Scalable and Efficient Dual-Region based Mobility Management for Ad Hoc Networks

Ing-Ray Chen[1], Yinan Li[1], Robert Mitchell[1], and Ding-Chao Wang[2]

[1] *Department of Computer Science, Virginia Tech, Northern Virginia Center*
7054 Haycock Road, Falls Church, VA 22043, United States
{*irchen,yinan926,rrmitche*}*@vt.edu*
[2] *Department of Information Management*
Southern Taiwan University
Tainan, Taiwan
dcwang@mail.stut.edu.tw

Abstract

In this paper, we propose and analyze the design notion of dual-region mobility management (DrMoM) for achieving scalable, efficient location management of mobile nodes in ad hoc networks. The basic idea is to employ local regions to complement existing home region based location service schemes that assign a home region through hashing to a mobile node and have mobile nodes in both the home and local regions serve as location servers for that node. The most salient feature of DrMoM is that the optimal home region size and local region size can be dynamically determined *per mobile user* based on mobility and service characteristics of individual mobile nodes to minimize the overall network cost incurred by location management and data packet delivery. Moreover, DrMoM is completely distributed. Each node determines its optimal home region size and local region size autonomously. A performance analysis is performed to demonstrate the benefit of DrMoM over existing region-based location management schemes.

Keywords: Mobility management; region-based location management; mobile ad hoc networks; performance analysis.

1. Introduction

A mobile ad hoc network (MANET) is a self-configuring network, in which mobile nodes form and maintain a dynamic network topology without a fixed infrastructure. In this paper, we will use the terms location management and mobility management interchangeably as there is no fixed infrastructure in MANETs for handoff management and the central issue for mobility management is location management. While location management research is well developed for wireless mesh networks [1], cellular networks [2, 3, 4, 5], and Mobile IP networks [6], scalable location management for MANETs is an open issue [7].

In this paper, we propose and analyze a scalable, efficient mobility management scheme for MANETs called **D**ual-**r**egion **M**obility **M**anagement (DrMoM) based on the idea of employing local regions to complement existing home region based location service schemes in MANETs that assign home regions to mobile nodes and have mobile nodes in both the home and local regions of a mobile node serve as location servers for that node. Relative to existing work utilizing home region based location service [8, 9, 10, 11, 12] and local region based location service [7, 13, 14, 15, 16, 17, 18], our contribution is to dynamically determine the optimal home region size and local region size for *each* mobile node based on the mobile node's runtime mobility and service characteristics to minimize network cost.

DrMoM is based on the design notion of integrated mobility and service management for network cost minimization [19]. Specifically, unlike existing location services that define the home region size statically at design time, DrMoM dynamically determines the optimal home region size and local region size (defined by their respective radii denoted by R_h and R_l), which together minimize the overall network cost incurred by location management and data packet delivery. Moreover, DrMoM is completely distributed. Each node determines its optimal home region size R_h and local region size R_l autonomously. We develop a performance model for deriving the optimal values of the two key design parameters R_h and R_l and for calculating the overall network cost incurred by DrMoM, given system parameters characterizing the mobility and service characteristics of mobile nodes. To demonstrate the benefit of our dual-region location management scheme, we compare location-based routing based on DrMoM against a well-known scheme called SLURP [9] based on static home regions as well as a region-based location management scheme called RUDLS [16] which claims to outperform contemporary region-based location management schemes. We show that DrMoM outperforms both SLURP and RUDLS in terms of the overall network cost incurred.

The paper is organized as follows. Section 2 surveys related works and contrasts DrMoM with existing approaches for mobility management in MANETs. Section 3 describes our scalable design for DrMoM. Section 4 presents a performance model for analytically evaluating the performance of DrMoM. Section 5 performs a comprehensive performance evaluation, focusing on the effect of various parameters on the performance of DrMoM, as well as a comparative performance analysis of DrMoM against SLURP and RUDLS. Section 6 performs simulation validation of the analytical results and tests the sensitivity of the results with respect to node distribution and node mobility patterns. Section 7 discusses the applicability. Finally Section 8 summarizes the paper and outlines future research areas.

2. Related Work

In contrast to other types of wireless networks such as cellular networks, IP networks, and wireless mesh networks, a MANET lacks pre-existing network infrastructures. Therefore, mobility management schemes proposed for other types of networks are generally not appropriate for MANETs.

A recent study [20] reveals that hierarchical region-based location management [7, 13, 14, 15, 16] is the most promising location management scheme for achieving scalability and efficiency.

A prevalent region-based location service in MANETs is *hashing-based* with which each mobile node is assigned a *home region* through hashing [8, 9, 10, 11, 12]. The nodes in the home region serve as *location servers* for that mobile node. A mobile node sends location updates to its location servers when it moves. To locate a destination node, a source node sends a location query to the destination node's home region location servers. Although a hashing-based location service is highly scalable, it has a major drawback: a source node has to contact the location servers of the destination node regardless of how close it is away from the destination node. If the two nodes are close to each other, contacting the location servers which may be far away geographically incurs unnecessary overhead. One way to solve this problem is to have a mobile node periodically exchange up-to-date location information with neighboring nodes in a *local region* [17, 18]. If some node in the local region of the source node knows the location of the destination node, the source node can locate the destination node utilizing only local location information from the neighboring nodes, without having to query the destination node's home region. It is also possible that the source node is within the local region of the destination node and therefore knows where the destination node is located using only local location information it keeps.

Among the above protocols cited, SLURP [9] and RUDLS [16] are introduced in more detail below as they are selected as baseline schemes against which DrMoM is compared in this paper. The reason we select these two schemes is that SLURP represents the most original work in region-based location management and RUDLS is a very recent location management protocol proposed which claims to outperform existing region-based location management protocols.

SLURP [9] handles location management using a scalable location service based on statically partitioned and assigned home regions. When a mobile node moves, it updates its location with the location servers in its home region by sending location update messages. To locate a destination mobile node D, the node's home region is queried to locate the current region in which D resides. Geographical routing is used to forward a data packet sent to D towards the center of the current region of D. When the data packet arrives at the first node within the current region, Dynamic Source Routing (DSR) is employed to deliver the data packet to D within the region. SLURP defines the region size statically when the coverage area of a MANET is partitioned into grids, each of which corresponds to a region. There are two major differences between DrMoM and SLURP:

- Although both DrMoM and SLURP assign each mobile node to a static home region center through hashing, the home region size in SLURP is fixed while the home region size in DrMoM varies dynamically in order to minimize the location query and update cost.

- SLURP does not use local regions for location query. As a result, SLURP incurs a high query cost because it always queries home region location servers for the location of the mobile user. DrMoM uses local regions for location query to save location query location overhead because it will query local region location servers first before querying the home region location servers for the location of the mobile node. Moreover, DrMoM assigns each mobile node to a local region which can change in size and location, depending on the mobility and service characteristics of the mobile node in order to minimize the location query cost.

RUDLS [16] on the other hand is a region-based hierarchical location management scheme consisting of level 1 and level 2 location servers. Each level 1 location server keeps track of the locations of mobile users in its region each covering 9 grids. When a mobile user moves from one grid to another grid within the same region, only the location database of the level 1 location server is updated. On the other hand, each level 2 location server covers a number of level 1 location servers (e.g., 9 $\times$ 9 grids). When a mobile user moves from one level 1 region to another level 1 region, the location database of the level 2 location server is updated. Finally, when a mobile user moves from one level 2 region to another level 2 region, all level 2 location servers are updated with the user's new location, which is an expensive location update operation. A location query always goes bottom-up, i.e., it will go from the local level 1 location server and if necessary to the local level 2 location server, and if necessary, to a remote level 2 location server. In contrast, DrMoM only maintains location servers in the dynamically adjustable local and home regions for a mobile node for efficiency.

3. Dual-Region Mobility Management

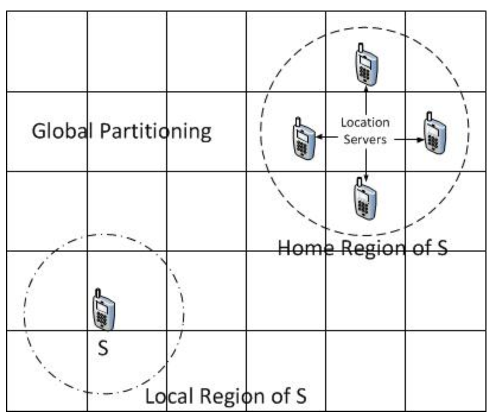

Figure 1: Global partitioning of the MANET coverage area into rectangular regions.

Like SLURP, DrMoM is highly scalable through hashing. The coverage area of a MANET is statically partitioned into equally sized rectangular regions, as shown in Fig. 1. This global partitioning of the MANET coverage area is used as the basis for home region assignment. Specifically, each mobile node is permanently assigned a *home region*, whose center co-locates with the center of one of the rectangular regions, as illustrated by Fig. 1. The assignment is calculated by hashing the unique ID of the mobile node (e.g., its IP or MAC address) to the ID of one of the rectangular regions. We assume that every mobile node has knowledge about the global partitioning as well as the hash function such that it is able to locate the center of the home region of any node. All mobile nodes within the home region of a mobile node serve as *home region location servers* for that node. DrMoM varies the home region size dynamically based on the mobile node's runtime mobility and service characteristics. The home region size can be expanded as needed to ensure that at least one node exists to serve as the location server. We assume that node distribution (e.g., random or city-style) is a predefined knowledge known to every node, so every node knows how far R_h should be in order to cover at least one node from the center of its home region.

Besides the home region, each mobile node is also associated with a *local region*, and it exchanges location information with neighbors in the local region. Unlike the home region, which does not move, the local region moves with the mobile node. The home region keeps location summary information of the node, i.e., the coordinate of the center and radius of the node's local region. Whenever the local region moves due to movement of the node, the location servers in the home region are updated with the location summary information. To locate the local region of a destination node, the source node sends a location query to the destination node's location servers.

The coordinates of the center of a home region is statically determined, whereas the radius is dynamically determined on a per-node basis, depending on the node's mobility and service characteristics. The *home region size*, determined by its radius denoted by R_h, is a key factor balancing the trade-off between the overhead for location queries/updates and the robustness of the location service. Specifically, a larger home region covers more location servers on average and consequently increases the chance of a successful location query. However, a larger home region also leads to larger overhead for location queries and updates. Because R_h is dynamic, the size of the home region is dynamic and not necessarily restricted by the size of the rectangular region. The *local region size*, determined by its radius denoted by R_l, is also a key parameter. Increasing the local region size increases the chance that a destination node is located using local location information, without querying the location servers. However, as the local region size increases, the cost of location query packet delivery increases because there are more hops to travel in a larger local region. The local region size also impacts on the rate of location updates to the home region, which is equal to the rate of local region boundary crossing.

Each mobile node maintains two location tables: the local region location table LT_l that stores location information of nodes for which it serves as a local region location server, and the home region location table LT_h that stores location information of nodes for which it serves as a home region location server. LT_l is updated whenever the mobile node receives a "local region location update" by which the exact location of the node within its local region is updated, whereas LT_h is updated whenever it receives a "home region location update" by which the node's new local region defined by the local region's (center, radius) is updated. More specifically, an entry in LT_l keeps the corresponding node's "exact" location obtained from the most recent local region location update from that node. An entry in LT_h stores the coordinates of the center and radius of the corresponding node's local region obtained from the most recent home region location update from that node. A timestamp is associated with each entry in the tables to indicate its freshness and is copied into the header of data packets when the entry (for the destination) is used by the source node for data packet delivery. Expired table entries are deleted periodically to make room for new entries.

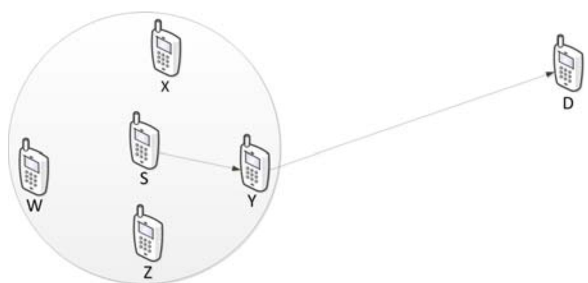

Figure 2: Greedy geographical packet forwarding.

DrMoM uses geographical routing to route data packets and control messages such as messages for location updates and queries, as illustrated in Fig. 2. For each hop, DrMoM selects the node from the one-hop neighbors of the current node that is *closest to the destination* (i.e., the node that makes the most progress towards the destination) as the next forwarding node. For example, in Fig. 2, node Y is selected by the source S as the next forwarding node because it is closest to the destination D among the neighbors of S. By selecting the next forwarding node this way, DrMoM guarantees that progress is made towards the destination for each hop, finally leading to the destination.

In the following sections, we present DrMoM in detail for location update, location query, data packet delivery, and home region maintenance operations. Table 1 lists the notations used.

3.1. Location Update

A mobile node uses a local region location update to notify neighbors within its local region of the coordinates of its current location as reported by the GPS module, and a home region location update to inform its location servers of the coordinates of the center and radius of its local region.

3.1.1. Local Region Location Update

Local region location updates follow a threshold-based approach. Specifically, a mobile node broadcasts a location update to its neighbors within its local region, when the distance between its current location and the location where the last update was triggered exceeds a threshold τ. Each mobile node S maintains a variable $S.loc\_last\_update$ that records the location where the last local region location update was performed. Given a chosen value of τ, the frequency of local region location updates depends on the *mobility rate* of S [21]. In this paper, τ is set to be equal to the wireless transmission range such that the difference between the location of a mobile node kept by neighbors in its local region and its actual location is never larger than the wireless transmission range. Note that because the local region of S is not restricted to its one-hop transmission range, a neighbor could potentially be multiple hops away.

The local region location update carries the following information about S: its node ID, current location ($S.curr\_location$), moving speed, and moving direction as reported by the GPS module. Whenever a local region location update is triggered, the radius of S's local region, i.e., R_l, is re-calculated and the up-to-date value is also carried by the location update. This information is necessary for neighboring nodes to dynamically determine if they are within S's local region, since the value of the radius may change over time, depending on S's mobility rate and service characteristics. Each neighboring node receiving the location update determines if it is within S's local region by comparing its distance to the current center of S's local region against R_l. If its is within S's local region, it updates its local LT_l, and rebroadcasts the location update. If a neighboring node determines that it is outside S's

Table 1: The notations used in this paper.

Notation	Meaning
S.id	the node ID of S
S.curr_location	current location of S
(S.hr_center, S.hr_radius)	center and radius of S's home region
(S.lr_center, S.lr_radius)	center and radius of S's local region
loc_info.ts	timestamp of location information stored in *loc_info*
S.loc_last_update	location where the last local region location update was triggered
local_loc_update	local region location update message
home_loc_update	home region location update message
local_loc_query	local region location query message
home_loc_query	home region location query message
NB(S)	list of one-hop neighbors of S
S.LT(D)	result of lookup for D in S's location table (LT)
dist(a, b)	distance between location a and location b
m.Header_Loc_Info(D)	location information of D carried in the header of packet m
S.Forward_Packet(m, NB(S))	procedure for S to forward a packet m to the next hop towards the destination
S.Broadcast(m, NB(S))	procedure for S to broadcast m to its one-hop neighbors
S.Send_Reply(D, m)	procedure for S to send the reply m to D
S.Update_Entry(LT(D), m)	procedure for S to update the entry for D in LT using location information carried by m
S.Update_Header(S, loc_info)	procedure for S to update the data packet header using *loc_info*
S.Calculate_Radius(selector)	procedure for S to dynamically calculate the radius for the local region or home region

local region, it simply drops the location update. Algorithm 1 describes the procedure for processing a local region location update.

```
if dist(S.loc_last_update, S.curr_location) ≥ τ then
    S.loc_last_update = S.curr_location;
    S.Calculate_Radius(S.R_l);
    S.Broadcast(local_loc_update, NB(S));
    foreach neighboring node i do
        /* if i is within S's local region */
        if dist(i.curr_location, S.curr_location) < S.R_l then
            i.Update_Entry(i.LT_l(S), local_loc_update);
            i.Broadcast(local_loc_update, NB(i));
        end
        else
            i drops local_loc_update;
        end
    end
end
```

Algorithm 1: Processing a local region location update.

3.1.2. Home Region Location Update

A home region location update is triggered whenever a mobile node S moves outside its current local region. Specifically, when S detects that it has moved outside its current local region (by comparing its current location with the center of its local region), it calculates the radius R_l for the new local region and sends a home region location update to the location servers in its home region. The location update carries the coordinates of the center and radius of S's new local region. The radius of S's home region, i.e., R_h, is also re-calculated based on S's mobility and service characteristics every time a home region location update is triggered, and is also carried by the location update. This information is necessary for other nodes to dynamically determine and mark if they are within S's home region and serve as location servers for S by comparing their distances to the center of S's home region with R_h.

The first location server within the home region receiving the location update broadcasts the update in the home region, as in [9]. Each subsequent node receiving the location update checks if it is within S's home region by comparing its location to the center of S's home region (the node obtains this information by applying the hash function to S's node ID) against the radius R_h carried by the location update. If the node is within the home region, it updates its local LT_h and rebroadcasts the location update. If the node determines that it is outside S's home region, it simply drops the location update. To reduce bandwidth and energy consumption and network congestion, each location server only broadcasts the first received update after waiting for a random amount of time.

Given a chosen value of R_l, the frequency of home region location updates depends on S's mobility rate. Algorithm 2 gives the procedure for processing a home region location update.

3.2. Location Query

A location query is a two-stage procedure: the first stage is a local region location query, optionally followed by a home region location query in the second stage if no location replies have been received in the first stage.

3.2.1. Local Region Location Query

In the first stage, the source node S broadcasts a location query within its local region, hoping that some neighboring nodes have the up-to-date location information of the destination node D. In the meantime, S starts a timer that expires after approximately the time for a round-trip transmission between S and the furthest neighbor in its local region (to ensure that location replies if any are received before the timer expires). The expiration of the timer before any location reply is received indicates a failed local region location query. The local region location query carries the current coordinates of the center and R_l of S's local region for neighboring nodes to dynamically determine if they are within S's local region.

Any neighboring node in S's local region that finds valid location information of D in either LT_l or LT_h sends a location reply to S. The location reply carries the timestamp of the table entry for D indicating the freshness of the location information. Upon the expiration of the timer, S collects all replies and uses the one with the most recent location information of D. To prevent a reply storm from happening, each neighbor waits for a random amount of time before sending out the reply [9]. The procedure for processing a local region location query is presented in Algorithm 3.

```
if dist(S.curr_location, S.lr_center) ≥ S.R_l then
    S.Calculate_Radius(S.R_l);
    S.lr_center = S.curr_location;
    S.Calculate_Radius(S.R_h);
    S.Forward_Packet(home_loc_update, NB(S));
    foreach intermediate node i do
        /* if i is the first location server of S receiving the update */
        if dist(i.curr_location, S.hr_center) < S.R_h then
            i.Update_Entry(i.LT_h(S), home_loc_update);
            i.Broadcast(home_loc_update, NB(i));
            /* Break out of the loop */
            Break;
        end
        else
            i.Forward_Packet(home_loc_update, NB(i));
        end
    end
    foreach node j receiving the broadcast location update do
        /* if j is within S's home region */
        if dist(j.curr_location, S.hr_center) < S.R_h then
            j.Update_Entry(j.LT_h(S), home_loc_update);
            j.Broadcast(home_loc_update, NB(j));
        end
        else
            j drops home_loc_update;
        end
    end
end
```

Algorithm 2: Processing a home region location update.

```
S.Broadcast(local_loc_query, NB(S));
foreach neighboring node i do
    /* if i is within S's local region */
    if dist(i.curr_location, S.lr_center) < S.R_l then
        if i.LT_l(D) ≠ null OR i.LT_h(D) ≠ null then
            i.Send_Reply(S, local_loc_reply);
        end
        i.Broadcast(local_loc_query, NB(i));
    end
    else
        i drops local_loc_query;
    end
end
S picks local_loc_reply with the latest timestamp;
S.Update_Table(S.LT_l(D), local_loc_reply);
```

Algorithm 3: Processing a local region location query.

3.2.2. Home Region Location Query

A home region location query is triggered in the second stage if the local region location query fails to locate D in the first stage. Specifically, S locates the home region of D by applying the hash function to D's node ID and sends a location query towards the center of the home region of D using geographical routing. The location query will ultimately reach a location server within the home region of D, which retrieves the entry for D in its LT_h, and sends a location reply back to S. Algorithm 4 describes the procedure for processing a home region location query.

3.3. Data Packet Delivery

Suppose the source node S has a data packet m to send to the destination node D. S needs to locate D first by looking up the location information of D in its LT_l and LT_h. Depending on the result of this table lookup, there could be three cases as follows:

- *Case 1*: A valid entry for D exists in LT_l. S uses Algorithm 5 for data packet delivery.
- *Case 2*: A valid entry for D exists in LT_h. S uses Algorithm 5 for data packet delivery.

S.Forward_Packet(home_loc_query, NB(S));
foreach *intermediate node i* **do**
  if i is a location server of D **then**
    i.Send_Reply(S, home_loc_reply);
    /* *Break out of the loop* */
    Break;
  end
  else
    i.Forward_Packet(home_loc_query, NB(i));
  end
end
S.Update_Table(S.$LT_h(D)$*, home_loc_reply)*;

Algorithm 4: Processing a home region location query.

- *Case 3*: No valid entry for D can be found because the entry has expired or no entry for D exists. In this case, S initiates a location query before sending any data packets to D. Upon receiving the location reply, S updates its location tables and uses Algorithm 5 for data packet delivery.

The procedure for processing data packet delivery given that a valid entry for D exists in either LT_l or LT_h is presented in Algorithm 5. A potential optimization is that in addition to the data payload, m also carries the up-to-date location information of the sender S such that intermediate mobile nodes can update their location tables using such location information. Data packets are forwarded towards the destination using geographical routing.

3.4. Maintenance of Home Region

Because location servers within the home region of a mobile node are also mobiles, they may leave the home region and therefore no longer serve as location servers. Other nodes may also move into the home region and become new location servers for the node. A location server B for a node A can detect whether it is within A's home region by periodically checking its distance to the center of A's home region against the radius R_h of A's home region (B knows R_h from home region location updates sent by A). If B's distance to the center of A's home region is larger than R_h, B is no longer within A's home region. After B leaves A's home region, it will not receive

if $S.LT_l(D) \neq null$ **then**
  $S.Forward\_Packet(m,\ NB(S))$;
  foreach *intermediate node i* **do**
    if $i.LT_l(D) \neq null$ *and* $i.LT_l(D).ts \geq m.Header\_Loc\_Info(D).ts$ **then**
      $i.Update\_Header(m,\ i.LT_l(D))$;
    end
    $i.Forward\_Packet(m,\ NB(i))$;
  end
end
else if $S.LT_h(D) \neq null$ **then**
  $S.Forward\_Packet(m,\ NB(S))$;
  foreach *intermediate node i* **do**
    if $i.LT_h(D) \neq null$ *and* $i.LT_h(D).ts \geq m.Header\_Loc\_Info(D).ts$ **then**
      $i.Update\_Header(m,\ i.LT_h(D))$;
    end
    $i.Forward\_Packet(m,\ NB(i))$;
  end
end

Algorithm 5: Processing data packet delivery.

home region location updates from A, and it will ignore any home region location queries for A's location information even though it may still receive it (because the querying node may not have the up-to-date information on the radius of A's home region).

When a node C moves and enters into the home region of A, C is notified of the current location information of A by existing location server nodes in the home region. DrMoM requires that each node serving as a location server in the home region of A periodically broadcast to its neighbors a message announcing its identity as a location server for A. The message carries the node ID of A, the coordinates of the center and R_h of A's home region, and the current location information of A. When C receives the message from one of A's location servers, it checks if it is within A's home region by comparing its distance to the center of A's home region against the R_h. If C detects that it is within A's home region, it stores the location information of A in its LT_h and starts serving as a location server for A.

4. Performance Model

In this section, we present a performance model for calculating the parameterized overall communication cost per user incurred by DrMoM as a function of R_l and R_h. We define the total communication cost incurred per user for location management and data packet delivery by the *total number of wireless transmissions per time unit per user*. It is worth emphasizing that because the total communication cost is a per time unit metric, a small amount of communication cost savings can be significant over time. Note that we use the total communication cost as the performance metric here because the focus of this paper is on integrated mobility and service management for minimizing the total communication cost. Minimizing the total communication cost will have a significant positive impact on other performance metrics, such as end-to-end packet delay and packet throughput (see Section 4.6). It also has the benefit of maximizing the life time of a MANET since minimizing the total number of wireless transmissions per time unit means minimized battery consumption. Table 2 lists the notations used for model parameters.

According to [9], as a mobile node moves with speed v, the rate σ at which it crosses local regional boundaries can be calculated as:

$$\sigma = \frac{v\pi}{4R_l} \tag{1}$$

Table 2: Notations used in performance analysis.

Notation	Meaning
n	total number of mobile nodes in the MANET
r	wireless transmission range
R_l	radius of a local region
R_h	radius of a home region
$b(R)$	broadcast cost in a region with radius R
v	moving speed (m/s) of a mobile node
σ	crossing rate of local region boundaries of a mobile node
$\bar{d}$	average distance between a node and its home region
α	average number of hops between a node and its home region
γ	node density (average number of nodes per unit area)
λ_l	rate of local region location updates
λ_h	rate of home region location updates
μ	rate of home region maintenance
ϕ	data packet rate
ζ	session rate

Because a home region location update is triggered every time a local region boundary crossing occurs, the rate of home region location updates λ_h is equal to σ. Local region location updates are triggered whenever the distance between the current location and the location where the last update happened exceeds the threshold τ, which is equal to the wireless transmission range. Thus, the rate of local region location updates λ_l of a mobile node depends on the wireless transmission range r and the moving speed v of the node, computed as follows:

$$\lambda_l = \frac{v}{r} \tag{2}$$

The broadcast cost $b(R)$ in a region with radius R is defined as the number of wireless transmissions to cover the entire region, and can be approximated as follows [9]:

$$b(R) = 1 + \frac{\pi R^2}{\pi r^2} = 1 + \frac{R^2}{r^2} \tag{3}$$

Assume that the geographic area of the MANET is an $m \times m$ square. The average distance $\bar{d}$ between any mobile node and its home region in the

$m \times m$ square area can be estimated as [22]:

$$\bar{d} = \frac{2m}{3} \tag{4}$$

Therefore, the average number of hops α between any mobile node and its home region in the $m \times m$ square area can be approximated as follows:

$$\alpha = \frac{\bar{d}}{r} \tag{5}$$

4.1. Location Update Cost C_u

The location update cost C_u consists of two parts: C_u^l, the cost for local region location updates, and C_u^h, the cost for home region location updates. A local region location update from a mobile node S requires broadcasting the location update message among the neighbors in S's local region, thus incurring a broadcast cost of $b(R_l)$. A home region location update requires sending the location update message to S's home region that incurs a cost of α, followed by a broadcast of the message within S's home region that adds a broadcast cost of $b(R_h)$. Therefore, C_u^l and C_u^h are calculated respectively as follows:

$$\begin{aligned} C_u^l &= b(R_l) \\ C_u^h &= \alpha + b(R_h) \end{aligned} \tag{6}$$

4.2. Location Query Cost C_q

The location query cost C_q consists of the cost for a local region location query and optionally the cost for a home region location query which happens only when the local region location query fails. Let C_q^l and C_q^h denote the cost for a local region location query and the cost for a home region location query, respectively. Let p_q^h denote the probability that the home region location query is needed to locate the target mobile node D, i.e., p_q^h is the probability that the local region location query fails. C_q is calculated as follows:

$$C_q = C_q^l + p_q^h \cdot C_q^h \tag{7}$$

A local region location query requires broadcasting the location query message among the neighbors in the local region of the source mobile node S, and collecting replies from these neighbors. Therefore, the cost for a local region location query consists of the broadcast cost $b(R_l)$ in the source mobile's local region and the cost for the neighbors who keep valid location information of D to send the relies back to S. The number of neighbors in S's local region who keep the location information of D can be estimated based on the node density. Specifically, a neighbor in S's local region keeps the updated location information of D when it is also within D's local region or home region, the probability of which is $\frac{\pi R_l^2+\pi R_h^2}{m^2}$, assuming that the n mobile nodes are uniformly distributed in the network. Therefore, the number of neighbors who keep the location information of D can be estimated as follows:

$$\frac{\pi R_l^2 + \pi R_h^2}{m^2} \cdot \pi R_l^2 \cdot \gamma \tag{8}$$

Given the estimated number of neighbors in S's local region who have the location information of D, C_q^l can thus be estimated as:

$$C_q^l = b(R_l) + \frac{\pi R_l^2 + \pi R_h^2}{m^2} \cdot \pi R_l^2 \cdot \gamma \tag{9}$$

A home region location query requires sending the location query message to D's home region, followed by forwarding the location reply back to S. Therefore, the cost for the home region location query C_q^h consists of the costs for sending the location query message and location reply message, calculated as follows:

$$C_q^h = 2\alpha \tag{10}$$

S needs to initiate a home region location query only if the local region location query fails when none of the mobile nodes in S's local region could find a valid entry for D in their LT_l and LT_h. A mobile node in S's local region could not find a valid entry for D if it's not in D's local region and home region, the probability of which is $1 - \frac{\pi R_l^2}{m^2} - \frac{\pi R_h^2}{m^2}$. p_q^h is the probability that all nodes in S's local region are not in D's local region or home region, which is computed as follows :

$$p_q^h = (1 - \frac{\pi R_l^2}{m^2} - \frac{\pi R_h^2}{m^2})^{\pi R_l^2 \cdot \gamma} \tag{11}$$

4.3. Data Packet Delivery Cost C_d

As discussed in Section 3.3, depending on whether there is a valid entry for the target mobile node in the source mobile node's location tables LT_l or LT_h, there could be three cases to consider. In the first two cases (i.e., a valid entry is found in LT_l or LT_h), data packet delivery follows Algorithm 5.

In the third case, however, a location query needs to be performed to first locate the target mobile node before data packets can be delivered using Algorithm 5.

Let C_d^1 and C_d^2 denote the cost for data packet delivery for the first two cases. Also let p_1 and p_2 denote the probability that a valid entry is found in LT_l and the probability that a valid entry is found in LT_h, respectively, then C_d is calculated as:

$$C_d = p_1 \cdot C_d^1 + p_2 \cdot C_d^2 + (1 - p_1 - p_2) \cdot C_q \tag{12}$$

Data delivery in the first case only involves mobile nodes in S's local region that make progress moving data packets towards D, and the distance from S to D is bound by the diameter of the region $2R_l$. Therefore, we can estimate an upper bound of C_d^1 as follows:

$$C_d^1 = \frac{2R_l}{r} \tag{13}$$

Data delivery in the second case consists of two stages: the first stage routes the data packet from S to the first mobile node (say X) on the route that is within D's local region, and the second stage is equivalent to data delivery in the first case, except that the source mobile node is X. Therefore, we can estimate C_d^2 as follows:

$$C_d^2 = \alpha + C_d^1 \tag{14}$$

The source mobile node S can find a valid entry in either LT_l or LT_h only if S is within the local region or home region of D. The probability p_1 (p_2) that S is within the local region (home region) of D can be calculated as follows, assuming that the n mobile nodes are randomly distributed in the MANET:

$$\begin{aligned} p_1 &= \tfrac{\pi R_l^2}{m^2} \\ p_2 &= \tfrac{\pi R_h^2}{m^2} \end{aligned} \tag{15}$$

4.4. Home Region Maintenance Cost C_m

As discussed above, DrMoM handles the case that a mobile node B enters into the home region of another node A and becomes a location server for A by requiring each node in A's home region to periodically broadcast an announcement message to its neighbors within its wireless transmission range. This incurs a home region maintenance cost C_m, consisting of the cost incurred for one wireless transmission by each node in the home region. Therefore, the calculation of C_m is shown as follows:

$$C_m = \pi R_h^2 \cdot \gamma \tag{16}$$

4.5. Total Communication Cost C

The total communication cost C consists of the data packet delivery cost (C_d), the location update cost (C_u), the location query cost (C_q, which is contained in the data delivery cost), and the home region maintenance cost (C_m), multiplied by their rates respectively. C is calculated as follows:

$$C = \phi \cdot C_d + \lambda_l \cdot C_u^l + \lambda_h \cdot C_u^h + \mu \cdot C_m \tag{17}$$

4.6. Discussion

The performance metric discussed thus far is based on C (specified by Equation 17) which is the number of hops of wireless transmissions incurred per mobile node per time unit. Below we relate C minimization with end-to-end delay minimization and throughput maximization. Since every hop incurs a packet transmission by a mobile node serving as a router, C essentially is the amount of traffic incurred to the network per mobile node. Let C_i denote the traffic generated by mobile node i. Then, the total traffic incurred to the network by all mobile nodes is given by $\sum_i C_i$. Consequently, the average input traffic toward each hop (i.e., toward a mobile node serving as a router) is the total traffic divided by the number of mobile nodes in the network. By utilizing simple arguments of collision theory [23] and queueing theory [24], it can be proven that the per-hop packet delay (including the queueing delay and the retransmission delay because of collision) at any router is minimized when the input traffic to the router is minimized. Consequently, the end-to-end delay of a packet is also minimized. By Little's Law [24] which states that throughput multiplied with response time (end-to-end delay) is equal to the packet population in transit, we can deduce that the network throughput is maximized when the end-to-end delay is minimized, which happens when node i operates at the optimal R_h and R_l settings as identified in our analysis to minimize C_i.

5. Performance Evaluation

We consider a scenario that n mobile nodes are randomly distributed in an area of dimensions $2000m$ by $2000m$, with n varying from 100 to 800 with an increment of 100, so that the density of nodes is a function of n. The wireless transmission range is $r = 200m$. We model the data stream between a source and a destination using a constant-bit-rate (CBR) stream at a rate of $\phi = 50$ $packets/s$. The speed of a mobile node (v) varies between $1m/s$ to $20m/s$.

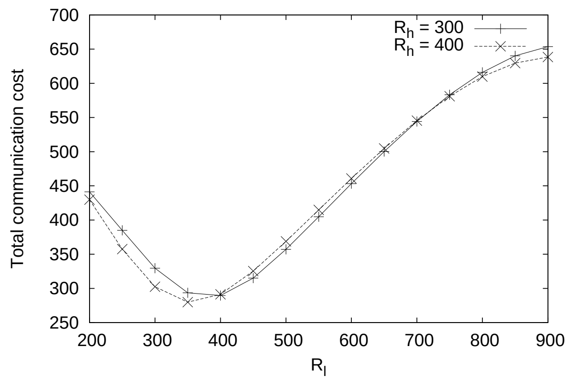

Figure 3: Total communication cost vs. R_l in DrMoM.

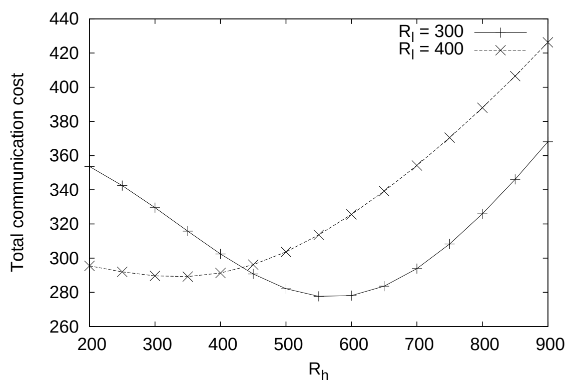

Figure 4: Total communication cost vs. R_h in DrMoM.

5.1. Performance Characteristics of DrMoM

We first evaluate the effect of R_l (R_h) on the performance of DrMoM by varying the value of R_l (R_h) but keeping R_h (R_l) fixed. Fig. 3 and Fig. 4 show the total communication cost as a function of R_l and R_h, respectively, for a scenario where $n = 100$ and $v = 2m/s$. As can be seen in the figures, both R_l and R_h are key parameters and have a significant effect of the total communication cost incurred by DrMoM. More importantly, there exists optimal R_l (R_h) that minimizes the total communication cost incurred by DrMoM. Increasing R_l of a mobile node (and thus the area of the local region) increases the chance that the node is located utilizing only local location information, but it also increases the location update cost as well as the data delivery cost because a data packet tends to travel a longer distance in the local region after it reaches the first node within the local region. The same reasoning applies to R_h.

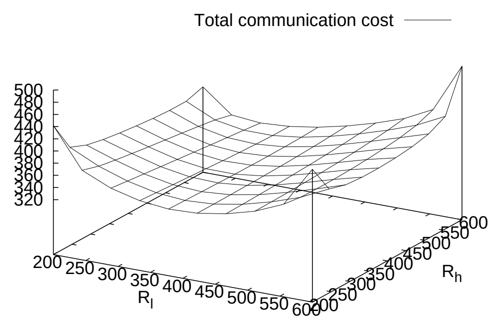

Figure 5: Total communication cost vs. R_l and R_h in DrMoM.

Fig. 5 further shows the total communication cost incurred by DrMoM as a function of both R_l and R_h. The figure depicts the effect of the interaction between R_l and R_h on the total communication cost incurred by DrMoM, and it justifies that there exists an optimal combination of R_l and R_h that minimizes the total communication cost incurred by DrMoM. It can also be seen in the figure that the total communication cost increases sharply when R_l and/or R_h are too large or too small.

5.2. Performance Comparison

In this section, we compare DrMoM with a well known location-based routing protocol called SLURP [9] based on static home regions as well as a region-based location management scheme called RUDLS [16] which claims to outperform contemporary region-based location management schemes. in terms of the overall network cost incurred. To make a fair comparison of

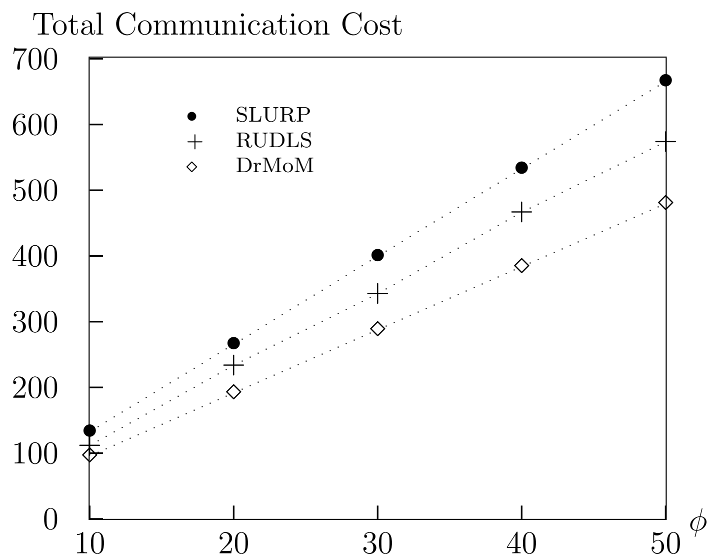

Figure 6: Predicted total communication cost vs. ϕ for DrMoM against SLURP and RUDLS.

DrMoM against SLURP and RUDLS, we use the same parameter values as reported in [9] and evaluate their performance under identical settings.

Fig. 6 compares the total communication cost incurred per time unit by DrMoM vs. SLURP and RUDLS as a function of the packet arrival rate ϕ in the range of 10 to 50 $packets/s$ for the scenario in which $n = 100$ and $v = 2m/s$. It shows that the overall communication cost per time unit per user increases linearly with the packet arrival rate. Fig. 7 compares the total communication cost incurred per time unit by DrMoM vs. SLURP and RUDLS as a function of the moving speed v in the range of 5 to 30 m/s for the scenario in which $n = 100$ and $\phi = 10$ $packets/s$. The communication cost is relatively insensitive to the moving speed v because the data packet delivery cost C_d dominates the location update cost C_u in the scenario considered. As can be seen in these two figures, DrMoM under the optimal setting (optimal R_l and R_h that together minimize the total communication cost) outperforms both SLURP and RUDLS over a wide range of moving speed and packet rate. This result clearly demonstrates the benefit of dynamically determining the optimal R_l and R_h for network cost minimization in DrMoM.

Fig. 8 compares the total communication cost incurred per time unit by DrMoM vs. SLURP and RUDLS as a function of the total number of mobile nodes n, or equivalently the node density, for the scenario in which $v = 2$ m/s and $\phi = 20$ $packets/s$. As the figure illustrates, the total communication

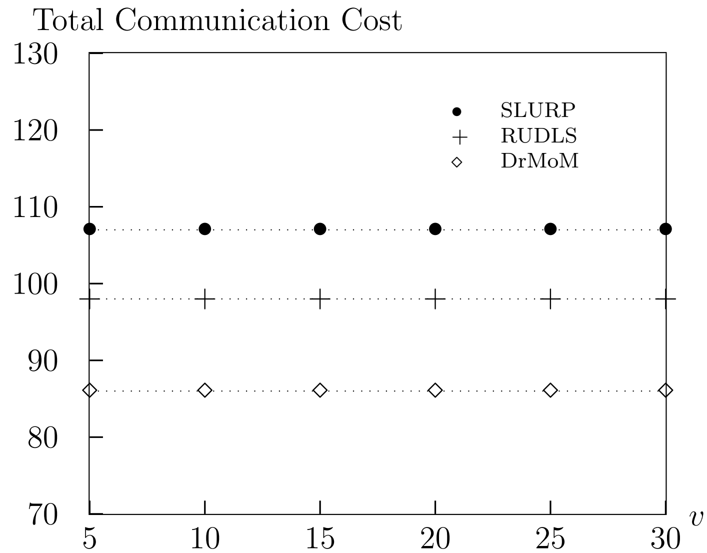

Figure 7: Predicted total communication cost vs. v for DrMoM against SLURP and RUDLS.

cost per time unit per user decreases as the node density increases because the success probability of local location queries increases as the number of neighbors increases. We again see that DrMoM is superior in terms of the total communication cost incurred per time unit per user. The advantage of DrMoM is particularly significant when the node density is relatively small. Again, the figure shows that the node density is a key parameter that affects the total communication cost incurred by a location management scheme for MANETs such as DrMoM, SLURP or RUDLS.

6. Simulation

In this section, we perform simulation validation of the analytical results obtained in Section 5. We also test the sensitivity of the results with respect to node distribution and mobility patterns.

We use a discrete event driven simulation package called smpl [25] for conducting simulation. To ensure statistical significance of simulation results, we use the *batch mean analysis* technique [25] by which the simulation period is divided into batch runs with each batch consisting of 2000 observations for computing the average value. A minimum of 10 batches were run to compute the grand mean for the *overall network cost* metric. Additional batches are added if necessary until the grand mean is within 95% confidence

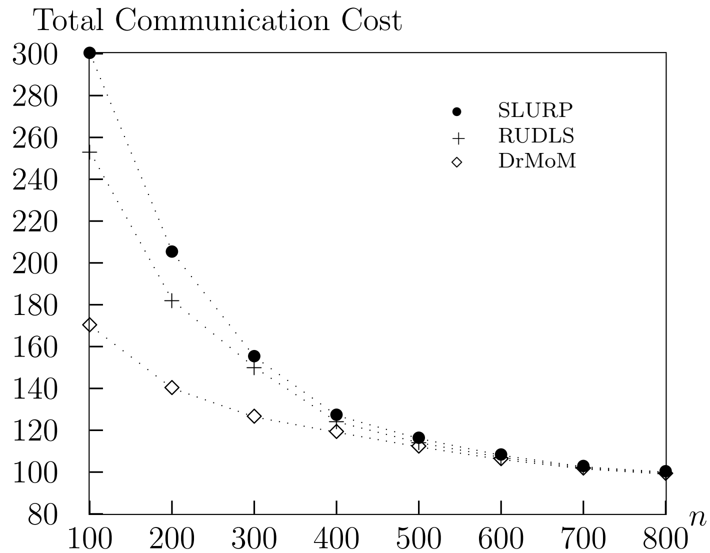

Figure 8: Predicted total communication cost vs. n for DrMoM against SLURP and RUDLS.

level and 10% accuracy from the true mean. With the batch mean analysis technique, a simulation run for collecting a data point will not end until the expected confidence level and accuracy are achieved. To achieve the confidence level and accuracy of 0.95 and 0.05, it normally takes more than 20,000 observations.

Fig. 9 shows the simulation results for the total communication cost incurred per time unit by DrMoM vs. SLURP and RUDLS, corresponding to the analytical results in Fig. 6. We use solid line for simulation results and dashed line for analytical results. We observe that simulation results obtained match well with analytical results with the same trend exhibited. We conclude that the analytical results in Fig. 6 are valid.

The results obtained so far are based on random node distribution and random mobility. Next we test the sensitivity of the result trend with respect to node distribution and mobility. We consider a city-style node distribution pattern such that inner regions have a more dense population than outer regions in the 10x10 regions, with each node following the SWIM mobility model [26] with a distinct mobility pattern. In SWIM [26], a node moves between its home location (not necessarily its home region) and a number of popular places. A node makes a move to one of the population places. The probability of a location being selected is higher if it has a higher popularity probability. When reaching the destination, the node pauses at the destina-

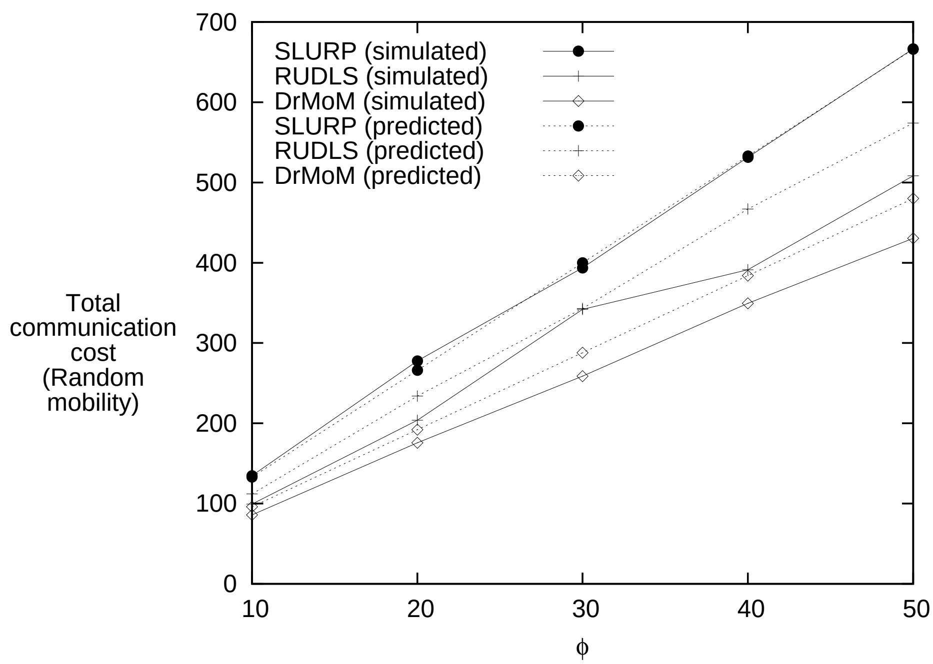

Figure 9: Total communication cost vs. ϕ given random mobility.

tion location for a period of time following a bounded power law distribution. Every node has a distinct set of speed, location popularity, and pause time parameter values. In the simulation, the speed v follows U(5, 30)m/s. The number of popular locations of a mobile node other than the home location follows U(5, 10). Once the number of popular locations is randomly generated, the exact locations (say locations i's) are randomly picked. Each location's popularity probability is calculated based on the mobile node's mobility behaviors: home and away. For the away mobility behavior, the popularity probability is calculated by $\frac{x_i}{\sum_i x_i}$ where x_i is the distance (at least 1) between location i and the mobile node's home location. This models the case in which the mobile user likes to travel to places away from home. For the home mobility behavior, the popularity probability is exactly the reverse. This models the case in which the mobile user likes to go to places close to home. The upper bound pause time is assumed to be 4 hrs as defined in [26].

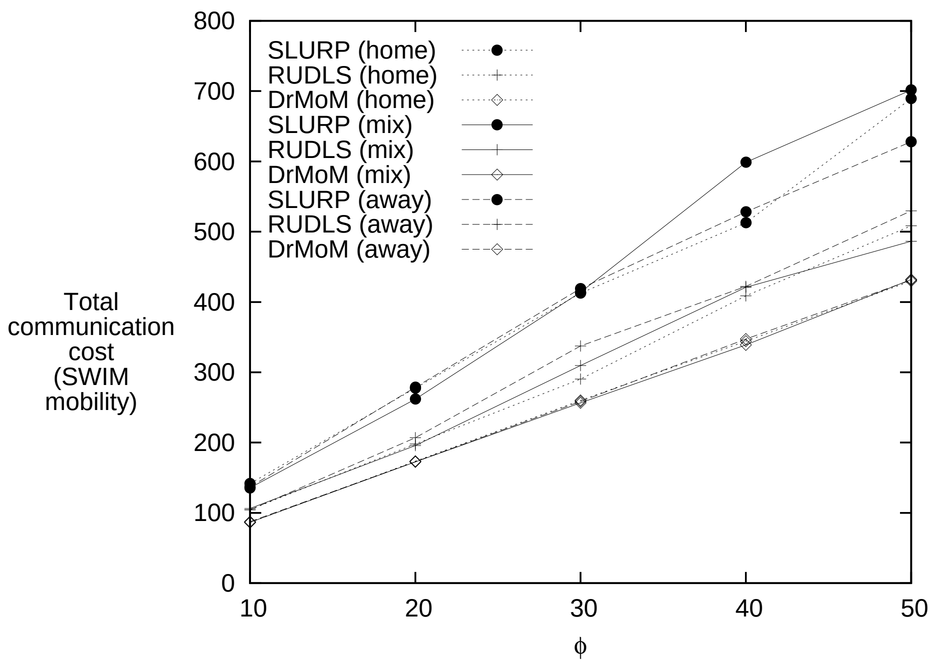

Figure 10: Total communication cost vs. ϕ given SWIM mobility.

Fig. 10 shows the simulation results for the total communication cost incurred by all n users per time unit by DrMoM vs. SLURP and RUDLS, when the nodes observe the SWIM mobility model with home, mixed and away parameterizations. We observe that for each parameterization the simulation results exhibit a similar trend as that in Fig. 6 based on random node distribution and mobility. We conclude that DrMoM outperforms both SLURP and RUDLS and the result is relatively insensitive to node distribution or node mobility.

7. Applicability

The identification of optimal R_h and R_l settings to minimize the overall communication cost per user is performed at static time. One way to apply the results is to build a lookup table at static time listing the optimal R_h and R_l settings discovered over a perceivable range of parameter values characterizing a user's mobility and service behaviors. Then, at runtime, upon sensing its own mobility and service behavior changes matching with a set of parameter values (like mobility rate v, data packet rate ϕ and session rate ζ), a mobile node can perform a simple table lookup operation augmented with extrapolation/interpolation techniques to determine and apply the optimal R_h and R_l settings to minimize the overall communication cost due to location management and packet delivery. We note that each node can autonomously perform dynamic adjustments of its R_h and R_l settings in a totally distributed manner and the computational complexity of such a table lookup operation is O(1).

8. Conclusion

In this paper, we designed and analyzed dual-region based location management (DrMoM) to provide efficient location service in MANETs. The novelty lies in dynamically identifying and applying the optimal home region size and local region size (defined by their respective radii denoted by R_h and R_l) for each mobile node based on the mobile node's runtime mobility and service characteristics to minimize the overall network cost incurred for location management and data packet delivery. We developed a performance model to derive optimal R_h and R_l values as well as the total communication cost incurred by DrMoM. By means of a comparative performance study, we demonstrated that DrMoM outperforms existing location management schemes including SLURP and RUDLS.

This work assumes that there are no malicious or selfish nodes in MANETs to disrupt mobility management. We plan to investigate how trust management protocols such as [27, 28, 29] can be used to select trustworthy nodes to serve as location servers utilizing more elaborated modeling techniques such as stochastic Petri nets [30, 31, 32, 33, 34, 35] to further enhance performance of dual-region based location management in MANETs.

References

[1] Y. Li, I. R. Chen, Design and performance analysis of mobility management schemes based on pointer forwarding for wireless mesh networks, IEEE Transactions on Mobile Computing 10 (3) (2011) 349–361.

[2] B. Gu, I. R. Chen, Performance analysis of location-aware mobile service proxies for reducing network cost in personal communication systems, ACM Mobile Networks and Applications 10 (4) (2005) 453–463.

[3] I. R. Chen, T. M. Chen, C. Lee, Performance evaluation of forwarding strategies for location management in mobile networks, The Computer Journal 41 (4) (1998) 243–253.

[4] I. R. Chen, T. Chen, C. Lee, Agent-based forwarding strategies for reducing location management cost in mobile networks, ACM Mobile Networks and Applications 6 (2) (2001) 105–115.

[5] I.-R. Chen, N. Verma, Simulation study of a class of autonomous host-centric mobility prediction algorithms for wireless cellular and ad hoc networks, 36th Annual Symposium on Simulation, 2003, pp. 65–72.

[6] I. R. Chen, W. He, B. Gu, DMAP: Efficient integrated mobility and service management for Mobile IPv6 systems, Wireless Personal Communications 43 (2) (2007) 711–723.

[7] H. Saleet, O. Basir, R. Langar, R. Boutaba, Region-based location-service-management protocol for vanets, IEEE Transactions on Vehicular Technology 59 (2) (2010) 917–931.

[8] S. Das, H. Pucha, Y. Hu, Performance comparison of scalable location services for geographic ad hoc routing, IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies., Miami, FL, USA, 2005, pp. 1228–1239.

[9] S. M. Woo, S. Singh, Scalable routing protocol for ad hoc networks, Wireless Networks 7 (2001) 513–529.

[10] J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, R. Morris, A scalable location service for geographic ad hoc routing, 6th ACM annual international conference on Mobile computing and networking, 2000, pp. 120–130.

[11] X. Wu, VPDS: Virtual home region based distributed position service in mobile ad hoc networks, 25th IEEE International Conference on Distributed Computing Systems, 2005, pp. 113–122.

[12] W. Wang, C. V. Ravishankar, Hash-Based virtual hierarchies for scalable location service in mobile ad-hoc networks, Mobile Networks and Applications 14 (2009) 625–637.

[13] E. Amar, S. Boumerdassi, Enhancing location services with prediction, IEEE Wireless Communications and Networking Conference, March 2011, pp. 593–598.

[14] C. Hsu, S. Wu, An efficient cost based location service protocol for vehicular ad hoc networks, IEEE international Conference on Communication, Networks and Satellite, July 2012, pp. 93–97.

[15] J. Zhou, J. Lu, Y. Deng, A grid-based predictive location service in mobile ad hoc networks, Future Wireless Networks and Information Systems LNEE 143 (2012) 7–16.

[16] J. Zhou, J. Lu, S. Huang, D. Luo, An efficient region-based location service in mobile ad hoc networks, Journal of Information and Computational Science 9 (15) (2012) 4531–4539.

[17] T. Park, K. G. Shin, Optimal tradeoffs for location-based routing in large-scale ad hoc networks, IEEE/ACM Transactions on Networking 13 (2005) 398–410.

[18] Z. Ye, A. A. Abouzeid, Optimal stochastic location updates in mobile ad hoc networks, IEEE Transactions on Mobile Computing 10 (5) (2011) 638–652.

[19] Y. Li, I. R. Chen, Mobility management in wireless mesh networks utilizing location routing and pointer forwarding, IEEE Transactions on Network and Service Management 9 (3) (2012) 226–239.

[20] M. Ayaida, H. Fouchal, L. Afilal, Y. Ghamri-Doudane, A comparison of reactive, grid and hierarchical location-based services for vanets, IEEE Vehicular Technology Conference, Sept. 2012, pp. 1–5.

[21] S. Basagni, I. Chlamtac, V. R. Syrotiuk, B. A. Woodward, A distance routing effect algorithm for mobility (DREAM), ACM/IEEE international Conference on Mobile Computing and Networking, 1998, pp. 76–84.

[22] L. E. Miller, Connectivity properties of mesh and ring/mesh networks, Wireless Communication Technologies, 2001.

[23] I.-R. Chen, Y. Wang, Reliability analysis of wireless sensor networks with distributed code attestation, IEEE Communications Letters 16 (2012) 1640–1643.

[24] K. Trivedi, Probability and Statistics with Reliability, Queueing, and Computer Science Applications, Wiley, 2002.

[25] M. H. MacDougall, Simulating Computer Systems: Techniques and Tools, MIT Press, 1987.

[26] S. Kosta, A. Mei, J. Stefa, Small world in motion (SWIM): Modeling communities in ad-hoc mobile networking, 7th IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks, June 2010, pp. 1–9.

[27] F. Bao, I.-R. Chen, M. Chang, J.-H. Cho, Trust-based intrusion detection in wireless sensor networks, IEEE International Conference on Communications, 2011, pp. 1–6.

[28] J.-H. Cho, A. Swami, I.-R. Chen, Modeling and analysis of trust management for cognitive mission-driven group communication systems in mobile ad hoc networks, International Conference on Computational Science and Engineering, 2009, pp. 641–650.

[29] J.-H. Cho, A. Swami, I.-R. Chen, Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks, Journal of Network and Computer Applications 35 (3) (2012) 1001–1012.

[30] G. Ciardo, J. Muppala, K. Trivedi, SPNP: stochastic petri net package, 3rd International Workshop on Petri Nets and Performance Models, 1989, pp. 142–151.

[31] I.-R. Chen, D.-C. Wang, Analyzing Dynamic Voting using Petri Nets, 15th IEEE Symposium on Reliable Distributed Systems, Niagara Falls, Canada, 1996, pp. 44–53.

[32] I.-R. Chen, D.-C. Wang, Analysis of replicated data with repair dependency, The Computer Journal 39 (9) (1996) 767–779.

[33] J.-H. Cho, I.-R. Chen, P.-G. Feng, Effect of intrusion detection on reliability of mission-oriented mobile group systems in mobile ad hoc networks, IEEE Transactions on Reliability 59 (1) (2010) 231–241.

[34] I.-R. Chen, F. Bao, M. Chang, J.-H. Cho, Dynamic Trust Management for Delay Tolerant Networks and Its Application to Secure Routing, IEEE Transactions on Parallel and Distributed Systems 25 (5) (2014) 1200–1210.

[35] R. Mitchell, I.-R. Chen, Effect of Intrusion Detection and Response on Reliability of Cyber Physical Systems, IEEE Transactions on Reliability 62 (2013) 199–210.