GrLS: Group-Based Location Service in Mobile Ad Hoc Networks

Hui Cheng, Jiannong Cao, *Senior Member, IEEE*, Hsiao-Hwa Chen, *Senior Member, IEEE*, and Hongke Zhang

***Abstract*—In this paper, we propose a group-based location service protocol named GrLS for Mobile Ad hoc NETworks (MANETs). The novelty of GrLS is in its exploitation of group mobility to improve the efficiency of the location service. GrLS uses different location management strategies for single nodes and for groups of nodes. A single node is responsible for recruiting its own location servers and performing location update. On the other hand, in a group of nodes, only the group leader recruits the location servers and updates its location to a specific home region called group home region. Since the location update cost normally dominates the location service cost for all practical purposes, the overhead of the location service protocol is significantly reduced. Furthermore, when the nodes join or leave groups, GrLS can provide seamless location service handoff. To the best of our knowledge, GrLS is the first location service protocol in MANETs that has explored group mobility and developed group location management for mobile nodes. Both theoretical analysis and simulation results show that GrLS can achieve a higher success ratio of location query and better load balance with much lower overhead than the existing protocols without considering group mobility.**

***Index Terms*—Group location management, group mobility, location service, Mobile Ad hoc NETworks (MANETs).**

I. INTRODUCTION

With the progress of positioning techniques such as Global Positioning System (GPS) and in-door positioning techniques [1], mobile nodes can easily obtain their own locations. This has motivated a new type of routing method, i.e., geographic routing (also called location-based routing). In geographic routing, nodes locally select next-hop nodes based on their neighborhood information and the destination's location. A variety of geographic routing protocols have been developed, e.g., location-aided routing (LAR) [2], greedy perimeter stateless routing (GPSR) [3], geographical routing algorithm (GRA) [4], and Terminode routing [5]. These protocols have good scalability since they allow stateless routing and, hence, reduce the total routing overhead.

A challenging problem in geographic routing is how to provide location service so that a source node can obtain the location of the destination. A number of location service protocols have been proposed, including Virtual home region-based Distributed Position Service (VDPS) [6], Geographic Hashing Location Service (GHLS) [7], Grid Location Service (GLS) [8], Distributed Location Management (DLM) [9], and Hierarchical Location Service (HLS) [10]. They can be divided into flooding-based and rendezvous-based approaches. In the flooding-based approach, the source floods the location query in the whole network. This approach is simple but costly. Therefore, most of the existing work focuses on the rendezvous-based approach, in which any node can query the location of any other node from that node's location servers, which are called the rendezvous nodes. Rendezvous nodes record the location updates from mobile nodes and answer the location queries.

In the rendezvous-based approach, all the nodes in the network need to keep a publicly known mapping, which maps each node's unique ID to its location servers. Each mobile node recruits at least one other node as its location server and, whenever necessary, sends location updates to the location servers to update its location. Once a source node wants to know the location of the destination, it will send a location query to the location servers of the destination. At least one of the location servers should receive the location query and send the location reply to the source. Hence, a rendezvous-based distributed location service protocol needs to address the following issues:

1) how to recruit location servers;
2) when to send location update;
3) how to determine the location servers of a node given its node ID.

Node mobility is one of the intrinsic characteristics in mobile ad hoc networks (MANETs). By single mobility, a node moves according to its own mobility pattern. In recent years, group mobility [11], where mobile nodes are organized into groups to coordinate their movement, has emerged from the demand of applications, where a team of mobile users work together. In the group, all the group members stay close and move together according to the same mobility pattern. Examples include military and disaster recovery operations, vehicular communications, etc. Since a group of nodes always moves as a whole and has similar location tracks, group mobility can further be exploited to improve the efficiency of location management.

We propose a novel location service protocol named Group-based Location Service (GrLS) for MANETs. To our knowledge, it is the first location service protocol that exploits group mobility. GrLS consists of two major components, i.e.,

---

Manuscript received May 17, 2007; revised October 29, 2007 and January 3, 2008. First published March 3, 2008; current version published November 12, 2008. This work was supported in part by the University Grant Council of Hong Kong under CERG Grant PolyU 5170/03E and in part by the China National 973 Program under Grant 2007CB307100. The review of this paper was coordinated by Prof. Y.-B. Lin.

H. Cheng and J. Cao are with the Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: cshcheng@comp.polyu.edu.hk; csjcao@comp.polyu.edu.hk).

H.-H. Chen is with the Department of Engineering Science, National Cheng Kung University, Tainan 701, Taiwan, R.O.C. (e-mail: hshwchen@ieee.org).

H. Zhang is with the College of Electronics and Information Engineering, Beijing Jiaotong University, Beijing 100044, China (e-mail: hkzhang@bjtu.edu.cn).

Digital Object Identifier 10.1109/TVT.2008.919614

single location management and group location management. In single location management, each node with single mobility sends location updates to the location servers in its home region, which also handle all the location queries for it. For nodes with group mobility, group location management applies, which consists of micro and macro group location management. With micro group location management, each group member is aware of the locations of all the other group members. Thus, intragroup communications can immediately be conducted. With macro group location management, a designated group leader updates its location to the location servers in the group home region and replies all the location queries for the group members. Thus, the overhead of location updates to the location servers can be saved for all the group members, except for the group leader.

The major contributions of this paper are as follows.

1) A novel network partition method is proposed to allocate the home regions for mobile nodes. The partition originates from the network center and spreads outward. Thus, all the home regions symmetrically spread around the network center. On an average basis, a home region can potentially be close to both source and destination nodes.
2) A novel strategy of recruiting location servers is proposed for both single nodes and group leaders. The strategy allows the load that maintains the location service to be evenly spread across all the nodes in the network. Moreover, when a location server moves, only one message is needed for location information handoff.
3) An effective and efficient group location management strategy is proposed. By micro and macro group location management, both communication locality awareness and low protocol overhead can be achieved. To manage the group membership information, ID servers are recruited. Correspondingly, ID update, query, and reply are designed. When nodes change their roles upon joining or leaving a group, a seamless handoff between single location management and group location management is supported.
4) Other desirable features of GrLS include the following: a) an adaptive location update scheme, which can achieve a reasonable tradeoff between location accuracy and protocol overhead; b) an optimal strategy of forwarding location update in the home region without using broadcasting or flooding, which has shown the best spatial and temporal performance; and c) effective methods to handle empty regions.

The rest of this paper is organized as follows. Section II describes related works. Section III presents the design of GrLS. Section IV describes the performance evaluation of GrLS and discusses the results. Finally, Section V concludes this paper.

II. RELATED WORK

Since the flooding-based approach usually degrades the network performance, we focus on the rendezvous-based approach. The rendezvous-based approach can further be divided into quorum-based and hashing-based location service protocols [7]. Quorum-based location service protocols usually contain two quorums, i.e., update quorum and query quorum. These two quorums are designed in a way that they have nonempty intersection so that the location query can be replied to by the location servers lying in the intersection. An example of a quorum-based location service is the column-row quorum-based protocol [12], and more methods on how to generate quorum systems can be found in [13].

In hashing-based location service protocols, a publicly known hash function is always available. The input of the hash function is a node ID, and the output can be either node IDs or geographic locations. The hash function is used to obtain information about the location servers of any given node. There are two kinds of hashing-based protocols, i.e., hierarchical or flat. In hierarchical hashing-based protocols, the network coverage area is partitioned into hierarchical layers of subareas. Each node ID is hashed to the location servers residing in different subareas at different levels. In flat hashing-based protocols, the network coverage area is partitioned into different subareas without hierarchy. Each node ID is hashed to the location servers residing in one or more subareas. The essential difference between quorum-based and hashing-based mechanisms has been theoretically analyzed and experimentally investigated in [7]. They compared three location service protocols, i.e., Column-Row Location Service (quorum based), GLS (hierarchical hashing based), and GHLS (flat hashing based). In the remainder of this section, we describe representative hashing-based location service protocols in detail.

GLS [8] is a well-known HLS protocol. It partitions the network coverage area into a hierarchy of squares, and the smallest square is referred to as an order-1 square. In this hierarchy, an order-n square contains exactly four order-$(n-1)$ squares. A node resides in one square at each hierarchy level. The other three squares at the same hierarchy level are the sibling squares. By the principle of the closest ID distance, a node recruits one location server in each sibling square at each hierarchy level. Hence, for a node, the density of location servers is high in the squares near it and low in the squares far from it. Moreover, the nearby location servers are updated more frequently than distant location servers. When a source node needs to know the location of a destination node, among all the nodes for whom it knows the locations, it will select the one whose ID has the least distance to the destination's ID and forward the location query. This way, the location query of GLS traverses a chain of nodes. Since the nodes are moving, the node chain is unstable. As a result, GLS is very susceptible to node mobility. Moreover, the search for a node with the closest ID within a square is costly.

DLM [9] is also a hierarchical hashing-based location service protocol. It partitions the entire network into a hierarchical grid. A hash function directly maps a node's ID to a set of minimum partitions in the network. The node recruits a location server in each minimum partition. In DLM, the location servers of a node are distributed in regions at different hierarchy levels. Different location servers may carry location information with different accuracy levels. Only a small set of location servers needs to be updated when a node moves. DLM is scalable and robust to node mobility. The disadvantage of DLM is that the average query length is relatively large since only a small set of location servers can directly reply the location queries.

HLS [10] is another hierarchical hashing-based location service protocol. The main idea of HLS is similar to DLM. The network coverage area is partitioned into cells, which are hierarchically grouped into regions of different levels. For a given node A, one responsible cell is selected at each hierarchy level by a hash function. An arbitrary node within or close to the responsible cell becomes A's location server. A location server on level-n needs to be updated only when the node moves to another level-$(n-1)$ region. So the location server on level-n only knows the level-$(n-1)$ region in which the node is residing. If node B wants to determine the location of A, it queries the responsible cells of A on the order of the hierarchy until it receives a reply containing the current location of A. HLS is scalable and well suited for networks where communication partners tend to be close to each other. Since an indirect location scheme is used in HLS to reduce the cost of location update, HLS has the same drawbacks as DLM.

VDPS [6] is a flat hashing-based location service system. In VDPS, each node is associated with a virtual home region (VHR), which is a geographic area. Nodes residing in a node's VHR function as its location servers at a probability. A VHR is further divided into subregions. A location update message arriving at the desired VHR is broadcast into each subregion to search the location servers and update them. The location query message is also sequentially broadcast into the subregions until it is received by a location server. Several approaches for improving the system robustness of VDPS are proposed and evaluated by detailed theoretical analysis. However, the protocol overhead is high due to frequent message broadcast. In addition, the protocol performance is affected by node mobility because there is no handoff of location information when a location server leaves a VHR.

GHLS [7] is another flat hashing-based location service protocol. Different from VDPS, the home region of a node consists of only one node who has the closest distance to the hashed location. A lightweight handoff procedure is introduced in GHLS. When a location server finds that another node is a better match for a subset of locations it stores, the location server hands off these locations to the new node. Another feature of GHLS is that it uses a hash function that generates locations within a scaled location server region near the center of the network. This can help alleviate a potential drawback of flat hashing-based protocols—a location server may be far away from both source and destination nodes. Intuitively, a drawback of GHLS is that using a scaled location server region can create service load imbalance among the nodes in the whole network, i.e., higher load in the scaled region.

Compared with hierarchical hashing-based protocols, flat hashing-based protocols avoid the complexity of maintaining a hierarchy of grids and the consequent maintenance due to nodes moving across grid boundaries [7]. The GrLS proposed in this paper is also a flat hashing-based protocol.

III. GrLS PROTOCOL

Without loss of generality, we assume that all the mobile nodes are aware of their own locations and have the same radio transmission range r. Periodic *HELLO* messages are used to exchange node IDs and location information between neighbor nodes. There are two types of nodes in the network, i.e., single nodes and group nodes. A single node moves according to its own mobility pattern. A group node joins a group and moves according to the group mobility pattern. In a group, one node is the leader, and all the other nodes are ordinary members.

The role of each node can change as time goes. A single node can become a group node by joining a group. On the other hand, a group node can become a single node by leaving a group. A group leader can become a group member or leave the group, requiring a new group leader to be designated and the handover of leadership to be performed. Before we describe the detailed protocol design, we present in Table I the definitions of the main concepts used in GrLS.

A. Geographic Area Partitioning

In GrLS, the coverage area of a MANET is partitioned for allocation of home regions to mobile nodes. A network center-based partition method is proposed to achieve the effect that a home region can potentially be close to both source and destination nodes. The center of the network coverage area is roughly estimated at the time the MANET is initialized.

The partition originates from the network center and spreads outward. As shown in Fig. 1, the area is partitioned into equal circle-shaped regions. As the dotted lines show, each circle contains a hexagon. These hexagons are nonoverlapping but can completely cover the entire network. Each circle has six neighbor circles since a hexagon has six sides. We denote the radius of the circle as R, $R = \sqrt{7}/2r$. Thus, there is a central region at the network center, and other regions are symmetrically spread around the central region. Each region is assigned a unique region ID. We do not require the network coverage area to be regular and symmetrical since the symmetry of network coverage partition has no effect on GrLS. We allow the border regions to be in irregular shape and still assign region IDs to them. Since the area occupied by an irregular region is a part of a circle-shaped region, GrLS can still work well in the irregular regions.

At startup, all nodes know the network center and the partition method. Thus, based on its location, a node can calculate the region in which it is staying. We assume that there exists a publicly known hash function that maps a node's ID to a specific region (called its home region), i.e.,

$$F(\text{Node ID}) \rightarrow \text{Region ID}$$

where F is a many-to-one mapping.

The central region is selected as the group home region, where all the group leaders recruit both location servers and ID servers. All the other regions are selected as home regions by single nodes, which recruit location servers there. All the home regions spread around the network center, which can alleviate the drawback of flat hashing-based protocols, i.e., location servers in a home region can potentially be far away from both source and destination nodes. A circle-shaped home region can further benefit from location management, as shown later in Section III-B.

TABLE I
Main Concept Used in GrLS

Concept	Definition
Group mobility	The phenomenon that all group members stay together and move following a certain group mobility model.
Group leader	The group member which is selected or designated as the local coordinator of the whole group.
Location server	A mobile node which is recruited to store the location information of some other mobile nodes.
ID server	A mobile node which is recruited to store the group membership information.
Home region	A circular-shaped geographic region where some mobile nodes recruit their location servers.
Group home region	The central circle-shaped geographic region where all the group leaders recruit their location servers and ID servers.
Basic location management	The location service provided for all the single nodes.
Group location management	The location service provided for all the group nodes.
Micro group location management	The location service provided within the same group.
Macro group location management	The location service provided beyond a single group.
Location update message	The message sent by mobile nodes to their location servers for updating their current location information.
Location query message	The message sent by the source to the destination's location servers for querying the destination's location information.
Location reply message	The message sent by the destination's location servers to the source for replying the destination's location information.
ID update message	The message sent by group leaders to the ID servers for updating the group membership information.
ID query message	The message sent by the proxy nodes to the ID servers for querying the group leader ID of the destination.
ID reply message	The message sent by the ID servers to the proxy nodes for replying the group leader ID of the destination.

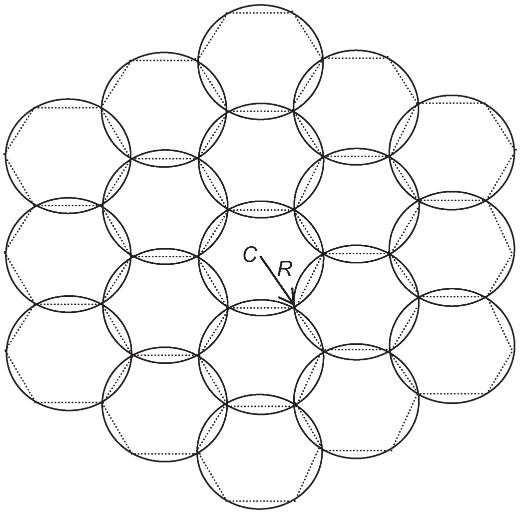


Fig. 1. Network coverage area partition.

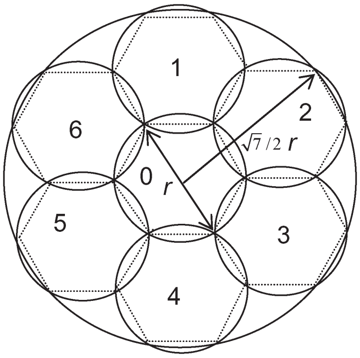


Fig. 2. Division of a home region.

B. Recruiting Location Servers

A mobile node needs to determine which nodes in the home region should be recruited as its location servers. Generally, there are three options: 1) one node; 2) all the nodes; and 3) some of the nodes. The first option has a number of problems, e.g., the centralized server is a single point of failure. The second option produces the heaviest protocol load since all the nodes in the home region are involved in the location service for all the nodes that have been hashed to this home region. The third option seems to be the best, and GrLS also adopts it.

In [6], a node functions as a location server at a probability. However, it will lead to uncertainty and incur high searching overhead. We propose a strategy to evenly distribute the load of location service across all the nodes in the home region. As shown in Fig. 2, we further partition a circle-shaped home region into seven circle-shaped subregions with subregion ID ranging from 0 to 6. A node will recruit one location server in each subregion of its home region. Each subregion is a small circle with a radius of $0.5r$. Thus, a node can directly send messages to all the other nodes in the same subregion. This is the reason why we partition the network into circle-shaped regions with a radius of $\sqrt{7}/2r$.

As previously mentioned, the location servers of node A will evenly be distributed in its home region. To further balance the load among the nodes in the same subregion, node A will recruit the node whose ID is closest to its own ID. We define the ID "close" relationship as follows. The node ID space is assumed to be circular in clockwise direction from small IDs to large IDs. In the space near its "closest" neighbor, a node has the least ID distance, which is measured clockwise from the node's ID to the neighbor's ID. For example, there are 60 nodes with ID ranging from 1 to 60. Now, node 20 wants to recruit a location server in a subregion with nodes 16, 25, and 40. According to the rule, node 25 is recruited. Hence, different nodes recruit different location servers in the same subregion. Overall, the responsibilities of acting as location servers are evenly shared among all the nodes in a subregion.

Through the above analysis, the proposed strategy of recruiting location servers has two desired properties: 1) Each node selects the same number of location servers that are evenly distributed in its home region; and 2) each location server in the home region also serves approximately the same number of nodes. As a result, this strategy is scalable and load balanced.

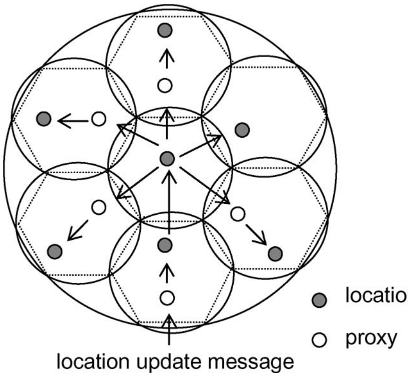


Fig. 3. Center-based forwarding of location update message.

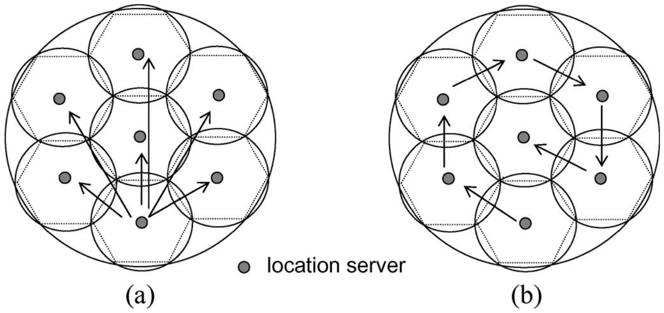


Fig. 4. Another two possible strategies of forwarding location update in a home region. (a) Parallel forwarding. (b) Sequential forwarding.

C. Basic Location Management in GrLS

1) Adaptive Location Update: Generally, similar to the cellular network, there are two kinds of schemes to trigger location update in MANETs, i.e., time based and distance based. In the time-based scheme, a mobile node periodically updates its location, e.g., every tenth of a second. In the distance-based scheme, a mobile node tracks the distance it has moved since the last update and triggers the location update when the distance reaches a predefined threshold.

GrLS adopts an adaptive location update scheme combining the advantages of both time-based and distance-based schemes. Initially, we set the minimum and maximum location update intervals and define the distance threshold of the location update. If the distance the node has moved since last update reaches the threshold value, but the time has not exceeded the minimum location update interval, the node will not send any location update; if the time is between the minimum and maximum interval, a location update will be sent. On the other hand, if the maximum location update interval is reached, but the distance the node has moved is less than the threshold value, the node will immediately trigger the location update. For the distance threshold of location update, according to [14], it can approximately be half of the radio transmission range of mobile nodes.

The adoption of a minimum interval can help reduce the frequency of location update when nodes are moving with high speeds. In highly mobile networks, if no restriction is put on the minimum interval, many location update messages will be generated, probably leading to network congestion. The maximum interval aims to guarantee a certain frequency of location update when nodes are moving slowly or staying stationary. Because many location servers will set an expiry timer for the location information they have stored, if a node has not updated its location for a long time, the location server will remove it from the database. Hence, when a node moves with low speed or stays stationary, it should still periodically send location update to its location servers upon the expiration of the maximum location update interval.

Now, we describe the basic location update mechanism in GrLS. Both single nodes and group leaders send the location update messages toward their home regions using geographic forwarding, where the center of each home region is the destination. Once a node residing in one subregion of the home region receives the message, this node becomes a proxy for this subregion. The proxy knows all its neighbors in the same subregion through *HELLO* message exchange. According to the strategy of recruiting location servers, given the source ID in the location update message, the proxy can easily determine which node is the desired location server in its subregion. The proxy then forwards the message to this location server. It is possible that the proxy itself is the desired location server. Upon receipt of the location update message, the location server updates the corresponding location information, appends its subregion ID to the location update message, and continuously forwards it toward the center of the central subregion, which is also the center of the home region. Since the location update message may traverse some other subregion before it arrives at the central subregion, the same process of message forwarding by the proxy and location update to the desired location server will be performed in each visited subregion. The subregion ID of each visited subregion is also appended to the message. When the location update message finally reaches the central subregion, a proxy forwards it to the desired location server. Then, the location server in the central subregion separately unicasts the location update message to the remaining unvisited subregions, where their centers are the destinations. Finally, the location servers are updated in these unvisited subregions. Thus, the location update of the node to its home region is completed. Fig. 3 shows how a location update message arrives at all the desired location servers once it reaches the home region, which is called center-based forwarding.

In a home region, our strategy of forwarding the location update message incurs a low overhead because neither broadcasting nor flooding is used. Moreover, it has good spatial and temporal performance. To show this, we compare it with two other possible forwarding strategies, which are illustrated in Fig. 4. In Fig. 4(a), the first location server separately unicasts the location update message to the other six subregions. We call it parallel forwarding. In Fig. 4(b), the first location server unicasts the location update message to its neighbor subregion in a clockwise direction. The neighbor subregion continuously forwards the message to its own neighbor subregion, also in a clockwise direction. The location server in each visited subregion is required to append its subregion ID to the message. This forwarding procedure is repeated until one location server finds that the message has already visited its neighbor subregion

TABLE II
COMPARISON RESULTS OF THE THREE STRATEGIES OF FORWARDING LOCATION UPDATE MESSAGE

Strategy	Hops traversed by the location update message	Total time spent on the location update procedure
Parallel forwarding	9	$2t$
Sequential forwarding	6	$6t$
Center-based forwarding	6	$2t$

in a clockwise direction. Then, the location server sends the location update message to the central subregion. Now, the location update procedure is completed. We call it sequential forwarding.

Without loss of generality, we assume a simplified model. In the model, each location server just lies at the center of the subregion. Thus, the forwarding between two neighbor centers is just one-hop transmission, which also brings two-hop reachability between two nonneighbor centers. We count the total number of hops traversed by the location update message in each forwarding strategy. In addition, we assume that the average time for one-hop transmission is t. Then, we get the approximate time spent on the location update procedure for each strategy. The counting begins from the time that the first location server receives the message. In both parallel forwarding and center-based forwarding, the update message needs to travel two hops to reach the farthest location server, whereas it is six hops in sequential forwarding. The comparison is shown in Table II.

Table II shows that the center-based forwarding strategy is the best one, which traverses the minimum number of hops and spends the least amount of time. These two values are also optimal for any possible forwarding strategy, which can easily be proved. Therefore, our forwarding strategy has good spatial and temporal performance.

2) Location Query: If a source node s wants to query the location of a destination d, it will first query its own location database. If d's location can be found, there is no need to trigger a location query. Otherwise, s sends a location query message to d's home region. Since s knows the hash function, d's ID, and the network center, s can easily calculate the location of the center of d's home region, which is just the destination of the location query message. The location query message also carries s's location, which is useful when a location server sends a location reply to s. Since d may be a single node or a group node (group member or leader), different location query strategies are proposed. Here, we describe the location query for single nodes. The strategy for querying group nodes will be described in Section III-D2.

For a single node d, the message for querying its location will first be received by a node in one subregion of d's home region. Then, the node acts as a proxy in this subregion. By d's ID, which is carried in the location query message, the proxy can easily determine which node is the desired location server of d. If the desired location server is just the proxy, a reply can immediately be sent back to s. Otherwise, the proxy directly forwards the location query message to d's location server in this subregion since the location server is one neighbor of the proxy. Upon receiving the location query, the location server sends a location reply message to the source s through geographic forwarding.

D. Group Location Management in GrLS

As pointed out in [15], in realistic MANET environments, random mobility and group mobility simultaneously occur. Several group mobility models have been proposed [11], [16]–[18], where groups exist in the network, and each group of nodes stay close and move as a whole. By far, group mobility has not been addressed in location service protocols. In GrLS, we propose specific group location management for nodes that have formed groups. The group location management consists of two parts: micro group location management, which helps each node acquire the locations of all the other nodes in the same group; and macro group location management, in which only the group leader updates its location to location servers and answers the location query for any node in the group.

Each group can be regarded as a local region. Initially, a group leader is selected. The group leader can be the node that is most stable and stays at the approximate center of the group. Here, "the most stable" means that the group leader has the most approximate velocity to group velocity. A group leader like this can guarantee that each group member has an average minimum distance to the group leader. However, the detailed method of group leader selection is out of the scope of this paper.

1) Micro Group Location Management:

a) Group initialization: Once a group leader is determined, it broadcasts its ID and location information to all the group members. Then, each group member is aware of the group leader. Upon receiving the announcement of the group leader, each group member makes a reply by sending its own ID and location information to the group leader. When the group leader has collected the information of all the group members, it generates a GroupView message containing both ID and location information of all the group members. The GroupView message is then broadcast to all the group members. Here, a location-guided multicast tree [19] from the group leader to all the group members can also be constructed to transmit messages. Once a group member receives the GroupView message, it can maintain a consistent view about the group and know the location of any other group member. Then, the group initialization is completed.

b) Group maintenance: We define a new concept, i.e., group relative location. In addition to the actual location, each group member also has a group relative location, which is the relative location of its actual location to the actual location of the group leader. The group relative location of the group leader itself is (0, 0). Each group member periodically calculates its group relative location. Once the distance change of its group relative location has reached a predefined threshold, the group member will send a location update to the group leader. In addition, when the maximum location update interval is reached, the group member also needs to immediately send a location update to the group leader.

If a group member has not updated its location for a predefined time period (i.e., the location expires), the group leader will think it has left the group and will then remove its ID and location information from the database. When the group leader finds that the number of group members, whose locations have changed or expired, has reached a certain percentage of the group size (i.e., the total number of nodes in the group), it broadcasts a GroupViewChange message to all the group members to refresh the group view. The group leader also broadcasts its own location update to all the group members based, however, on the distance change of its actual location.

Since group initialization and group maintenance are necessary components in the group management protocol, we piggyback the location information into the group management messages to realize micro group location management. Both the group leader selection and the group management protocol are not part of the location service protocol. Hence, the extra overhead caused by micro group location management can be ignored. Furthermore, the communications within the same group are locality aware since each group member directly knows the locations of all the other group members.

2) Macro Group Location Management:

a) Group home region: In Section III-B, we have introduced how to recruit location servers for single nodes. GrLS does not provide home regions for group nodes except group leaders. All the group leaders share the same group home region, i.e., the central region at the network center. Similar to other home regions, the group home region is also divided into seven subregions with subregion ID ranging from 0 to 6. Each group leader recruits one location server, which has the closest ID to its own ID, in each subregion of the group home region.

As we have mentioned, one drawback of flat geographic hashing protocols is that a home region can potentially be far away from both source and destination nodes, causing location update and query with high overhead. To alleviate this problem, we let all the group leaders recruit location servers in the central region. The number of group leaders is exactly the same as the number of groups, which is intuitively small. Thus, the nodes within the central region will not be overloaded. If we want to further reduce the load in the group home region, we can scale it to the central region plus its six neighbor regions.

b) Reactive ID update: In each subregion of the group home region, the node with the least ID is recruited as the ID server by all the group leaders. Totally, there are seven ID servers in the group home region. The ID server is used for group membership management. It stores the group membership information of each group, i.e., the IDs of both the group leader and all the group members.

ID update, a new type of update message, is created to update the group membership information stored in the ID server. An ID update message is generated on demand by the group leader when a new node joins the group or a group member leaves the group. Since most groups are purposely formed by nodes, group membership does not drastically change. Hence, ID update is triggered much less than location update. The overhead incurred by ID update is also much lower than the one incurred by location update.

c) Location service handoff:

1) When a node joins a group, it will notify its home region to disable the location service for it. Then, it sends its ID and location to the group leader.

   The node sends a location update to nullify its location information stored in its location servers, but the node ID is still kept in the location servers to indicate that the node has joined a group. This is different from the case in which all the information of a node is removed from the location servers due to expiry. Once receiving the message from the new group member, the group leader sends an ID update to the ID servers.

2) When a group node leaves its group, if it becomes a single node, it will notify its home region to enable the location service for it; if it joins a new group, it sends its ID and location to the new group leader.

   When a group node leaves its group, its old group leader needs to report the group membership change to the ID servers. If the node becomes a single node, it sends a location update to its original location servers in its original home region. Thus, the location query for it can directly be answered by its location servers. If the node joins a new group, its new group leader also reports the group membership change to the ID servers.

Thus, GrLS can support seamless handoff between single location management and group location management.

d) Query for group nodes: If d is a group member, its original location servers have been disabled. However, the source s does not know this due to distributed location service. So the location query message will still be sent to the original home region of d. When an original location server of d receives this message, it finds that the location information of d has been disabled. It then forwards the message toward the group home region, where the network center is the destination. Once the location query is received by a node in one subregion of the group home region, the node acts as a proxy. Since an ID server exists in each subregion, the proxy sends an ID query message to the ID server requesting the ID of d's group leader. The ID server sends the requested group leader ID back to the proxy by an ID reply message. Then, according to the strategy of recruiting location servers, the proxy can determine which node is the desired location server of d's group leader. If the desired location server is just the proxy, it forwards the location query to d's group leader. Otherwise, it forwards the location query message to the desired location server, which continuously forwards the message to d's group leader. When d's group leader receives the location query message for d, it directly sends a location reply to the source s.

If d is a group leader, the location query procedure is the same as the other group members before the location query message arrives at one of the desired location servers of d. When the location server finds that it has knowledge of d's location, it directly sends a location reply to the source s.

For single nodes or group nodes, neither broadcasting nor flooding is used in our location query procedures. Location query messages from different network areas can be processed by the location servers in different subregions because the group home region lies at the network center. Therefore, on average, each border subregion of the group home region handles 1/6 of the location queries for group nodes. When one border subregion is empty, the location servers in the central subregion will be queried. So the query load can also be evenly distributed over the entire group home region. In addition, both ID query and ID reply only experience one hop transmission and incur trivial overhead.

E. Location and ID Information Handoff in GrLS

Due to node mobility, nodes may move into or out of subregions in a home region. When a location server moves out of the subregion it resides, the location information stored in it needs to be migrated to other nodes in the same subregion. Since the leaving node is aware of all the other nodes in the same subregion, it knows which one has the next closest ID to each source for whom it is acting as the location server. Thus, each source will have a new location server. The leaving node then separately hands off the location information of each source to its new location server. If the leaving node is the ID server, it directly hands off the group membership information to the node with the next least ID in the same subregion. Then, this node becomes the new ID server.

When a new node enters a subregion, all the other nodes in the subregion will know it by beaconing. If one node finds that the new node has a "closer" ID to the sources of some location information it stores, it will handoff this location information to the new node. Thus, the new node will act as the location server for all the sources of the migrated location information. If the ID server in this subregion finds that the new node has a smaller ID, it will hand off all the group membership information to the new node, which becomes the new ID server.

Only one handoff message is needed when an old location server leaves a subregion or a new node enters a subregion and becomes the new location server. This point is proved in Theorem 1 in the Appendix. It benefited from our strategy of recruiting location servers. Through such a simple location and ID information handoff procedure, the location update or query message can still reach the desired location servers, regardless of the change of location servers or ID servers.

F. Handling Empty Regions in GrLS

1) Handling Empty Subregions: A home region consists of seven subregions, i.e., one central subregion and six border subregions. In a home region, one subregion may be empty, i.e., no nodes in it. It is impossible that all the subregions are empty except the central one if we assume a connected ad hoc network. However, the network may sometimes be disconnected due to node movement. If the following three cases occur, the location service protocol will be affected: 1) The destination d is disconnected from all its location servers; 2) the source s is disconnected from all of d's location servers; and 3) s is disconnected from all the location servers connected with d. To handle the temporary network partition, the source s can set a timer for each initiated location query. Once a timer expires and s still has not received the reply, s will increase the timer and resend the location query. After a few retries, if the query still fails, s will discard the query.

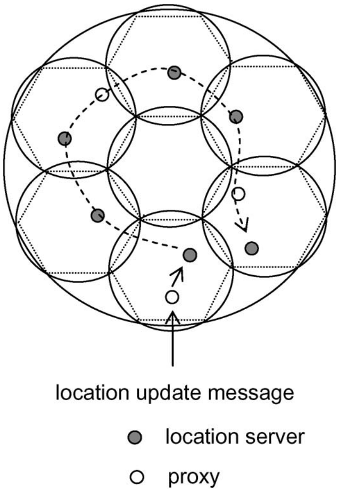


Fig. 5. Forwarding procedure of a location update in a home region with an empty central subregion.

If some border subregions are empty but the central subregion is nonempty, the location update message can still arrive at the desired location server in the central subregion. Upon receipt of the location update message, the location server in the central subregion separately unicasts it to the remaining unvisited subregions. The messages to empty subregions will finally be dropped. If the central subregion is empty, our location update forwarding strategy still works by exploiting the advantage of geographic forwarding. For example, GPSR [3], which is a widely used geographic routing protocol, can route packets around the perimeter of an empty region. If the central subregion is empty, the location update message is routed along a perimeter formed by nodes surrounding the central subregion. In each nonempty border subregion, at least one node will lie on the perimeter. So a location update message can be received by its location servers in all these nonempty subregions. Fig. 5 illustrates an example of the location update procedure in a home region with an empty central subregion. Similarly, an ID update message can also arrive at all the ID servers in nonempty subregions.

For the location query message, since its destination is also the center of the home region, it can similarly arrive at one nonempty subregion by exploiting the advantage of geographic forwarding. As a result, one location server will receive the location query if there exists at least one nonempty border subregion in the home region. The location server then directly sends the location reply to the source or continuously forwards the location query to the group home region. Fig. 6 shows how a location query message arrives at a desired location server in a home region with empty subregions. In the group home region, once a location query message arrives at a nonempty subregion, the proxy node will send an ID query to the ID server in the same subregion.

2) Handling Empty Home Region: Normally, it rarely occurs that the entire home region is empty for dense and

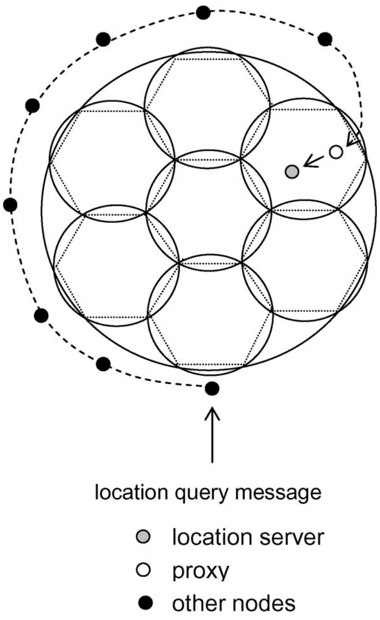


Fig. 6. Forwarding procedure of a location query in a home region with empty subregions.

large-scale MANETs. In the following, we give some suggestions about how to handle an empty home region.

If a node is the last node in a home region, when it wants to leave the region, it needs to hand off location information or group membership information stored in it to one neighbor region of the home region. The nearest-neighbor region to the leaving node will be selected. Once a node in this neighbor region receives the migrated information from a different region, it forwards them to the node with the least ID in its subregion. The node with the least ID will store these migrated location or group membership information.

To make a node know it is the last node in the entire home region, the advantage of GPSR can be exploited again. As mentioned in Section III-C, a location server will append its subregion ID to the location update message before forwarding it. When node x receives a location update or query message destined to the home region in which it resides and it finds that greedy forwarding is impossible, it will set the message to enter the perimeter mode. By GPSR perimeter mode, if x is the only node in the home region, a message destined to the center of the home region will return to x again with only x's subregion ID appended. So if x receives a duplicate location update message with only its subregion ID appended, x will know it is the last node in the home region. After that, when x receives the first location update message with the other subregion ID appended or a forwarded message is not looped back, x will know that some other nodes have entered this home region.

Since the nodes that have selected an empty region as their home region do not necessarily know about its emptiness, they continuously send location update messages to it. Similarly, the location queries to these nodes will still be sent to the empty home region. When a location update or query message finds that the destined home region is empty, it will search the neighbor regions of the empty region. The node with the least ID in each subregion of each neighbor region is checked until the node, which has the migrated location information for the source of the message, is found.

When a new node enters an empty home region and receives a location query message that it cannot answer, the node will append its subregion ID to the message and continue to forward it toward the center of the home region. Similar to the last node in a home region, if the new node receives the duplicate location query message with only its subregion ID appended, it knows there is still no location server to answer the location query. Then, it will forward the location query to the neighbor regions to search the desired location information, as previously mentioned. The location update and query overhead incurred by empty home regions are higher than those incurred by nonempty home regions.

G. Theoretical Analysis

In this section, we conduct a theoretical analysis on the performance of GrLS. The feature of GrLS is that it exploits group location management to reduce protocol overhead. If group location management is not adopted in GrLS, all the nodes then need to send location update messages to their home regions, even if they have formed groups. This kind of scheme still works as a location service protocol, and we denote it as GrLS-. Since GrLS- is a simplified version of GrLS, we first analyze GrLS- and then extend the analysis to GrLS. To make the analysis tractable, referring to [7], we also assume that the network is static, the nodes are uniformly distributed in the geographic area, and location information is not cached at forwarding nodes. Since all the nodes stay stationary, the location update is only triggered by the maximum location update interval.

Before proceeding further, let us introduce the following notations used in the analysis:

N	network size, i.e., number of nodes in the network;
$T_{\mathrm{loc\_upd\_max}}$	maximum location update interval;
$T_{\mathrm{net\_time}}$	network lifetime;
$\overline{L_{\mathrm{upd}}}$	average location update path length in GrLS-;
$\overline{L_{\mathrm{que}}}$	average location query path length in GrLS-;
$\overline{L_{\mathrm{rep}}}$	average location reply path length in GrLS-;
$\overline{g}$	average group size in GrLS;
n	number of groups in GrLS;
n_que	number of location queries.

In GrLS-, a source node sends a location query to the home region of the destination node. Once the query reaches one border subregion of the destination's home region, a location server in this subregion will receive the query message and no longer forward it. By this definition, the average hop number a query message travels is denoted as $\overline{L_{\mathrm{que}}}$. Since a location reply message will be sent back to the source by the location server, the average hop number a query message travels is also $\overline{L_{\mathrm{que}}}$, i.e., $\overline{L_{\mathrm{rep}}} = \overline{L_{\mathrm{que}}}$. For a location update message, on average, it also travels $\overline{L_{\mathrm{que}}}$ hops to reach a location server in a border subregion of its home region. However, since the location servers located in the remaining six subregions also need to be updated, the update message will further be forwarded in the home region, as shown in Fig. 3. In each remaining subregion, the update message will either directly reach the location server or be forwarded to the location server by a proxy node. Hence, on average, the extra transmissions for updating the remaining six location servers are 9 (1.5 × 6) hops. So, we have

$$\overline{L_{\text{upd}}} = \overline{L_{\text{que}}} + 9. \tag{1}$$

In GrLS, only single nodes and group leaders need to send location update messages to their location servers. For single nodes, their average location update path length is also $\overline{L_{\text{upd}}}$. For group leaders, they have the common group home region at the network center. Hence, their average location update path length is less than $\overline{L_{\text{upd}}}$. However, since $n \ll N$, to simplify the analysis, we just assume that their average location update path length is also $\overline{L_{\text{upd}}}$.

We denote the total location update message transmissions in GrLS- and GrLS as U' and U, respectively. Based on the above analysis, we have

$$U' = N * \frac{T_{\text{net\_time}}}{T_{\max\text{\_loc\_upd}}} * \overline{L_{\text{upd}}} \tag{2}$$

$$U = (N - n\overline{g} + n) * \frac{T_{\text{net\_time}}}{T_{\max\text{\_loc\_upd}}} * \overline{L_{\text{upd}}}. \tag{3}$$

In GrLS-, all the queries are for single nodes. In GrLS, the cases are different since the queries may be for the group nodes. For queries to single nodes, the average hop length is also $\overline{L_{\text{que}}}$. The queries to group nodes can be divided into two categories, i.e., one is for group leaders and the other is for group members. Because $O(n\overline{g}) = O(N)$, we have $n \ll n\overline{g}$. Therefore, to simplify the analysis, we assume that all the queries to group nodes are for group members. According to GrLS, a query for a group node will first be forwarded to one of its original location servers with the average path length $\overline{L_{\text{que}}}$. Then, it is forwarded to the group home region with average path length of 3.8 hops [$(12 * 2\sqrt{3}R + 6 * \sqrt{3}R)/18r$, referring to Fig. 1]. Finally, the query message is forwarded to the group leader of this group node with average path length of three hops [$(2\sqrt{3}R + R)/2r$, also referring to Fig. 1]. Hence, in GrLS, a query message for a group node will travel $\overline{L_{\text{que}}} +$ 6.8 hops on average.

We denote the total location query message transmissions in GrLS- and GrLS as Q' and Q, respectively. Based on the above analysis, we have

$$Q' = n\_que * \overline{L_{\text{que}}} \tag{4}$$

$$\begin{aligned} Q &= n\_que * \left(1 - \frac{n\overline{g}}{N}\right) * \overline{L_{\text{que}}} + n\_que * \frac{n\overline{g}}{N} * (\overline{L_{\text{que}}} + 6.8) \\ &= n\_que * \overline{L_{\text{que}}} + n\_que * \frac{n\overline{g}}{N} * 6.8. \end{aligned} \tag{5}$$

In GrLS, the average location reply path length is approximately the same as $\overline{L_{\text{rep}}}$. Hence, for the same n_que queries, the total location reply message transmissions triggered in GrLS are the same as GrLS-. Since we have assumed a static network and static groups, there is no ID update, ID query, ID reply, and location or ID information handoff messages triggered. In mobile networks, these control messages will be triggered. However, the number of these control messages is trivial compared to both location update and query messages, as shown and explained in Section IV-B.

Hence, compared to GrLS-, the reduction in the protocol overhead (i.e., number of control messages) of GrLS is

$$\begin{aligned} &(U' + Q') - (U + Q) \\ &\quad = N * \frac{T_{\text{net\_time}}}{T_{\max\text{\_loc\_upd}}} * \overline{L_{\text{upd}}} + n\_que * \overline{L_{\text{que}}} \\ &\qquad - (N - n\overline{g} + n) * \frac{T_{\text{net\_time}}}{T_{\max\text{\_loc\_upd}}} * \overline{L_{\text{upd}}} \\ &\qquad - \left(n\_que * \overline{L_{\text{que}}} + n\_que * \frac{n\overline{g}}{N} * 6.8\right) \\ &\quad = (n\overline{g} - n) * \frac{T_{\text{net\_time}}}{T_{\max\text{\_loc\_upd}}} * \overline{L_{\text{upd}}} - n\_que * \frac{n\overline{g}}{N} * 6.8. \end{aligned} \tag{6}$$

Since $n \ll n\overline{g}$, we have

$$\begin{aligned} &(U' + Q') - (U + Q) \\ &\quad \approx n\overline{g} \left(\frac{T_{\text{net\_time}}}{T_{\max\text{\_loc\_upd}}} * \overline{L_{\text{upd}}} - n\_que * \frac{6.8}{N}\right). \end{aligned} \tag{7}$$

From (7), since all the parameters except $n\overline{g}$ are the same for both GrLS- and GrLS, we can see $n\overline{g}$, i.e., the number of group nodes plays the most important role in the reduction of control messages. Hence, in theory, with more group nodes in the network, GrLS can reduce more protocol overhead. The following simulations also verify the theoretical declaration.

IV. PERFORMANCE EVALUATION

To study the performance of GrLS, we implement it, as well as geographic forwarding in GloMoSim 2.03 [20]. GloMoSim is a widely used wireless network simulator with a comprehensive radio model. It is designed using the parallel discrete-event simulation capability provided by Parsec. For comparison purposes, we have also implemented GLS. Geographic forwarding adopts GPSR with activated perimeter mode. We use the 802.11 MAC protocol with distributed coordination function (DCF) and a transmission range of 250 m. The network coverage area is a square of 3 km × 3 km, which can be partitioned into 19 full regions, as shown in Fig. 1. In mobility scenarios, single nodes follow the random waypoint mobility model, where each node moves at a constant speed randomly chosen from a predefined speed range. The speed range is different for each simulation scenario. For group mobility, we use the Reference Point Group Mobility (RPGM) model [17], where different group motion vectors are assigned for different groups. As mentioned in Section III-C1, the predefined update threshold is fixed at 125 m, which is half of the transmission range. The minimum update interval is set to be 12.5 s, which is the approximate result of the update threshold divided by the average node speed (125 m/10 ms^{-1}). The simulation duration is 900 s. All these important simulation parameters are listed in Table III.

TABLE III
SIMULATION PARAMETERS

Parameter	Value
Simulation Time	900 sec
Simulation Area	3km x 3km
Transmission Range	250m
Speed Range	1 – 20ms^{-1}
MAC Protocol	IEEE 802.11
Mobility Model	Random Waypoint, RPGM
Update Threshold	125m
Minimum Update Interval	12.5 sec
Maximum Update Interval	40 sec

We assume two types of network models, i.e., quasi-static ad hoc networks and MANETs. In the mobility model followed by quasi-static ad hoc networks, the pause time is set to be 30 s, and the node speed range is [1 ms^{-1}, 5 ms^{-1}]. Since the location update threshold is 125 m, in quasi-static ad hoc networks, the nodes send the location update messages using the maximum update interval if their speeds are less than 3.125 ms^{-1} (125 m/40 s). Quasi-static ad hoc networks simulate networks where nodes stay stationary or move slowly. In the mobility model followed by MANETs, the pause time is 0 s, and the node speed range is set to be [5 ms^{-1}, 20 ms^{-1}].

A. *Load Balance*

In GrLS, by the hash function, each home region is selected by approximately the same number of nodes. Further, each home region is divided into seven subregions. Then, each node, which has selected this home region, recruits a location server in each of its subregions. Therefore, a location update message is received by the location servers evenly distributed in the home region. When a location query message arrives at the destined home region, it is directly replied to by the location server in the first nonempty subregion that the message has reached. Thus, the load of acting as location servers is well balanced over the entire network.

In the network, we count the total number of location update and query messages received by the desired location servers in all the subregions with the same subregion ID. Here, ID update and query messages are also counted as location update and query messages, respectively. To guarantee fairness, we only use subregions that belong to the 19 full regions. In GrLS, since there are seven subregions with subregion ID ranging from 0 to 6 in each home region, seven numerical values regarding these seven subregion IDs can be collected. These values are normalized by the total number of location update and query messages generated by all the nodes during simulation. These normalized values are termed normalized LS load. We evaluate GrLS with 60% group nodes in four ad hoc network scenarios: a 450-node quasi-static, a 450-node mobile, a 900-node quasi-static, and a 900-node MANET.

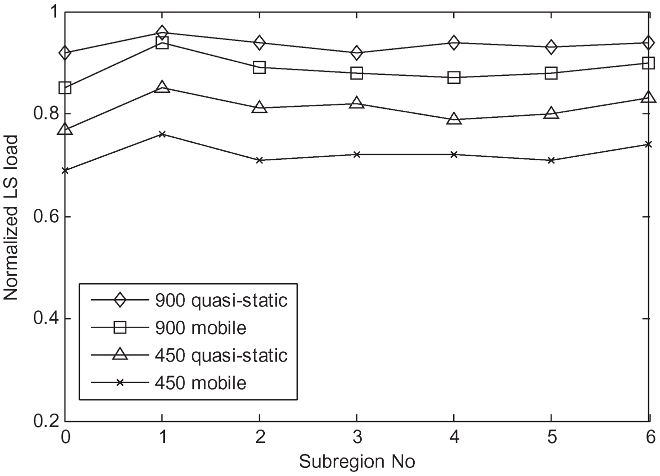


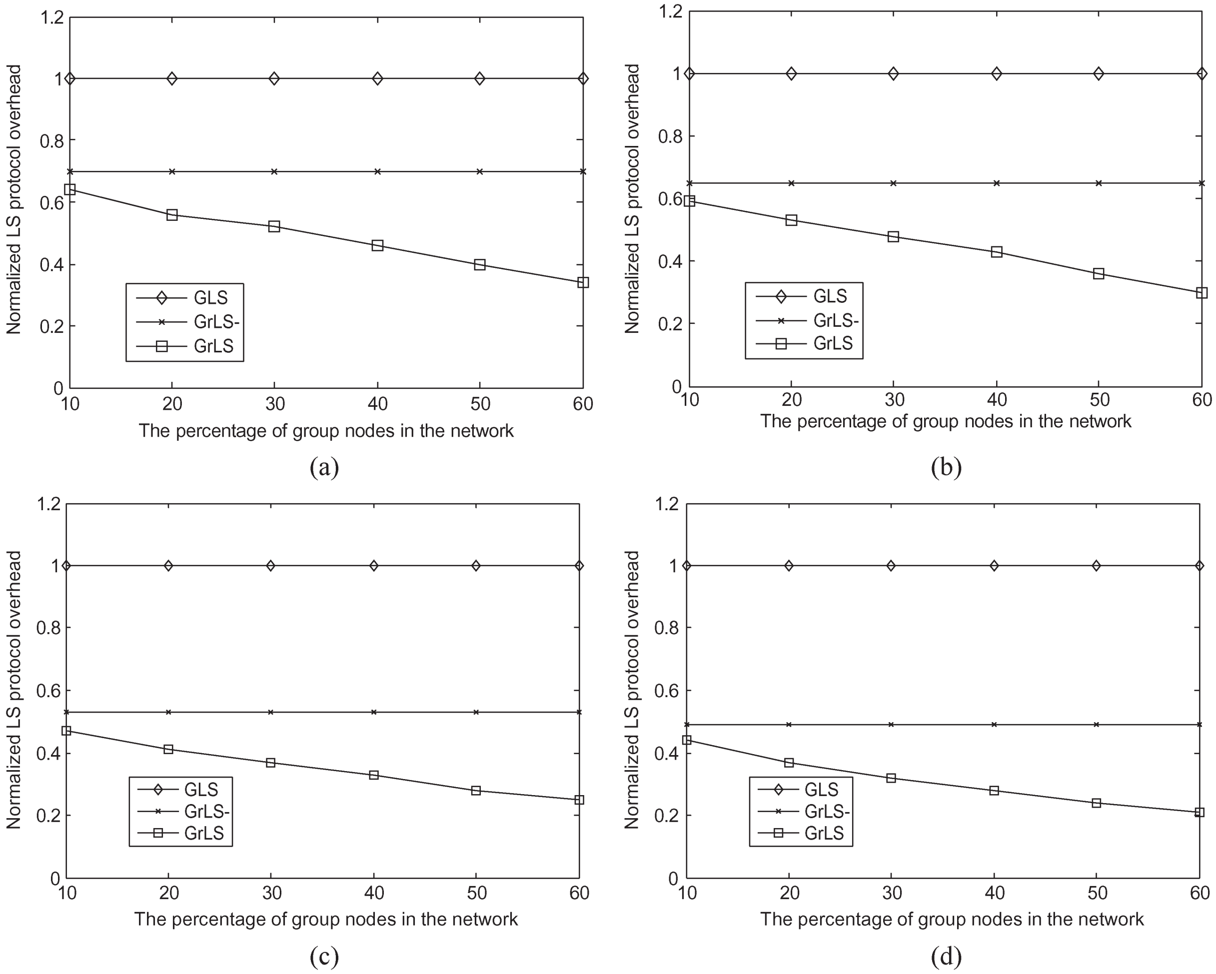
Fig. 7. Comparison of normalized LS load borne by different subregions.

Fig. 7 plots the normalized LS load in different kinds of subregions under the four network scenarios. It shows that the LS load is approximately evenly distributed in the network. For each network scenario, the load borne by subregion 0 is always less than the other subregions. This is because most of the location query messages are received by border subregions (subregions 1–6) and cannot reach the central subregion. Compared with 900-node networks whose node density is 100 nodes/km^2, the LS load borne by subregions in both 450-node networks is lower. This is because some subregions are empty in 450-node networks due to low node density. From Fig. 7, in 900- and 450-node networks, the LS load in mobile networks is always lower than the quasi-static networks. We think the reason for this is that high node mobility causes more drops of location update and query messages. Hence, for GrLS, the location service load is more evenly distributed in networks with both higher node density and slower node mobility.

B. *LS Protocol Overhead*

Here, we compare the LS protocol overhead of GLS, GrLS-, and GrLS with 60% group nodes. The four network scenarios used in Section IV-A are still adopted. In each network, every node initiates a location query to look up the location of a randomly chosen destination at times randomly distributed between 45 and 900 s. The first 45 s are used for nodes to send the initial location update messages to their location servers. When a node sends out a location query message, a location query timer is also set for this message. If no location reply is returned when the timer expires, the node does not resend the location query. If a location reply is successfully received before the timer expires, the node sends a data packet of size 128 B to that destination using the replied location.

In each network, we count all the LS protocol messages for each location service protocol. The LS protocol messages of GrLS include location update, query, reply messages, ID update, query, reply messages, and both location and ID handoff messages. The LS protocol overhead is calculated by the number of LS protocol messages transmitted, with each hop-wise transmission of the protocol message as one transmission. Then, we evaluate the normalized LS protocol overhead (normalized by the number of LS protocol messages generated by GLS). Hence, the normalized LS protocol overhead of GLS is always 1. In addition, since most groups are purposely formed by nodes, group membership rarely changes. So we simulate a small quantity of group nodes join/leave events to verify the



(a) (b) (c) (d)

Fig. 8. Comparison of normalized LS overhead in (a) 450-node quasi-static network, (b) 900-node quasi-static network, (c) 450-node mobile network, and (d) 900-node mobile network.

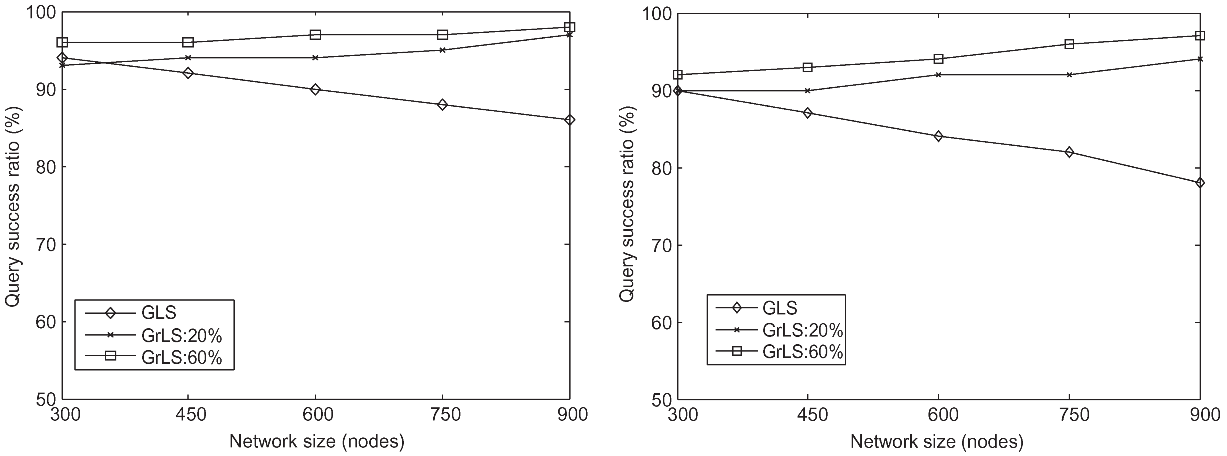
effectiveness of location service handoff. However, we do not count the overhead.

Fig. 8 plots the normalized protocol overhead of all three protocols. It shows that GLS always has the maximum overhead. In mobile networks, the gap between GLS and the other two protocols is much larger than in static networks. GLS incurs a high protocol overhead because it relies on node chain consisting of mobile nodes to update and query location information. In a grid with high hierarchy level, a location update message needs to travel almost the whole grid to search its location server. Furthermore, huge amounts of location update messages are triggered in highly mobile networks because nodes frequently cross grid boundaries. Both GrLS- and GrLS rely on home regions with fixed locations to update and query location information. Hence, they are more robust to node mobility. In addition, only one message needs to be sent to its home region per location update, and, at most, seven location servers need to be updated within the home region. So both GrLS- and GrLS incur a lower protocol overhead than GLS.

Compared to GrLS-, the protocol overhead of GrLS is significantly reduced. This is because the nodes that have formed groups, except the group leaders, do not need to send location update messages to their home regions in GrLS. ID update, query, and reply are first introduced by GrLS. Since we assume relatively stable groups, the reactive ID update rarely occurs. Both ID query and ID reply are only triggered when a location query message sent for a group member has reached the group home region. Moreover, both are just one-hop transmission. So these three new control messages account for a small portion of the LS protocol messages. In addition, we have proved that only one handoff message is needed each time a node leaves or enters one subregion. The amount of handoff messages also depends on node mobility. A higher node mobility leads to more handoff messages. Hence, the amount of handoff messages roughly stays the same in both GrLS- and GrLS. So the saving of location update messages contributes to the reduction of protocol overhead of GrLS. As the percentage of group nodes increases in the network, more reductions of LS protocol overhead are achieved by GrLS.

C. Query Success Ratio

The objective of the location service is to help the source node get the location of a destination. Hence, an important metric to evaluate the location service protocol is the query success ratio. The query success ratio is the ratio of the number of location replies received by all the sources to the number of location queries initiated by all the sources. As stated in



(a)

(b)

Fig. 9. Comparison of query success ratio in (a) quasi-static networks and (b) mobile networks.

Section IV-B, each node initiates a location query to look up the location of a randomly chosen destination. If the location query fails, no retransmission is triggered. Here, we compare GrLS with GLS. To investigate the effect of group location management on GrLS, we choose two cases for evaluation, i.e., where the percentage of group nodes is 20% in the network and 60% in the other.

Fig. 9(a) and (b) depicts the query success ratio as a function of the network size for GLS and GrLS with 20% group nodes and for GrLS with 60% group nodes. The difference between Fig. 9(a) and (b) is that the networks used in Fig. 9(a) are all quasi-static, but the networks used in Fig. 9(b) are all mobile. The results show that the query success ratio of GLS is always the lowest and quickly drops as the network size increases. Moreover, the query success ratio of GLS is much lower in mobile networks than in quasi-static networks. As explained in Section IV-B, GLS is the most susceptible to node mobility because it relies on node chains. Furthermore, as the network size increases, the node chains for both location update and query become longer and weaker, which reduce the query success ratio and the location information accuracy.

With the increasing network size, the node density also becomes higher. In GrLS, more nodes can act as location servers in each home region due to high node density. Since the query success ratio is relatively high at 300 nodes for GrLS, it slowly increases when the network size goes beyond 300. In addition, as the percentage of group nodes increases from 20% to 60%, more source–destination pairs are within the same group. It increases the probability that the source node can immediately get the location of the destination, which also helps improve the query success ratio. By using group location management, GrLS has a very good performance under traffic patterns with locality. Like other protocols, a high node mobility also reduces the performance of GrLS, as shown by the query success ratios in Fig. 9(a) and (b).

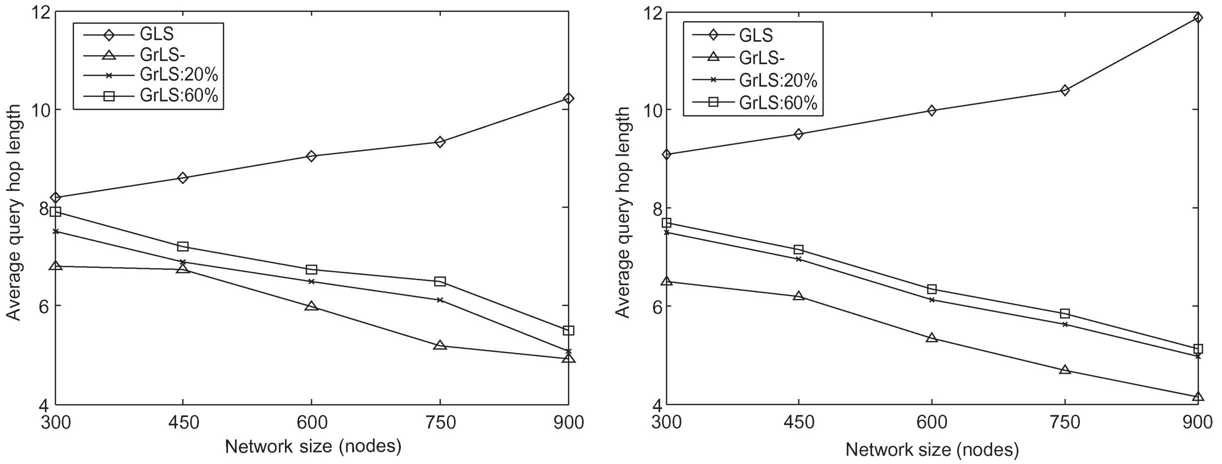
D. Average Query Hop Length

If the location query is for a group node, GrLS will forward it to the group home region when the location information of the group node is found to have been disabled in the original location servers. If the group node is exactly a group leader, the location server in the group home region will immediately answer the location query; otherwise, the location query will continuously be forwarded to the leader of the group node. Thus, the query hop length of GrLS is incremented when group nodes are queried. To investigate how much the query length is affected, we compare the average query hop length in GLS, GrLS-, and GrLS. Similarly, to see the effect of group location management on GrLS, we still adopt the two different cases of GrLS, as used in Section IV-C.

Fig. 10 plots the average query hop length under the four protocol cases. It shows to all, except the GLS, that the average query hop length drops as the network size increases. For GLS, the query success ratio becomes lower as the network size increases, which also leads to a longer average query hop length. The average query hop length of GrLS- is shorter than GrLS because it does not need the forwarding of the location query messages for group nodes to their group leaders. For GrLS with 60% group nodes, its average query hop length is longer than GrLS with 20% group nodes, but not much. This also benefits from the locality. When most of the nodes have formed groups, the probability that a source–destination pair is within the same group becomes higher. When this happens, the query hop length is 0, which helps reduce the average query hop length. So the increment in the average query hop length is insignificant, although the percentage of group nodes increases from 20% to 60% in the network.

V. CONCLUSION

In this paper, we have proposed the first GrLS protocol. By exploiting group mobility, GrLS provides group location management for nodes that have formed groups, thereby significantly reducing the protocol overhead. Moreover, GrLS supports seamless handoff between single location management and group location management. Extensive simulations are conducted to compare GrLS with GLS and GrLS-, which is GrLS without utilizing group location management. The results show that GrLS has a decent load balance, low protocol



(a) (b)

Fig. 10. Comparison of average query hop length in (a) quasi-static networks and (b) mobile networks.

overhead, and high query success ratio. The cost that GrLS pays for performance improvement is the increment in the average query hop length compared to GrLS-. Even so, it is still smaller than GLS. A good location service protocol should be efficient, scalable, robust, load balanced, and locality aware. GrLS shows all these characteristics.

The performance of GrLS can be improved by the following optimization techniques.

1) Location cache can be used. When forwarding a location update message, a node adds the location information it learns from the message to its location cache. The node associates a relatively short timeout value with the cached location information.
2) When a node relaying a location query message finds that it is just the destination that the location query is for, it directly sends location reply to the source and drops the location query message.
3) Location maintenance is used between two communication partners. When data transmission is conducted between a pair of nodes, their location information is periodically piggybacked to the data packets destined for the other end. Thus, they can know each other's accurate location information without querying them again.

APPENDIX

Theorem 1: Assume there exists a set of nodes $\{s_1, s_2, \ldots, s_{m-1}, s_m | s_1 < s_2 < \cdots < s_{m-1} < s_m\}$ in subregion SR. When 1) node $s_i (i \in \{1, 2, \ldots, m-1, m\})$ leaves SR or 2) node $s_j (j \notin \{1, 2, \ldots, m-1, m\})$ enters SR, only one location information handoff message is needed.

Proof: As described in Section III-B, the node ID space is assumed to be circular in the clockwise direction from small IDs to large IDs, as shown in Fig. 11. According to our strategy, a node $s_k (k \in \{1, 2, \ldots, m-1, m\})$ will be recruited as location servers by the nodes, which have selected this home region and have the ID between s_{k-1} and s_k. Here, s_0 represents s_m.

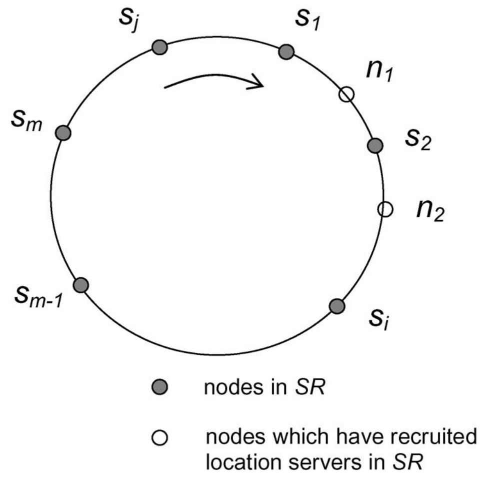


Fig. 11. Circular node ID space.

When node $s_i (i \in \{1, 2, \ldots, m-1, m\})$ leaves SR, it needs to hand off all the location information stored in it to other nodes in SR. As we have analyzed, the IDs of all the sources that have stored location information in s_i are between s_{i-1} and s_i. Since s_i leaves, their new location servers will be the same node s_{i+1} by our strategy. Hence, s_i will hand off all the location information to s_{i+1}. So only one location information handoff message from s_i to s_{i+1} is needed.

Without loss of generality, we assume $s_k < s_j < s_{k+1} (k \in \{1, 2, \ldots, m-1, m\}$ and $j \notin \{1, 2, \ldots, m-1, m\})$. Here, s_{m+1} represents s_0. When node s_j enters SR, it will be the new location server for all the nodes that have selected this home region and have the ID between s_k and s_j. Since all of these nodes have recruited the same location server s_{k+1}, only s_{k+1} needs to hand off their location information to s_j. So only one location information handoff message from s_{k+1} to s_j is needed.

ACKNOWLEDGMENT

The authors would like to thank the two anonymous reviewers for their valuable comments.

REFERENCES

[1] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AoA," in *Proc. INFOCOM*, Mar. 30–Apr. 3, 2003, vol. 3, pp. 1734–1743.

[2] Y. Ko and N. Vaidya, "Location-aided routing in mobile ad hoc networks," in *Proc. MobiCom*, Oct. 25–30, 1998, pp. 66–75.

[3] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. MobiCom*, Aug. 6–11, 2000, pp. 243–254.

[4] R. Jain, A. Puri, and R. Sengupta, "Geographical routing using partial information for wireless ad hoc networks," *IEEE Pers. Commun.*, vol. 8, no. 1, pp. 48–57, Feb. 2001.

[5] L. Blazevic, J.-Y. L. Boudec, and S. Giordano, "A location-based routing method for mobile ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 4, no. 2, pp. 97–110, Mar./Apr. 2005.

[6] X. Wu, "VDPS: Virtual home region based distributed position service in mobile ad hoc networks," in *Proc. ICDCS*, Jun. 6–10, 2005, pp. 113–122.

[7] S. M. Das, H. Pucha, and Y. C. Hu, "Performance comparison of scalable location services for geographic ad hoc routing," in *Proc. INFOCOM*, Mar. 13–17, 2005, vol. 2, pp. 1228–1239.

[8] I. Stojmenovic, "A routing strategy and quorum based location update scheme for ad hoc wireless networks," SITE, Univ. Ottawa, Ottawa, ON, Canada, Tech. Rep. TR-99-09, 1999.

[9] Z. J. Haas and B. Liang, "Ad hoc mobility management with uniform quorum systems," *IEEE/ACM Trans. Netw.*, vol. 7, no. 2, pp. 228–240, Apr. 1999.

[10] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proc. MobiCom*, Aug. 6–11, 2000, pp. 120–130.

[11] Y. Xue and B. Li, "A scalable location management scheme in mobile ad hoc networks," in *Proc. IEEE Conf. Local Comput. Netw.*, Nov. 14–16, 2001, pp. 102–111.

[12] K. Wolfgang, F. Holger, and W. Jorg, "Hierarchical location service for mobile ad hoc networks," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 8, no. 4, pp. 47–58, Oct. 2004.

[13] B. Li, "On increasing service accessibility and efficiency in wireless ad hoc networks with group mobility," *Wirel. Pers. Commun.*, vol. 21, no. 1, pp. 105–123, Apr. 2002.

[14] X. Wu and B. Bhargava, "AO2P: Ad hoc on-demand position-based private routing protocol," *IEEE Trans. Mobile Comput.*, vol. 4, no. 4, pp. 335–348, Jul./Aug. 2005.

[15] B. An and S. Papavassiliou, "A mobility-based clustering approach to support mobility management and multicast routing in mobile ad hoc wireless networks," *Int. J. Netw. Manag.*, vol. 11, no. 6, pp. 387–395, Nov./Dec. 2001.

[16] K. H. Wang and B. Li, "Group mobility and partition prediction in wireless ad hoc networks," in *Proc. IEEE ICC*, Apr. 28–May 2, 2002, vol. 2, pp. 1017–1021.

[17] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A group mobility model for ad hoc wireless networks," in *Proc. ACM Int. Workshop Model. Anal. Simul. Wireless Mobile Syst.*, Aug. 20, 1999, pp. 53–60.

[18] W.-T. Chen and P.-Y. Chen, "Group mobility management in wireless ad hoc networks," in *Proc. IEEE VTC—Fall*, Oct. 4–9, 2003, vol. 4, pp. 2202–2206.

[19] K. Chen and K. Nahrstedt, "Effective location-guided tree construction algorithms for small group multicast in MANET," in *Proc. INFOCOM*, Jun. 23–27, 2002, vol. 3, pp. 1180–1189.

[20] X. Zeng, R. Bagrodia, and M. Gerla, "GloMoSim: A library for parallel simulation of large-scale wireless networks," in *Proc. 12th Workshop Parallel Distrib. Simul.*, May 26–29, 1998, pp. 154–161.

Hui Cheng received the B.Sc. and M.Sc. degrees in computer science from Northeastern University, Shenyang, China, in 2001 and 2004, respectively, and the Ph.D. degree in computer science from The Hong Kong Polytechnic University, Kowloon, Hong Kong, in 2007.

He is currently a Research Associate with the Department of Computing, The Hong Kong Polytechnic University. His research interests include multicast routing, QoS mobile group communication, and evolutionary computation for dynamic optimization problems.

Jiannong Cao (M'93–SM'04) received the B.Sc. degree in computer science from Nanjing University, Nanjing, China, in 1982 and the M.Sc. and Ph.D. degrees in computer science from Washington State University, Pullman, in 1986 and 1990, respectively.

He was on the faculty of computer science with James Cook University, Townsville, Australia; the University of Adelaide, Adelaide, Australia; and City University of Hong Kong, Kowloon. He is currently a Professor with the Department of Computing, The Hong Kong Polytechnic University, Kowloon, where he is also the Director of the Internet and Mobile Computing Laboratory. He has published over 160 technical papers. His research interests include parallel and distributed computing, networking, mobile computing, fault tolerance, and distributed software architecture and tools.

Dr. Cao is a member of the IEEE Computer Society, the IEEE Communication Society, and the Association for Computing Machinery (ACM). He is also a member of the IEEE Technical Committee on Distributed Processing, the IEEE Technical Committee on Parallel Processing, the IEEE Technical Committee on Fault Tolerant Computing, and the Technical Committee on Computer Architecture of the China Computer Federation. He has served as a member of the editorial boards of several international journals, as a reviewer for several international journals and conference proceedings, and an organizing/program committee member for many international conferences.

Hsiao-Hwa Chen (SM'00) received the B.Sc. and M.Sc. degrees in electrical engineering from Zhejiang University, Hangzhou, China, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from the University of Oulu, Oulu, Finland, in 1990.

He was the Founding Director of the Institute of Communications Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C. He was also an Adjunct Professor with Zhejiang University and Shanghai Jiaotong University, Shanghai, China. He is currently a Full Professor with the Department of Engineering Science, National Cheng Kung University, Tainan, Taiwan. He has authored or coauthored over 200 technical papers in major international journals and conference proceedings, five books, and several book chapters in the areas of communications, including the books *Next Generation Wireless Systems and Networks* (Wiley, 2005) and *The Next Generation CDMA Technologies* (Wiley, 2007).

Dr. Chen has been an active volunteer for various technical activities of the IEEE for over 20 years. He is currently serving as the Chair of the IEEE ComSoc Radio Communications Committee and the Vice Chair of the IEEE ComSoc Communications and Information Security Technical Committee. He has served or is serving as symposium Chair/Co-Chair of many major IEEE conferences, including VTC, ICC, Globecom, WCNC, etc. He has served or is serving as Associate Editor or/and Guest Editor of numerous important technical journals in communications. He is serving as the Chief Editor (Asia and Pacific) for Wiley's *Wireless Communications and Mobile Computing* (WCMC) and Wiley's *International Journal of Communication Systems*. He is currently the Editor-in-Chief of Wiley's *Security and Communication Networks* (www.interscience.wiley.com/journal/security).

Hongke Zhang received the M.Sc. and Ph.D. degrees in communication and information engineering from the University of Electronic Science and Technology of China, Chengdu, China.

He is currently a Professor with the College of Electronics and Information Engineering, Beijing Jiaotong University, Beijing, China. He is also the Chief Scientist of the project "a universal network architecture for supporting pervasive services" founded by the 973 Program of China. His research interests include computer communication and router design.