

Proposal and Analysis of Region-based Location Service Management Protocol for VANETs

Hanan Saleet[¶], Rami Langar^{¶‡}, Otman Basir[¶], and Raouf Boutaba[¶]

([¶]) University of Waterloo ; 200 University Ave. W., Waterloo, ON, Canada

([‡]) LIP6, UMR 7606, University of Paris 6 ; 104 av. du President Kennedy, 75016 Paris, France

E-Mail: {hsaleet, rlangar, obasir, rboutaba}@uwaterloo.ca

Abstract—One of the major challenges for Vehicular Ad Hoc Networks (VANETs) is related to efficient location management issue. In this paper, we propose a new Region-based Location Service Management Protocol (RLSMP) that uses mobility patterns as means to synthesize vehicle movement and thus can be used in VANETs applications. One of the key distinguishing features of our solution from existing literature is its scalability since it uses message aggregation in both updating and querying, and promises locality awareness as well as minimum signaling overhead. To evaluate the efficiency of our proposal, we compare our scheme with existing solutions using both analytical and simulation approaches. To achieve this, we develop analytical models to evaluate the location updates cost. Numerical and simulation results show that our protocol scales better than existing schemes, when increasing the size of VANET which enhances the feasibility of such large scale ad hoc networks.

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 (MNs).

One of the main concerns in VANETs is the design of scalable and robust routing protocols. Studies in [1] have shown that geographic routing is a well-suited solution for the challenging task of routing in highly dynamic MANETs. Such routing protocol utilizes the location information of nodes available from positioning systems such as GPS to forward packets. As a prerequisite of geographic routing, each node in the network must be able to determine the position of the target node it wants to communicate with [2]. As such, a central challenge in geographic routing protocols has been the design of efficient location service management that can track the locations of MNs and reply to location queries for them.

Numerous proposals for location service management have been proposed for MANETs [3] – [11]. Two different approaches exist: flooding-based and quorum-based approaches. The former one involves global network flooding [5] [6], which results in severe performance degradation and scalability reduction. In the latter approach, the location servers (i.e., quorums) result from mapping the nodes' identifiers or geographical information to quorums in random or static way [9] – [11]. In view of this, XYLS [9] disseminates the location updating information in a direction such that a query can intersect an update quorum. However, this scheme assumes that every node will traverse the entire network frequently which is a rare case in VANET and will result in unnecessary use of the wireless link bandwidth. In [10], SLURP uses the concept of home

region to track the MN's mobility. All nodes in this region will act as location servers for the corresponding MN. However, the main concern of such a scheme is the lack of efficient locality awareness since the MN's home region is randomly assigned. The locality awareness is defined as the ability to support effectively local traffic patterns within the network. Authors in [11] propose a distributed location service to track the MNs position using a hierarchy of pointers. However, for a particular MN, the pointer assignment will be critically affected by the frequent crossing of cell boundary.

To overcome these limitations, we propose in this paper a scalable and locality awareness Region-based Location Service Management Protocol (RLSMP) designed for VANETs environment. RLSMP aims to investigate the effect of deploying a self organized framework for managing the location information using message aggregation enhanced by geographical clustering. To the best of our knowledge, RLSMP is the first protocol that achieves both properties (i.e., scalability and locality awareness) in VANETs.

To gauge the effectiveness of our proposal, we derive analytical expressions based on Markov chain for both location updates overhead and cell crossing boundary rate. We also investigate the location information query and updating by simulations. Numerical and simulation results show that our proposal can significantly reduce the location updates cost and also provide low querying overhead when compared to the most prominent existing schemes (XYLS [9] and SLURP [10]) under various scenarios.

The remainder of this paper is organized as follows. Section II describes our proposed protocol. Section III presents the analytical framework used to evaluate the communication overhead. In section IV, a comparison between our proposal and existing solutions is drawn using both analytical and simulation results. Finally, section V contains our concluding remarks.

II. REGION-BASED LOCATION SERVICE MANAGEMENT PROTOCOL (RLSMP)

In this section, we describe our new scheme called RLSMP. Recall that our aim is to provide a scalable location management protocol as well as incorporate locality awareness.

The key property of our proposal is that the location information is restricted in a geographical cluster in the network as shown in Fig. 1. Each vehicle automatically determines its geographical cluster while moving, without any additional communication or delay. Indeed, building clusters based on nodes' ID in VANET is often a daunting task due to problems arising from cluster management. Nevertheless, in RLSMP,

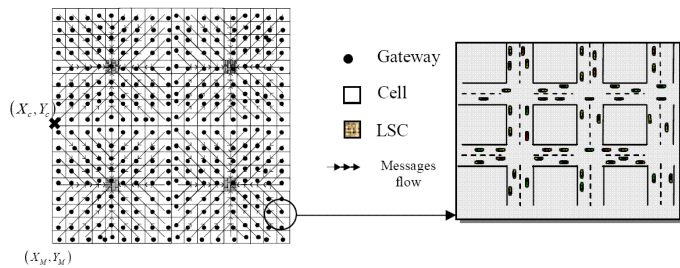


Fig. 1. System description and LSC updating process

using the location of nodes decreases the traffic overhead required for cluster maintenance. In addition, the geographical clustering suggests the possibility of message aggregation which is essential to decrease the communication overhead. Specifically, RLSMP uses message aggregation to avoid the drawbacks of individual location updates and querying messages. The messages are thus combined resulting in more efficient channel usage. For better understanding of our protocol functionalities, we consider the following network structure.

The vehicles 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 as shown in 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) which identifies its location relative to the grid origin. Each cell has a particular node called Cell Leader (CL) that is responsible of aggregating the location information about all nodes within the cell. Furthermore, the grid is divided into segments, and 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 81 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. The strategy of our proposed protocol is to denote the central cell of each cluster as location service management entity. Therefore, the nodes physically located in the central cell of one cluster are responsible for saving current location information about all nodes that belong to that cluster. This central cell is called the Location Service Cell (LSC).

RLSMP achieves scalability by the fact that location updates and querying messages are aggregated. The aggregated location information forwarded to the LSC describes a node in a specific cell instead of the detailed information about its exact location. The cell ID is used to identify the location of nodes. We believe 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 detail decreases with distance and time.

In RLSMP, the location updating cost involves two terms. The first term is regarding the messages sent by the MNs to the CL in order to update their locations within the cell. The second term is related to the LSC updates sent from the CLs to the LSC in order to update the location information of the

Algorithm 1 LSC 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);
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 towards the LSC;
14:  if (next Cell Leader is the LSC) then
15:    Stop;
16:  end if
17: end if

```

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 discussed thoroughly in the following sections.

A. CL Updating in RLSMP

As stated before, the CL tracks the mobility of nodes within the cell and keeps the MNs location information up-to-date. For each movement, the MN updates its location to the CL as follows. The decision of CL updating is based on the MN's location relative to the center of the cell. By analyzing this information, the MN can make a decision without consulting other nodes, which minimizes the overhead. The MN will send its current location information to the CL each time it crosses one cell element (see Fig. 3(a)). In this figure, we consider one cell with side length of l divided into 25 cell elements. Accordingly, the CL updating procedure is performed whenever the MN moves a distance d from its current location, where d depends on the transmission range T_r of the MN.

In addition, each MN is aware of the current cell boundary. Therefore, each time the MN crosses that boundary, 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 which triggers the renewal process as shown in section III.

It is worth noting that when the CL is about to leave the central cell element, it looks into its local table and chooses the MN that is closer to the center of the cell with minimum velocity V . This node will be selected as the new CL and will receive from the old CL all the stored information about the MNs located in the cell.

B. LSC Updating in RLSMP

The Location Service Cell (LSC) is defined as the central cell whose member nodes are responsible for keeping track

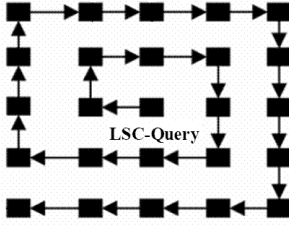


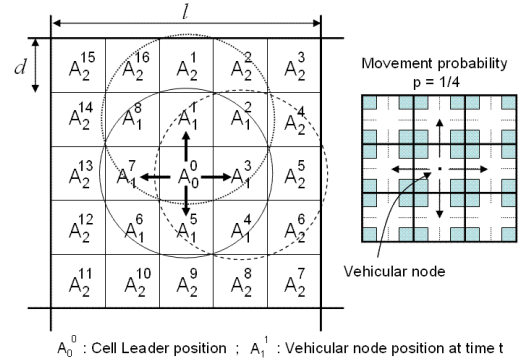
Fig. 2. Spiral shape of the location information retrieval

of all the MNs located in the cluster. Our protocol relies on aggregation and forwarding of location updating messages. This process is achieved by all CLs residing in the cluster and must be synchronized among them. Indeed, a time schedule, 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 of forwarding the aggregated packets of all MNs residing in the cell. Each CL stores detailed information about the MNs that it manages. This information contains node ID, X-Y coordinates of the node location, time of the last update, and velocity and direction of the node. At the same time the CL forwards a summarized (node ID and cell ID) information 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 flow of message aggregation, each square represents a cell with one CL, and the shaded cells represent LSCs. When a CL receives location information messages from another CL in its sub-trees, it collects and combines them into one aggregated message, which is forwarded to its parent until it reaches the LSC of the cluster. Such a message aggregation effectively suppresses the number of messages in the whole network. More formally, the LSC updating algorithm is described by the pseudocode in Algorithm 1.

C. Location Information Retrieval in RLSMP

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 location servers in the LSC of its cluster. If the destination is not registered in the same cluster, the query messages are not forwarded directly. Instead, the messages are delayed for a pre-specified time. This delay is essential for aggregating queries which are sent by vehicles residing in that cluster. The aggregated queries that are forwarded pass through the different LSCs as shown in Fig. 2. The aggregated querying messages are forwarded in a spiral around the LSC where this spiral visits all surrounding LSCs until it finds information about locations of the destinations. The nodes inside the LSCs will make use of the information stored in their own tables to find the required destinations IDs. Note that while visiting different LSCs, each message is relayed to its corresponding destination.

In the following, we present our framework for location management analysis. To do so, we develop Markovian models to estimate the communication overhead. The obtained results will be used, in a later stage, to derive the protocol performance metrics such as the cell boundary crossing rate and the CL location updates cost.



A_0^0 : Cell Leader position ; A_1^1 : Vehicular node position at time t

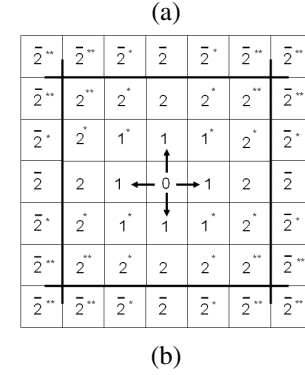


Fig. 3. Square-cell modeling approach. (a) Cell model with $R = 2$ rings, (b) Corresponding states aggregation

III. FRAMEWORK FOR LOCATION MANAGEMENT ANALYSIS

As stated before, we consider square-based 2-D model. Typically, a cell with side length of l is divided into $n \times n$ square cell elements as shown in Fig.3. The side length d of each cell element is determined such that each node n_i in the cell element i can communicate directly (i.e., in one hop) with a node n_j located in the adjacent cell element j . In this case, $d = T_r / \sqrt{5}$. In addition, the MN can move to one of the neighboring cell element with equal probability p ($p = 1/4$).

Fig. 3(a) represents an example of the cell model used in our analysis 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 MN'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 $i = 0, 1, \dots, R$ designate the i th ring away from the cell leader. 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 MN's location within the cell at time t . The sojourn time of a MN in each element A_i^j is assumed to be exponentially distributed with the 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 MN's position within the cell in order to predict its evolution. According to its next location, we can compute the CL location updates cost as well as the cell boundary crossing rate corresponding to our protocol.

The resolution of the Markovian chain $X(t)$, as defined above, is time-consuming. Moreover, this chain suffers from

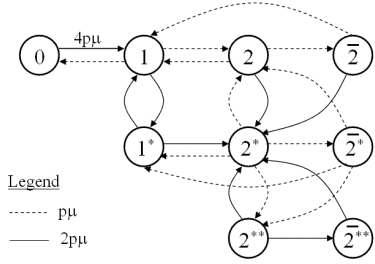


Fig. 4. State transition diagram of the aggregated Markov chain for $R = 2$

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 MN exhibits exactly the same behavior will be aggregated. Hence, the size of the state space E will be extremely reduced. To achieve this, we profit from the symmetric property of the 2-D model. Based on [12], we describe our algorithm to perform the state aggregation as follows.

- 1) Let A_i^j denote the cell element that contains the MN. As presented in figure 3(a), the state A_i^1 is chosen to be the one at the top of state A_{i-1}^1 . Afterwards, each ring i consists of $8i$ elements labeled in a clockwise direction as A_i^1, \dots, A_i^{8i} . Let A_i^{j*} denote the new aggregated state of the cell, where i always designates the ring reference, and j^* the state label inside the ring. Since all cells of the cluster have the same size, the aggregated states A_R^{j*} located at the ring R of the adjacent cells will be denoted by A_R^{j*} . These states represent the boundary states of the cell under consideration.

2)

Start with $i = 1$;

until $(i = R)$

Repeat{

$$\text{set } A_i^{0*} = A_i = \bigcup_{0 \leq j \leq 3} A_i^{2ij+1}$$

For $m = 1$ to $m = i$

$$\text{set } A_i^{m*} = \bigcup_{0 \leq j \leq 3} A_i^{2ij+m+1} + \bigcup_{1 \leq j \leq 4} A_i^{2ij-m+1}$$

$i = i + 1$;

}

For instance, for $R = 2$, we obtain the following aggregated states.

$$\begin{aligned} A_0^{0*} &= A_0 = \{A_0^0\}, \\ A_1^{0*} &= A_1 = \{A_1^1, A_1^3, A_1^5, A_1^7\}, \\ A_1^{1*} &= \{A_1^2, A_1^4, A_1^6, A_1^8\}, \\ A_2^{0*} &= A_2 = \{A_2^1, A_2^5, A_2^9, A_2^{13}\}, \\ A_2^{1*} &= \{A_2^2, A_2^4, A_2^6, A_2^8, A_2^{10}, A_2^{12}, A_2^{14}, A_2^{16}\}, \\ A_2^{2*} &= \{A_2^3, A_2^7, A_2^{11}, A_2^{15}\}. \end{aligned}$$

For ease of use, the aggregate states were assigned numbers, as follows [see Fig. 3(b)]:

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

Theorem:

Let $F = \{A_0, A_1, A_1^{1*}, \dots, A_i, A_i^{1*}, \dots, A_i^{j*}, \dots, A_R, A_R^{1*}, \dots, A_R^{R*}, \bar{A}_R, A_R^{1*}, \dots, A_R^{R*}\}$ designate the state space of the

new chain $Y(t)$ obtained by aggregation of the initial Markovian chain $X(t)$. We can demonstrate that the resulting aggregated process $Y(t)$ is also Markovian. Due to space limitation, we do not provide the proof of this result.

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 , ($i = 1, \dots, M$), where M is the set size (i.e., $M = (R + 1) \times (R + 4)/2$). Denote by Π_i and $\Pi_i^{(m)}$ ($i = (0, 1, \dots, R)$ and $m = (1, \dots, i)$) the stationary probability of the system for the aggregated state A_i and A_i^{m*} , respectively. Denote also by $\Pi_{\bar{R}}$ and $\Pi_{\bar{R}}^{(m)}$ $m = (1, \dots, R)$ the stationary probability of the system for the boundary states. The balance equations for the aggregated Markov chain are obtained recursively $\forall R \geq 2$ as follows:

$$\begin{cases} \Pi_0 = p\Pi_1 \\ \Pi_1 = 4p\Pi_0 + 2p\Pi_1^{(1)} + p\Pi_2 + \alpha p\Pi_{\bar{2}} \\ \forall 2 \leq i \leq R-1 \\ \quad \Pi_i = p\Pi_{i-1} + p\Pi_{i+1} + p\Pi_i^{(1)} + \beta p\Pi_{i+1} \\ \Pi_R = p\Pi_{R-1} + p\Pi_R^{(1)} + p\Pi_{\bar{R}}^{(1)} \end{cases} \quad (1)$$

$$\begin{cases} \Pi_1^{(1)} = 2p\Pi_1 + p\Pi_2^{(1)} + \alpha p\Pi_{\bar{2}}^{(1)} \\ \forall 2 \leq i \leq R-1 \\ \quad \Pi_i^{(i)} = p\Pi_i^{(i-1)} + p\Pi_{i+1}^{(i)} + \beta p\Pi_{i+1}^{(i)} \\ \Pi_R^{(R)} = p\Pi_R^{(R-1)} + p\Pi_{\bar{R}}^{(R-1)} \end{cases} \quad (2)$$

$$\begin{cases} \forall 2 \leq i \leq R-1, j=1 \\ \quad \Pi_i^{(1)} = 2p\Pi_i + p\delta\Pi_{i-1}^{(1)} + p\delta\Pi_i^{(2)} \\ \quad \quad \quad + p\Pi_{i+1}^{(1)} + \beta p\Pi_{i+1}^{(1)} \\ \forall 2 \leq j \leq R-2 \text{ and } j+1 \leq i \leq R-1 \\ \quad \Pi_i^{(j)} = p\Pi_i^{(j-1)} + p\delta\Pi_{i-1}^{(j)} + p\delta\Pi_i^{(j+1)} \\ \quad \quad \quad + p\Pi_{i+1}^{(j)} + \beta p\Pi_{i+1}^{(j)} \end{cases} \quad (3)$$

$$\begin{cases} \Pi_R^{(1)} = 2p\Pi_R + p\Pi_{R-1}^{(1)} + p\Pi_{\bar{R}}^{(2)} + 2p\Pi_{\bar{R}} + p\Pi_{\bar{R}}^{(2)} \\ \forall 2 \leq j \leq R-1 \\ \quad \Pi_R^{(j)} = p\Pi_R^{(j-1)} + p\theta\Pi_{R-1}^{(j)} + p\theta\Pi_{\bar{R}}^{(j+1)} \\ \quad \quad \quad + p\theta\Pi_{\bar{R}}^{(j+1)} + p\Pi_{\bar{R}}^{(j-1)} \end{cases} \quad (4)$$

$$\begin{cases} 3p\Pi_{\bar{R}} = p\Pi_{\bar{R}} \\ \forall 1 \leq j \leq R-1, \quad 3p\Pi_{\bar{R}}^{(j)} = p\Pi_{\bar{R}}^{(j)} \\ 2p\Pi_{\bar{R}}^{(R)} = 2p\Pi_{\bar{R}}^{(R)} \end{cases} \quad (5)$$

where

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

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

and

$$\sum_{i=0}^R \Pi_i + \sum_{i=1}^R \sum_{j=1}^i \Pi_i^{(j)} + \Pi_{\bar{R}} + \sum_{j=1}^R \Pi_{\bar{R}}^{(j)} = 1 \quad (6)$$

Given the balance equations ((1)-(5)) and the normalization equation (6), 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. Recall that the costs of

LSC updates and location information query will be derived by simulations.

A. Cell Boundary Crossing Rate

Let P_b denote the probability that a MN crosses the cell boundary when moving within the cell. Such situation happens when the MN 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. Based on the above analysis, this probability can be given by:

$$P_b = p \times (\Pi_R + \Pi_{\bar{R}} + \Pi_R^{(R)} + \Pi_{\bar{R}}^{(R)} + \sum_{j=1}^R (\Pi_R^{(j)} + \Pi_{\bar{R}}^{(j)})) \quad (7)$$

B. CL Location updates cost

Let C_u denote the cost of CL location updates when the MN moves within the cell. According to the node mobility, this cost can be written as follows:

$$CL_{updates} = Cost_{intra} + Cost_{inter} \quad (8)$$

where $Cost_{intra}$ and $Cost_{inter}$ denote, respectively, the signaling cost of location updates when the MN moves within the same cell (i.e., intra-cell movement) and when the MN crosses the cell boundary (i.e., inter-cell movement). Using the results of section III, the expressions of $Cost_{intra}$ and $Cost_{inter}$ are given as

$$\begin{aligned} Cost_{intra} = & \sum_{i=1}^{R-1} 2p(2i+1)\Pi_i + \sum_{i=1}^{R-1} \sum_{j=1}^i 4p(i+j)\Pi_i^{(j)} \\ & + p(3R+1)\Pi_R + 2p(2R-1)\Pi_R^{(R)} \\ & + \sum_{j=1}^{R-1} p(3R+3j-1)\Pi_R^{(j)} \end{aligned} \quad (9)$$

and

$$\begin{aligned} Cost_{inter} = & p(2R+1)(\Pi_R + \Pi_{\bar{R}}) + p(4R+1)(\Pi_R^{(R)} + \Pi_{\bar{R}}^{(R)}) \\ & + \sum_{j=1}^R p(2R+2j+1)(\Pi_R^{(j)} + \Pi_{\bar{R}}^{(j)}) \end{aligned} \quad (10)$$

IV. NUMERICAL AND SIMULATION RESULTS

In this section, we compare our proposal with respect to XYLS [9] and SLURP [10] schemes through both simulations and analytical approaches. To evaluate the cost of LSC location updates and location information query by simulations, we developed our own discrete-event simulator.

The simulation environment consists of a large scale wireless ad hoc network with 81 cells placed in a 9×9 grid. 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 equals to 8 nodes/km². In this case, N is varied between 650 and 6000 nodes. The mobility of nodes is simulated using a random walk model [12]. According to each location management policy, the average CL updates, the LSC

TABLE I
PARAMETER SETTINGS

Parameter	Value	Parameter	Value
t_s	1000 sec	$Time_to_Send$	60 sec
V	10 m/s	T_r	750 m
R	1 ~ 9	γ	8 nodes/km ²
Q	2	$Cluster_Size$	$3 \times 3 \sim 9 \times 9$

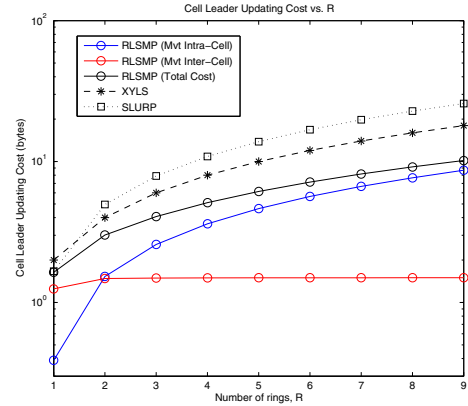


Fig. 5. Cell Leader updating cost

location updates and the location information query costs are calculated. We considered several scenarios by varying the cell size (Fig. 5), the cluster size (Fig. 7), and the network size (Fig. 8). The parameter settings in our experiments are listed in table I, where t_s denotes the simulation time. V is the MN's velocity, and Q is the number of queries per node during the simulation time.

Fig. 5 plots the different CL updates cost as a function of the number of rings R . In this case, R is varied between 1 and 9 to represent different cell size (i.e., 1 km \times 1 km, ..., 6 km \times 6 km). Recall that the side length l of a cell is equal to $(2R+1) \times d$, where $d = T_r/\sqrt{5}$. We can observe in Fig. 5 that the CL updates for all protocols increase with R , since the cell size increases. Hence the average distance between the MN and the CL increases with R . In view of this, the optimal cost of the CL updates depends on the CL position. Specifically, in SLURP, the CL is chosen randomly inside one cell whereas the CLs of XYLS are chosen to be all nodes located along the cell column. Hence, the CL updates are not optimal for these two protocols, as shown in Fig. 5. RLSMP, on the other hand, reduces the updates cost since the CL is located in the center of the cell. It is worth noting that the cost of CL updates in RLSMP is equivalent to the cost of intra-cell movement when R is large as it is a dominant cost. Indeed, the probability of crossing the cell boundary by a MN decreases with R as shown in Fig. 6, which implies that the cost of inter-cell movements becomes non-dominant compared to that of intra-cell movements.

Fig. 7 depicts the LSC updates cost (measured in number of bytes) of all underlying protocols as a function of the cluster size. In this experiment, the cluster size is varied between 3×3 and 9×9 cells placed in a grid. The side length of each cell is fixed and equals to 1 km. Likewise the CL updates cost, the LSC updates using SLURP and XYLS is higher than that of RLSMP. The reason is that, in both SLURP and XYLS, the distance between the MN and its location servers grows dramatically

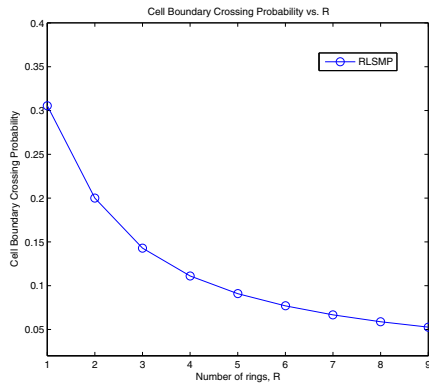


Fig. 6. Cell Boundary crossing probability

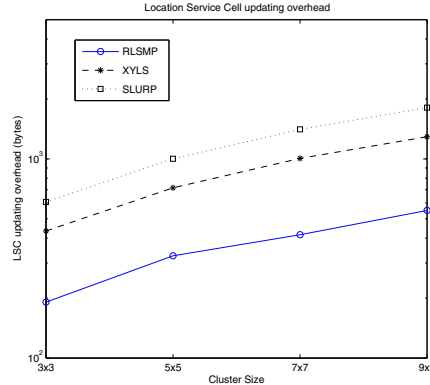


Fig. 7. LSC updates cost

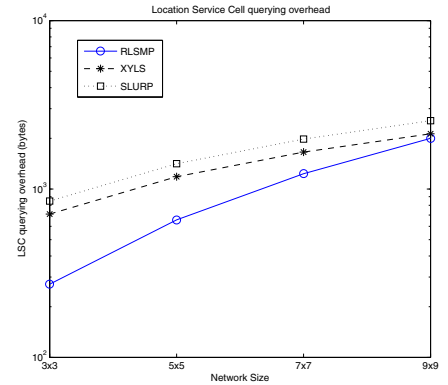


Fig. 8. Location information query cost

with the cluster size since the location servers are randomly chosen in the cluster for the SLURP case and along the cluster column for the XYLS case. This allows longer forwarding path to be formed and more byte transmissions. However, in RLSMP, even if the cluster size increases, there is no dramatic increase in the overhead since the LSC is located at the center of the cluster. In addition, our protocol employs message aggregation using a tree structure for the forwarding process with the LSC as a root. This avoids the high volume of overhead produced by each MN to update its location inside the cluster.

Fig. 8 shows the location information query of all protocols as a function of the network size. In this experiment, we considered different number of clusters varied between 3×3 and 9×9 . The cluster size is kept constant and includes 3×3 cells with side length equals to 1 km. This allows a gradual extension of the network. We can see in Fig. 8 that RLSMP reduces the location queries cost, especially when the network size is relatively small. Specifically, in SLURP, the distance between the source and the destination's location server increases with the network size since it is assumed that any two MNs in the network are equally likely to communicate with each other. Considering the XYLS scheme, the node can move in the whole network frequently. This results in an increase of the distance between the MN and the quorum (i.e., the cell where the updating and querying intersect) when the network size increases. On the other hand, RLSMP incorporates the locality awareness by forwarding the queries to the LSCs in the MN's vicinity. Note that, in RLSMP, the cost of location queries is higher than the cost of LSC updates since the queries path, which follows a spiral around the LSC, grows significantly with the network size as shown in Fig. 8.

Therefore, the RLSMP scheme stands out as the best location service management from the communication overhead perspective. This enables RLSMP scheme to outperform the remaining protocols although an extra time is needed to answer long distance queries due to the use of the spiral technique. Note that the reduction of the number of transmissions by the in-network aggregation happens at the cost of longer delay, since packets need to be buffered and processed in the CL. However, in RLSMP, the delay to answer a query can be reduced when the local communication pattern becomes more frequent, which is the case in VANETs.

V. CONCLUSION

This paper described a new location service management protocol, called RLSMP that supports scalability and locality awareness in VANETs. Our proposal uses message aggregation enhanced by geographical clustering to reduce the signaling overhead in the network. It also resolves the localization of a destination node by using local search, which begins by exploring the vicinity of the source node. Doing so, we avoid the relatively long distance signaling incurred in the other protocols in both location updating and querying. Using both analytical and simulation approaches, we compared our proposed scheme with existing solutions (XYLS and SLURP). To achieve this, we developed analytical models to evaluate the location updates cost. We found that our protocol achieves substantial communication overhead reduction and improves the locality awareness when increasing the network size. As such, our scheme stands out as a promising candidate for large scale wireless ad hoc network such as VANETs.

REFERENCES

- [1] B. N. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", in Proc. ACM MobiCom'00, pp. 243-254, Boston, Massachusetts, Aug. 2000.
- [2] W. Kiess, H. Fssler, J. Widmer, M. Mauve, "Hierarchical Location Service for Mobile Ad-Hoc Networks", ACM MC2R, vol. 8, no. 4, Oct. 2004.
- [3] W. Kiess and M. Mauve, "A survey on real-world implementations of mobile ad-hoc networks", Ad Hoc Networks, vol. 5, no. 3, Apr. 2007.
- [4] R. Friedman and G. Kliot, "Location Services in Wireless Ad Hoc and Hybrid Networks: A Survey," TR CS-2006-10, Department of Computer Science, The Technion. April 2006.
- [5] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks", in Proc. of ACM MobiCom, October 1998.
- [6] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A distance routing effect algorithm for mobility (DREAM)", in Proc. of ACM MobiCom, Oct. 1998.
- [7] J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing", in Proc. of ACM MobiCom, August 2000.
- [8] 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.
- [9] D. Liu, I. Stojmenovic and X. Jia, "A scalable quorum based location service in ad hoc and sensor networks," in Proc. IEEE MASS, 2006.
- [10] S. Woo, S. Singh, "Scalable routing protocol for ad hoc networks", Wireless Networks, vol. 7, pp. 513-529, Kluwer Publishers, 2001.
- [11] R. Flury and R. Wattenhofer, "MLS: an efficient location service for mobile ad hoc networks", ACM MOBIHOC, Italy, May 2006.
- [12] K.-H. Chiang, N. Shenoy, "A 2-D Random-Walk Mobility Model for Location-Management Studies in Wireless Networks", IEEE Trans. Vehicular Technology, Vol. 53, No. 2, pp. 413-424, Mar. 2004.