation, and quality-of-service support.

Raouf Boutaba (SM'01) received the M.Sc. and Ph.D. degrees in computer science from the University of Pierre and Marie Curie—Paris 6, Paris, France, in 1990 and 1994, respectively.

He is currently a Full Professor of computer science with the University of Waterloo, Waterloo, ON, Canada. His research interests include network, resource, and service management in multimedia wired and wireless networks.

Dr. Boutaba is the Founder and the Editor-in-Chief of the electronic publication IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT. He is also on the editorial boards of several other journals. He is currently a distinguished Lecturer of the IEEE Communications Society and the Chairman of the IEEE Technical Committee on Information Infrastructure and the International Federation of Information Processing Working Group 6.6 on Network and Distributed Systems Management. He has been a recipient of several best paper awards and other recognitions, including the Premier's Research Excellence Award.