

A scalable mobility-based replica allocation scheme in a mobile ad-hoc network

Sun-Kyum Kim¹ · Ji-Hyeun Yoon¹ · Kwang-Jo Lee¹ ·
Jae-Ho Choi¹ · Sung-Bong Yang¹

Published online: 2 April 2015
© Springer Science+Business Media New York 2015

Abstract Various issues on network routing in a mobile ad-hoc network (MANET) have been studied with outstanding results. However, there is still room for improvement with regard to data accessibility, especially when the number of nodes increases in the network. To enhance the data accessibility in a MANET, a few replica allocation schemes have been proposed. In these schemes, the nodes are partitioned into groups so that each group has as many different data items as possible. However, grouping in a MANET requires an enormous amount of communication cost, since the network topology is changed dynamically. In this paper, we propose a scalable replica allocation scheme in which the nodes are grouped based on the mobility of the nodes. Grouping in the proposed scheme is done with one-hop communications in a fully distributed manner. Within each group, replicas are allocated to the group nodes based on the estimated data access frequencies that are computed periodically by the allocator of the group. It turns out that grouping with mobility and utilizing the estimated access frequency information enhances the stability of the topology of each group as well as the data accessibility. We evaluate the proposed scheme by simulations with the Network Simulator

NS-3 along with the analytical evaluations. The results of simulations and analytical evaluations demonstrate the effectiveness of the proposed scheme in terms of communication cost. The proposed scheme reduces the communication cost by 347.1 % on average over the existing replica allocation scheme, while achieving comparable data accessibility.

Keywords Replica allocation · MANET · Data accessibility · Scalability · Grouping · Mobility · Estimated frequency

1 Introduction

With the advances in wireless communication technologies and the popularity of mobile devices, a mobile ad-hoc network (MANET) has attracted considerable attention in the network community [1–4]. A MANET is a self-organizing [5], rapidly deployable network that consists of wireless nodes without a fixed infrastructure. A large variety of MANET applications have been introduced [6]. For example, a MANET can be used in special situations, where installing infrastructure may be difficult or even infeasible, such as battlefields or disaster areas. In some applications such as battlefield applications, hundreds or even thousands of nodes must be supported [7].

In a MANET, network divisions occur frequently and inevitably since the nodes move freely in the network, causing some data to be inaccessible to some of the nodes. Hence, data accessibility is often an important performance metric in a MANET [1]. Various schemes have recently been proposed for replica allocation in a MANET [1–3]. These schemes utilize grouping in replica allocation to achieve high data accessibility. Although grouping is an effective way to achieve high data accessibility, the traditional grouping

✉ Sung-Bong Yang
sbyang@yonsei.ac.kr

Sun-Kyum Kim
skyum@yonsei.ac.kr

Ji-Hyeun Yoon
y_chick@yonsei.ac.kr

Kwang-Jo Lee
kjlee5435@yonsei.ac.kr

Jae-Ho Choi
redcolor25@yonsei.ac.kr

¹ Department of Computer Science, Yonsei University, Seoul, Korea

schemes for replica allocation may lead to a large communication cost to build groups. Consequently, the traditional schemes can be used only in mesoscale ad-hoc networks [1].

In this paper, we propose a scalable replica allocation (SRA) scheme to resolve the communication cost problem especially when the number of nodes in the network increases. We observe that grouping should be done with great care in order to reduce the communication cost and to achieve high data accessibility as well. With this motivation in mind, we devise a mobility-based grouping method in which each group can be built by communicating the mobility information with only the neighbor nodes—the nodes within a one-hop distance. Consequently, we can achieve much lower communication cost as the number of nodes in the network increases.

The proposed SRA scheme has two phases. In the first phase the nodes in the network are grouped based on the mobility values of the nodes. The mobility value of each node is the approximated average speed of the node that is acquired by computing the division of the distance between the initial position and the final position of the node in each replica relocation period by the duration of a period. The node with lower mobility than its neighbor nodes becomes the central node of the group to ensure greater group stability. In the second phase, the central node in each group allocates replicas to the group nodes based on the estimated access frequencies that are computed by the central node (i.e., the allocator) at the replica relocation time. The central node first sorts the data items in terms of their estimated frequencies. Then it greedily allocates the data items with higher estimated frequencies to itself, and allocates the rest of data items to the group nodes in sequence in a round-robin manner. With the help of the SRA scheme, we aim to reduce the communication cost, while still achieving good data accessibility.

We also introduce a mobility-based allocation to replace the round-robin method in the SRA scheme. It has been found that grouping based on mobility and utilizing the estimated access frequency information enhances the stability of the topology of each group as well as the data accessibility.

The technical contributions of this paper are summarized below.

- *Grouping method* We devise a novel grouping method for replica allocation with communication among only one-hop neighbor nodes. After the grouping is complete, the network topology of each group is a set of star networks. The central node in each group has the lowest mobility among the group nodes.
- *Replica allocation method* We propose an effective replica allocation scheme, called SRA, based on the roles of the nodes and the estimated access frequencies. The central node of each group plays the role of the replica allocator. The estimated data access frequencies are updated dynam-

ically at each relocation time. We also modify the proposed replica allocation method by considering the mobility of each node.

- *Analytical evaluations and simulations* We verify the proposed scheme by analytical evaluations as well as simulation using the NS-3 network simulator. The results demonstrate that the proposed replica allocation scheme reduces the communication cost considerably as the number of nodes increases and achieves high data accessibility.

The rest of this paper is organized as follows. Section 2 provides a brief overview of related work. In Sect. 3, we describe the system model. In Sect. 4, the proposed scheme is described in more detail and the analytical evaluations are provided. In Sect. 5, we present the simulation results of the proposed scheme. We conclude the paper in Sect. 6.

2 Related work

The major research issues on replica allocation in a MANET are “when and where to” allocate replicas. Several schemes have been proposed to improve data accessibility by replicating data items in a MANET. We use the terms ‘data items’ and ‘replicas’ interchangeably hereafter. There are two replica allocation schemes in [1]; one is the static access frequency scheme (SAF) and the other is the dynamic connectivity based grouping scheme (DCG). In SAF, each node greedily allocates replicas for itself according to its own access frequencies; that is, the most frequently accessed data item is stored first in its memory, then the next frequently accessed item is stored, and so on, until the memory gets full. Since each node allocates replicas based on its own access frequencies in SAF, the replicas need not be reallocated. Consequently, SAF requires minimal communication cost, but suffers from poor data accessibility.

Figure 1 illustrates the other allocation scheme, DCG, in which groups are constructed by finding biconnected components; each biconnected component becomes a group. Note that a biconnected component in a graph is a maximal biconnected subgraph. The network is considered as a graph in which the nodes are the points and the communication links are the edges. After grouping is complete in DCG, the lowest suffix node in a group—the node with an ID that is lexicographically the smallest—is selected as the replica allocator of the group. Although DCG achieves higher data accessibility than SAF, the communication cost for grouping is extremely high due to the broadcast to find biconnected components in the network.

Other researchers have also addressed data replication issues in a MANET. Karumanchi et al. proposed the quorum-based scheme to improve data accessibility [8]. The scheme targets applications where inaccuracy is preferred to no infor-

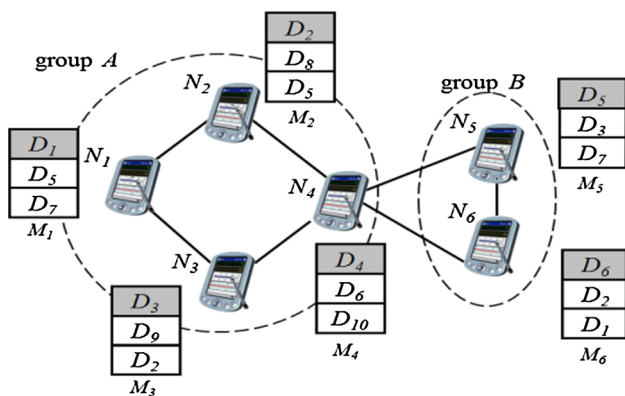


Fig. 1 Grouping in the DCG scheme

mation at all. Luo et al. also proposed a set of protocols that use a gossip-based multicast protocol [9]. The objective of this technique is to maximize the probability of accessing up-to-date data items. Contrary to the aforementioned schemes, we focus on reducing the communication cost with high data accessibility for scalable replica allocation in a MANET.

With respect to data accessibility, grouping is a very important tool to achieve low communication cost. Several grouping schemes were proposed recently. The least cluster change algorithm maintains groups in an event-driven way [10]. When a node moves out of the predefined range, the algorithm reconstructs groups. Adaptive clustering for mobile wireless network has also been proposed [11]. This algorithm reduces the communication cost for building groups, but it creates too many groups as time passes. Consequently, almost every node forms a single-node group. A three-hops between adjacent cluster-heads scheme has also been introduced [12]. Although these schemes can construct groups with low communication cost, they are mainly designed for reducing the routing control overhead and do not consider query processing. However, our proposed replica allocation scheme attempts to decrease the communication cost as the number of nodes in the network increases.

3 System model

We model a MANET in an undirected graph $G = (N, E)$ consisting of a finite set of nodes N and a finite set of communication links E , where each link connects two nodes within the communication range. The wireless medium is subject to losses such as fading and multipath effects. Therefore, signal reception is modeled as being binary—either a node is connected to another node or not. For our model, let the network consist of heterogeneous nodes embedded in a planar region. The system in this paper is assumed to work in an ad-hoc mobile network where each node has its own memory space and issues queries via broadcast. Since most replica allocation schemes in a MANET assume broadcast-based query

processing, we adopt this processing scheme in this paper. Further details of our assumptions are given below.

Each node in a MANET has a unique identifier. The nodes in the MANET are denoted by $N = \{N_1, N_2, N_3, \dots, N_m\}$, where m is the total number of nodes in the network. Each node has the memory space of equal size to hold data items. This assumption is widely accepted in the literature [3].

All data items are of equal size. Each data item has a unique identifier. The data items are denoted by $D = \{D_1, D_2, D_3, \dots, D_n\}$, where n is the total number of data items. Each node N_i is assumed to own D_i throughout the simulation; we call D_i the original data item of N_i .

Data items are not updated. This assumption is made for the sake of simplicity (i.e., we do not have to address data consistency or currency issues). Many applications satisfying this assumption have been introduced in previous research [1,2]. Data items are reallocated periodically at a regular interval except in the static allocation scheme.

Each node moves freely within the constraints of maximum speed and is aware of its own speed (or velocity). To compute the speed, each mobile node is equipped with a GPS or some other positioning system. This assumption is widely accepted in the literature [13–15]. Each node measures its mobility periodically. Although such measurement would incur additional computational cost, we do not consider such computational cost throughout this paper because the computation can be done with a simple equation. Moreover, the focus of our paper is not on reducing the computational cost, but the communication cost, while maintaining high data accessibility. Note that the node’s maximum speed varies from 0.1 to 9 m/s.

4 Proposed scheme

4.1 Overview

The proposed scheme consists of two phases: (1) constructing groups in the network utilizing the speed of each node and (2) allocating replicas to the nodes in each group. At a regular time interval (called the relocation period) [1], the nodes in the network are newly partitioned into groups. For constructing groups in the proposed scheme, each node should maintain a neighborhood list—a list of nodes connected to the node within one hop. We call such nodes neighbor nodes. Each node acquires its neighborhood list by exchanging simple messages with only its neighbor nodes. The proposed scheme builds groups based on the neighborhood list as well as the mobility of each node.

After grouping is done, the network topology of each group can be viewed as a set of star networks. The central node in each group allocates replicas to the group nodes, including itself, based on the estimated data access frequencies.

4.2 Grouping method

In the proposed scheme, each node constructs its neighborhood list at every relocation time in order to reflect the current topology. The neighborhood list of a node is maintained as a set of tuples (*node ID, mobility value*). Each node N_i computes its mobility value $m_i(t_s, t_l)$ that denotes an approximate speed recorded between the previous relocation time t_s and the current relocation time t_l . Hence the mobility value of a node indicates the speed of the node during a specific period.

Equation 4.1 shows how to measure the mobility value. In the equation, (x_s, y_s) and (x_l, y_l) are the x and y -coordinates of N_i at times t_s and t_l , respectively. Hence, we divide the distance between the positions t_s and t_l by the length of the relocation period to obtain a value that indicates how fast N_i has been moved.

At every relocation time, each node N_i sends $(ID_i, m_i(t_s, t_l))$ to every other node within one hop. Hence, each node will receive $(ID_j, m_j(t_s, t_l))$ from each node N_j , which is within a one-hop distance. After each node has exchanged the ID and the mobility value, if a node N_i finds that it has the lowest mobility value, then N_i becomes the group allocator as a leader [13] and tells each node N_j in its neighborhood list that N_j is now a group member. Note that when a node receives multiple requests to become a group member, the node accepts only the first request. Algorithm 1 outlines how to construct groups in the proposed scheme.

$$m_i(t_s, t_l) = \left| \frac{\sqrt{(x_s - x_l)^2 + (y_s - y_l)^2}}{t_s - t_l} \right| \tag{4.1}$$

Algorithm 1. Pseudo code for constructing groups

```

01: At relocation time  $t_l$  // The previous relocation time
    is  $t_s$ .
02: constructing_group() {
03: Each node  $N_i$  sends  $(ID_i, m_i(t_s, t_l))$  to each of its neighbor
    nodes;
04: while (during the predefined time  $\omega$ ) {
05: if ( $N_i$  receives the mobility value  $m_j(t_s, t_l)$  from  $N_j$ ) { //
     $N_j$  is a neighbor node of  $N_i$ .
06:     if ( $m_i(t_s, t_l) > m_j(t_s, t_l)$ ) {
07:          $N_i$  becomes a member node of a group;
08:         break; }
09:     else  $N_i$  becomes the allocator; }
10: else  $N_i$  becomes the allocator;
11: }
12: if ( $N_i$  is the allocator)  $N_i$  sends join message to each
    of its neighbor nodes;
13: else  $N_i$  receives join message and becomes a mem-
    ber node;
14: }
    
```

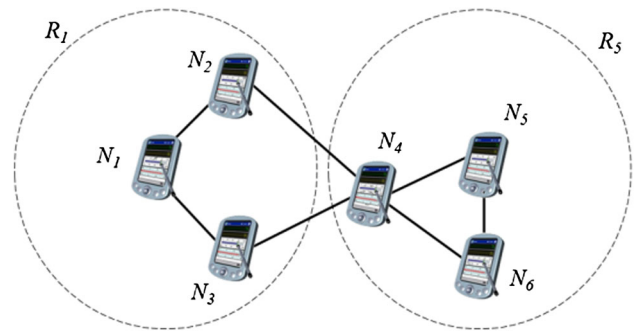


Fig. 2 Grouping based on the neighborhood list

As described in Algorithm 1, each node becomes either the group allocator or a group member. Each node N_i waits for others' mobility values after sending its mobility value to each of its neighbor nodes during the predefined time ω , where ω is the expected maximum time taken to exchange a round of *send-receive* message with the neighbor nodes.

Figure 2 shows an example of grouping based on the neighborhood list. In this figure, the dotted circles R_1 and R_5 denote the communication ranges of nodes N_1 and N_5 , respectively. We assume that N_1 and N_5 have the lowest mobility values among their own neighbor nodes. Therefore, N_1 and N_5 become the allocators and they send *join* messages to N_2 and N_3 , and N_4 and N_6 , respectively. The nodes that receive *join* messages become group members. Consequently, $N_1, N_2,$ and N_3 constitute one group and the rest comprise the other. Note that when a node receives multiple *join* messages, the node accepts only the first message.

4.3 Replica allocation method

After grouping is complete, each group has the star network topology. We now let the central node in each group play the role of the replica allocator. Since we use the mobility value as a criterion for grouping, the allocator of a group could be considered a more 'reliable' node (i.e., the node that moves slowly) than other nodes in the group. Note that if the criterion is changed to incorporate some other factor such as energy (battery power) or memory size, an allocator can be a more 'powerful' or 'spacious' node (i.e., having high power or larger memory space). Since, in the proposed scheme, the allocators are reliable nodes, groups are expected to be more robust than those created by the traditional schemes. The philosophy behind allocating replicas within a group is that 'important' data items should be held by more reliable nodes in a group. In this case, important data items are those items that are accessed more frequently.

For the proposed allocation method, a fully distributed measuring method is introduced to obtain the *estimated data access frequencies* for each node independently. Each node N_i monitors the queries issued by itself and requested by

Table 1 Examples of $esTable_1$ and $esTable_5$

Data item	Estimated access frequency	
	N_1	N_5
D_1	0.23	0.02
D_2	0.16	0.10
D_3	0.24	0.07
D_4	0.12	0.24
D_5	0.04	0.14
D_6	0.03	0.15
D_7	0.07	0.05
D_8	0.00	0.09
D_9	0.06	0.03
D_{10}	0.05	0.11

other nodes—they are not just the group nodes but all other nodes in the network; that is, we measure the ‘popularity’ of data items from the perspective of the observer. While monitoring, N_i accumulates the access count of each data item as well as the total number of queries issued during a period between the previous and current relocation times. With such information, N_i is able to construct its own estimated data access frequency table, called $esTable_i$ (Table 1), each entry of which is a set of pairs (data item ID, the estimated access frequency), where the estimated access frequency to each data item D_j is calculated with the following equation:

$$\begin{aligned} & \text{the estimated access frequency of } D_j \\ &= \frac{\text{total number of accesses to } D_j}{\text{total number of queries issued}} \end{aligned} \tag{4.2}$$

Algorithm 2. Proposed replica allocation method

```

01: At each relocation time
02: /*In each group, the allocator  $N_i$  allocates replicas to the group nodes based on  $esTable_i$ */
03: allocating_replicas (){
04:  $N_i$  sorts the data items in  $esTable_i$  in descending order of access frequency and calls the sorted list  $D\_list_i$ ;
05:  $N_i$  greedily allocates the first  $M_i - 1$  data items from  $D\_list_i$  to itself, except those original items that are owned by other group nodes; //  $M_i$  is the memory size of node  $N_i$ .
06:  $N_i$  allocates the next unallocated item from  $D\_list_i$  to each node in the group in a round-robin fashion until the memory space of all of the group members is full;
07: }
```

Note that it does not matter whether or not each request observed by N_i is eventually satisfied since we are seeking only the popularity of data items from the perspective of N_i . These estimated data item access frequencies provide more ‘up-to-date’ access information—without any assumption of

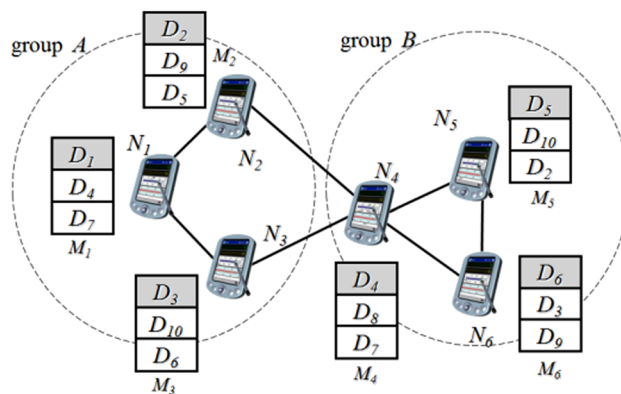


Fig. 3 Replica allocation by the proposed method based on Table 1

the access frequencies—and do not require any additional communication cost. The estimated frequencies of all data items at each node are initialized to one over the total number of data items, that is, $1/n$. Although each node may not know the total number of data items in the system, the node estimates the total number of data items based on its observation.

Since $esTable$ is constructed by each node individually and independently, the table in each node may possess different content. Note that a request query to a data item may go through only a few nodes while many other nodes in the network are not involved in processing the query.

Algorithm 2 describes how replicas are allocated at each relocation time with the proposed replica allocation method. In the algorithm, $esTable_i$ and M_i denotethe $esTable$ and the memory size of node N_i , respectively. Table 1 shows the content of the $esTables$ of the allocator nodes as examples; we assume that the values in the table entries are obtained through the observations made by N_1 and N_5 during a relocation period.

Figure 3 illustrates the results of the data item allocation to the nodes in groups A and B according to the round-robin method. In this example, the memory space of each node can hold up to three data items and N_1 and N_5 are the allocators of groups A and B, respectively. Based on $esTable_1$, $D_list_1 = [D_3, D_1, D_2, D_4, D_7, D_9, D_{10}, D_5, D_6, D_8]$. Note that the sequence of the data items in D_list_1 is sorted in descending order of estimated access frequencies. Now N_1 already has D_1 as its original data item and allocates D_4 and D_7 to itself since N_1 knows D_2 and D_3 are the original items kept at N_2 and N_3 , respectively, and since D_4 and D_7 have the next two highest frequencies according to $esTable_1$. Then N_1 allocates D_9 to N_2 , D_{10} to N_3 , finally D_5 to N_2 , D_6 to N_3 in a round-robin manner. For group B, $D_list_5 = [D_4, D_6, D_5, D_{10}, D_2, D_8, D_3, D_7, D_9, D_1]$. Hence N_5 keeps D_5 , D_{10} , and D_2 , and N_5 similarly allocates D_8 and D_7 to N_4 and D_3 and D_9 to N_6 .

The proposed replica allocation method allocates the replicas in the order of node IDs (i.e., the lowest suffix node first) from the sorted item list; that is, it does not consider any other factors in allocating the replicas. However, once again we use the mobility factor in the allocation and call this method *the mobility-based allocation method*. In this method we need a threshold δ to determine whether a node moves fast; if a node has a mobility value greater than δ , then it is regarded as a fast mover. Algorithm 3 describes the mobility-based allocation method in pseudo code.

Figure 4 shows how the mobility-based method allocates replicas to the nodes in groups A and B, where m_i indicates the mobility value of N_i . In this example, the memory space of each node can hold up to three data items and N_1 and N_5 are the allocators of groups A and B, respectively. Based on D_list_1 , N_1 keeps D_1 , D_2 , and D_4 , since D_1 is the original item of N_1 and D_2 (a popular item) may not be available soon due to the high mobility of N_2 . Then, N_1 allocates D_7 and D_9 to N_3 first since $m_3 < \delta$, that is, N_3 is more stable than N_2 . Finally, N_1 allocates D_{10} and D_5 to N_2 . Note that D_6 and D_8 are ignored in the allocation since there is no room in the group for them. Similarly, for group B, N_5 keeps D_5 , D_4 , and D_{10} , allocates D_2 and D_8 to N_6 , and finally D_3 and D_7 to N_4 .

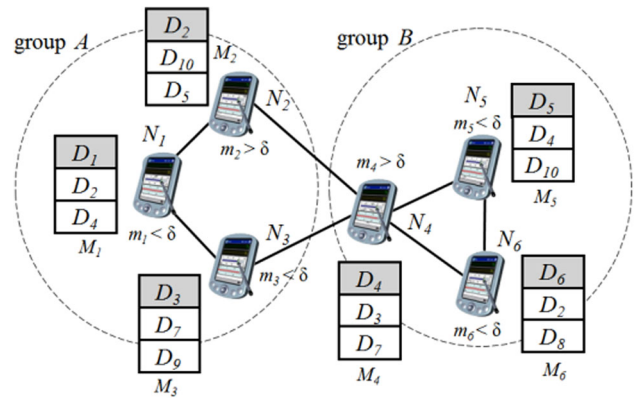


Fig. 4 Mobility-based replica allocation method

allocation, and the cost for accessing the replicas, respectively. In this paper, we count the number of hops for the communication cost. This assumption is widely accepted in the literature [1,3]. First, the grouping cost C_{g-SRA} of the proposed scheme can be estimated as follows:

$$C_{g-SRA} \leq \sum_{c \in CC} 3E_c \tag{4.4}$$

In Eq. 4.4, CC is the set of all connected components in the network, and E_c denotes the number of edges among the nodes in a connected component c . The grouping cost of SRA is at most three times the number of edges in c , since each edge (link) is used twice for sending the IDs and mobility values and once for the group membership notification by each central node.

Next, we formulate C_{r-SRA} of our scheme as follows:

$$C_{r-SRA} \leq \sum_{g \in GG} N_g \tag{4.5}$$

In the above equation, GG is a set of all groups constructed by SRA and N_g is the number of nodes in a group g . Notice that in our scheme the allocator of a group and each of the group members are connected within one hop. Therefore, the allocator sends only one message to each group member. Since the sum of the number of nodes in each group is the number of nodes in the entire network, C_{r-SRA} is at most m . Last, we formulate C_{a-SRA} of our scheme as shown below.

$$C_{a-SRA} \leq \sum_{c \in CC} N_c(2E_c + Diam_c) \tag{4.6}$$

In Eq. 4.6, $Diam_c$ is the diameter of connected component c . Note that the *diameter* of a graph is the longest distance between any pair of vertices in the graph. Hence, $(2E_c + Diam_c)$ is the upper bound on the number of hops to access a data item by a node—it does not matter whether or not the data item has been accessed successfully. Then, we multiply the number of nodes in a connected component c by the term for the access cost of all nodes in c . For C_{a-SRA} , we sum the

Algorithm 3. Mobility-based replica allocation method

- 01: At each relocation time
- 02: /*In each group, the allocator N_i allocates replicas to the group nodes based on $esTable_i$ */
- 03: **allocating_replicas** (){
- 04: N_i sorts the data items in $esTable_i$ in descending order of access frequency and call the sorted list D_list_i ;
- 05: N_i greedily allocates the first M_i-1 data items from D_list_i to itself, except those original items that are owned by other group nodes with a mobility value larger than δ ; // δ is a threshold to determine whether a node moves fast.
- 06: N_i sorts the group nodes based on the mobility in ascending order and calls the sorted list N_list_i ;
- 07: N_i allocates the next unallocated M_j-1 items from D_list_i to the next node N_j from N_list_i until the memory spaces of all of the group members are full or there is no node in N_list_i to allocate;
- 08: }

4.4 Analytical evaluation

In this section we analyze the communication costs of the SAF and DCG replica allocation schemes along with the posed scheme, called SRA. We formulate the communication cost for replica allocation as follows:

$$C_t = C_g + C_r + C_a \tag{4.3}$$

In Eq. 4.3, C_t , C_g , C_r , and C_a denote the total communication cost, the cost for building groups, the cost for replica

costs for all of the connected components. Now C_{t-SRA} can be written as follows:

$$C_{t-SRA} \leq \sum_{c \in CC} 3E_c + m + \sum_{c \in CC} N_c(2E_c + Diam_c) \tag{4.7}$$

We also formulate the communication costs of SAF and DCG similarly. C_{t-SAF} and C_{t-DCG} can be formulated as shown below.

$$C_{t-SAF} \leq \sum_{c \in CC} N_c(2E_c + Diam_c) \tag{4.8}$$

$$C_{t-DCG} \leq \sum_{c \in CC} (E_c + 2E_c d_{max} Diam_c) + \sum_{g \in GG} N_g E_g + \sum_{c \in CC} N_c(2E_c + Diam_c) \tag{4.9}$$

C_{t-SAF} is equal to C_{a-SAF} since there is no grouping or replica reallocation in SAF. In DCG, the grouping cost C_{g-DCG} is much higher than C_{g-SRA} because finding biconnected components requires that each node broadcast its ID to every other node in the network and find biconnected components independently after receiving all of the information it can get. The term $\sum_{c \in CC} (E_c + 2E_c d_{max} Diam_c)$ is the upper bound on the number of hops for broadcasting in DCG. For each connected component c , each node first sends its ID to its neighbor nodes. This entails E_c ; that is, each edge in c is used once. Then, each node N_i receives d_i messages from its neighbor nodes, where d_i is the degree of N_i —the number of neighbor nodes of N_i . Therefore, N_i should transfer these messages to its neighbor nodes one-by-one. At the time each node sends a single message to each neighbor, each edge in c will be used twice; so $2E_c$ is the cost in this case. Since all of the nodes in c should receive the message sent by each node, the upper bound of the overall cost is $2E_c d_{max} Diam_c$, where d_{max} is the maximum degree of the nodes in c . Hence, the total cost C_{g-DCG} for grouping is $O(m^4)$. However, C_{g-SRA} is only at most $O(E_c) = O(m^2)$.

The replica allocation cost C_{r-SRA} is also much smaller than C_{r-DCG} by up to a factor of $O(m^2)$. Although SRA, DCG and SAF have the same upper bound on the accessing cost, since the replicas are allocated within one hop neighbors (h is only 1 in most cases) in SRA, C_{a-SRA} is almost the same as the number of data items, i.e., $O(n)$. Overall, $C_{t-SRA} \ll C_{t-DCG}$, as m increases.

5 Simulation results

5.1 Simulation environments

For the experiments, we use the network simulator NS-3 v.3.10 [14]. The network simulator NS-3 models the network

Table 2 Parameters used in the simulation

Parameter	Value (default value)
Network area	450 × 450 m ²
Number of nodes	40–70 (40)
Communication range	10–90 m (70)
Speed of nodes	0.1–9.0 m/s
Size of memory space	10–80 (10)
Number of data items	40–70 (40)
Query issuing rate	10 s
Relocation period	64–1024 s (256)
Simulation time	6000 s

link problems (e.g., fading, multipath effects, congestion, interference, and so on). The system parameters are determined based on the parameters that are adopted in [1] to ensure a fair comparison. Note that some parameter values are rescaled proportionately since we are using the NS-3 simulator. The number of nodes varies from 40 to 70. Each node has memory space that can hold up to 70 data items. The network area is set to 450 × 450 m² and the length of the communication range varies from 10 to 90 m.

The movement of the nodes follows the random way point model [15], which is one of the most frequently used moving patterns in MANET simulations. In this model, each node remains stationary for a pause time and then it selects a random destination and moves to that destination. After reaching the destination, it again stops for a pause time and repeats this behavior. The maximum speed of a node randomly varies from 0.1 to 9.0 m/s. In the simulator, a node issues a query every 10 s. The replicas are periodically relocated at a specific relocation time [1]. The relocation period varies from 64 to 1024 s. The data access frequencies of node N_i are determined based on the equation, $p_i = 0.5(1 + 0.01i)$, where i is the suffix of the ID [1]. Over 6000 s we simulate and compare the proposed scheme, SRA, with SAF and DCG. Table 2 summarizes the parameters of our simulation environment. Note that the numbers in parentheses indicate default values for the experiments. We evaluate these schemes using the two performance metrics described below while varying the relocation period, the number of nodes, the communication range, and the memory space.

- **Data accessibility** This is the ratio of the number of successful queries to the total number of issued queries.
- **Communication cost** This is the total hop count of data transmission for grouping, replica allocation and accessing data items.

Note that when an experiment varying a particular parameter (e.g., the relocation period, the number of nodes, the communication range, or the memory space) is performed,

the other parameters are set to their default values given in Table 2.

5.2 Experimental results

5.2.1 Effect of the relocation period

First, we examine the effect of the relocation period on each of the three schemes (SRA, DCG, and SAF). Figure 5a, b show the simulation results while varying the relocation period from 64 to 1,024. In this simulation, other parameters except the relocation period are set to their default values as indicated in Sect. 5.1.

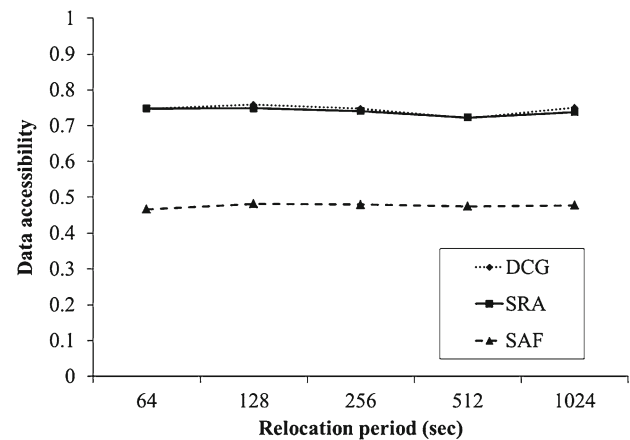
Since SAF does not require grouping or reallocation, each node may hold many duplicated data items. As a result, SAF shows the lowest data accessibility as evidenced by Fig. 5a. In Fig. 5a, SRA shows a similar performance to DCG. Although SRA builds groups in a simple way in order to reduce the communication cost, SRA achieves comparable data accessibility to DCG. The results in Fig. 5a confirm the effectiveness of SRA in that replica allocation based on the estimated access frequencies provides high accessibility. Note that how well the grouping is made is not so important in this environment. As we can see in Fig. 5a, b, SRA improves the data accessibility by an average of 35.2% over SAF.

Next, we examine the communication cost while varying the relocation period. In both the DCG and SRA schemes, the communication cost decreases as the relocation period increases because these schemes perform relocations more infrequently. Figure 5b compares the communication costs of the schemes. We expect the communication cost of SRA to be slightly higher than that of SAF and significantly lower than that of DCG. SRA reduces the communication cost by an average of 219.0% compared to DCG, while SRA costs 88.7% more than SAF. The reason is that the nodes in SRA only communicate with the neighbor nodes (i.e., the nodes within a one-hop communication range) to build a group. However, the nodes in DCG communicate with all of the connected nodes (i.e., the nodes connected within one hop and/or multiple hops).

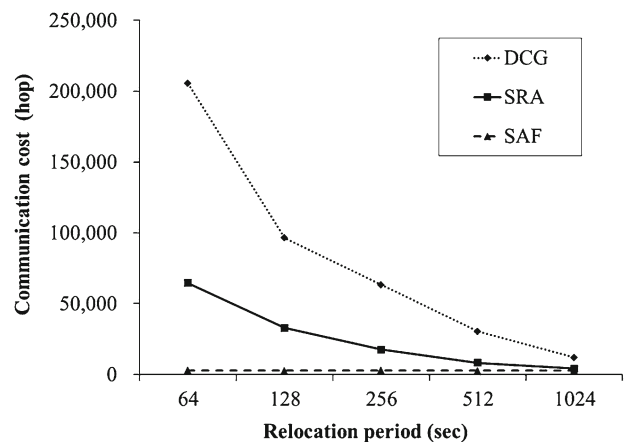
As we can see in Fig. 5b, the difference in communication cost between SRA and DCG narrows tremendously as the relocation period is reduced. This phenomenon proves that DCG requires too much communication for grouping compared with SRA.

5.2.2 Effect of the number of nodes

We evaluate the scalability of the schemes as the number of nodes in the network increases. We vary the number of nodes from 40 to 70 while the other simulation parameters are set to the default values given in Table 2.



(a) Data accessibility

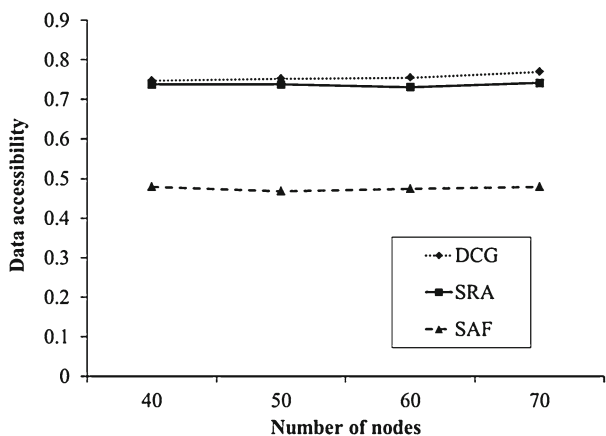


(b) Communication cost

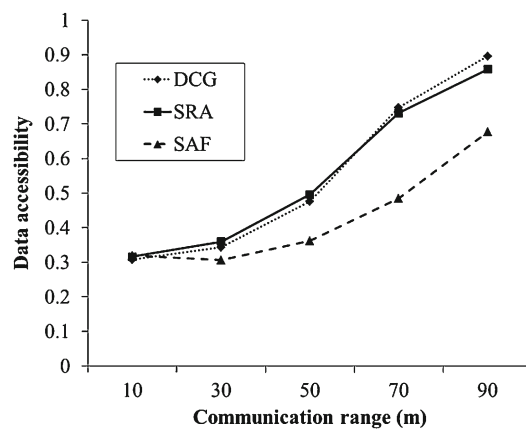
Fig. 5 Effect of the relocation period duration

We expect data accessibility to increase as the number of nodes increases. However, interestingly, as can be seen in Fig. 6a, the data accessibility of all of the schemes does not seem to increase or decrease noticeably. We perform an in-depth analysis of this phenomenon. Since the network simulator NS-3 models the network link problems, as the number of nodes increases, such problems occur more frequently. Consequently, a large amount of requests fail. The average data accessibility of SRA is increased by 35.4% over SAF and shows comparable data accessibility to DCG.

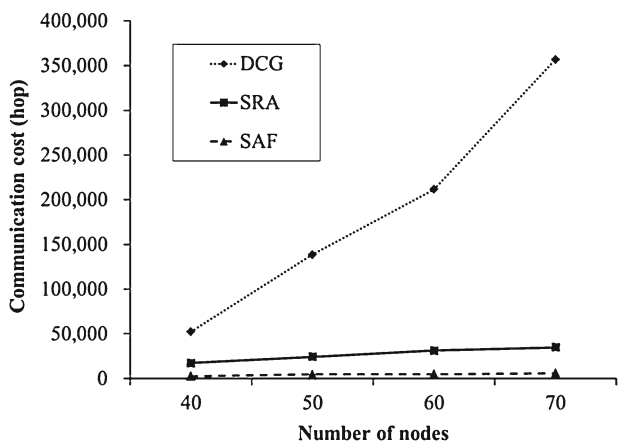
Figure 6b shows the communication costs when the number of nodes increases. As expected, the communication cost of DCG increases almost exponentially as the number of nodes increases. SRA reduces the average communication cost by 597.3% over DCG, but increases the cost by 82.4% compared with SAF. Interestingly, the communication cost of SRA seems to increase minimally. This is because most communication regarding messages and data items is processed within one hop in SRA. Such results demonstrate that SRA



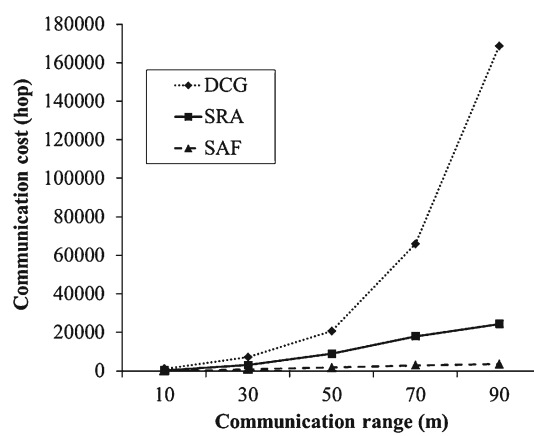
(a) Data accessibility



(a) Data accessibility



(b) Communication cost



(b) Communication cost

Fig. 6 Effect of the number of nodes

Fig. 7 Effect of the length of the communication range

is satisfactorily scalable and can be deployed in a real-world environment.

5.2.3 Effect of the communication range

We compare the effect of the communication range of the nodes for each scheme. We vary the communication range from 10 to 90 while other simulation parameters are set to the default values.

Figure 7a, b show the data accessibility and the communication cost as a function of the communication range, respectively. In Fig. 7a, as the communication range widens, each scheme achieves higher data accessibility. As expected, DCG and SRA outperform SAF. The average data accessibility of SRA is 22.2% higher than that of SAF. However, when the communication range is very short, each scheme shows similar data accessibility since the number of nodes connected to each other is small. When the communication range is very wide, DCG and SRA show almost the same data accessibility, which is superior to that of SAF. SRA reduces

the average communication cost by 382.1% over DCG and increases it by 83.0% over SAF. The results demonstrate the effectiveness of the proposed scheme.

5.2.4 Effect of the memory space

We examine the effect of the memory space of each node for each scheme. We vary the size of the memory space from 0 to 40 and other simulation parameters are set to the default values. In this simulation, all nodes hold only one original data item when the size of the memory space is zero.

Figure 8a, b show the data accessibility and the communication cost as the size of the memory space increases. As shown in Fig. 8a, each scheme shows higher accessibility as the memory size gets larger. It is evident that each node holds more data items when its memory space is large. DCG and SRA have similar data accessibilities since the data items are evenly distributed among the group nodes and hence there are not many duplicates among the replicas in the nodes of each group. However, SAF is linearly affected by the mem-

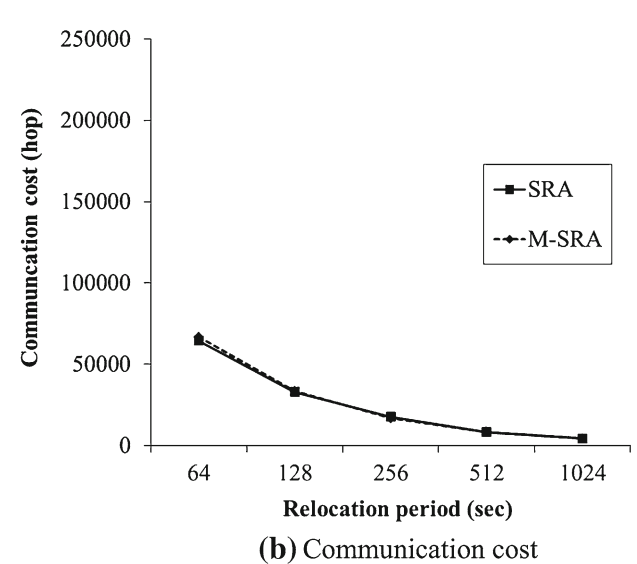
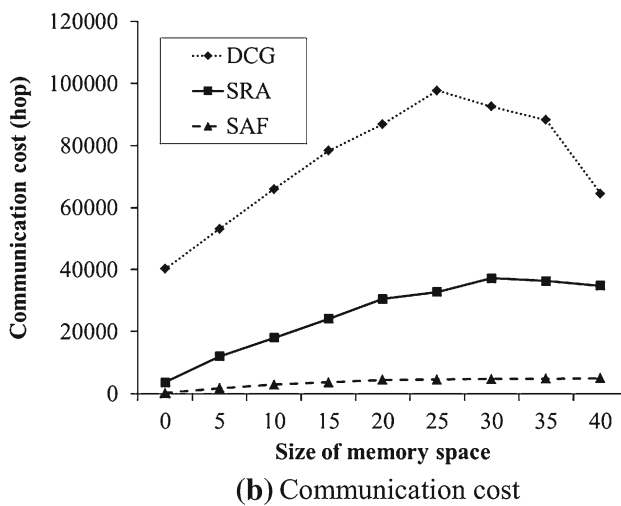
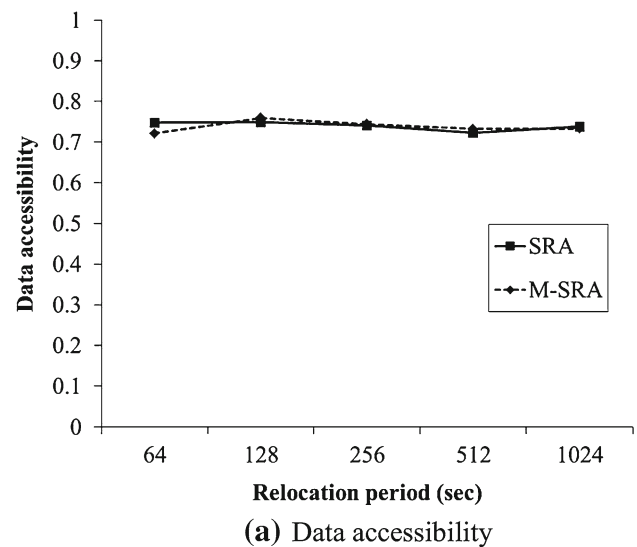
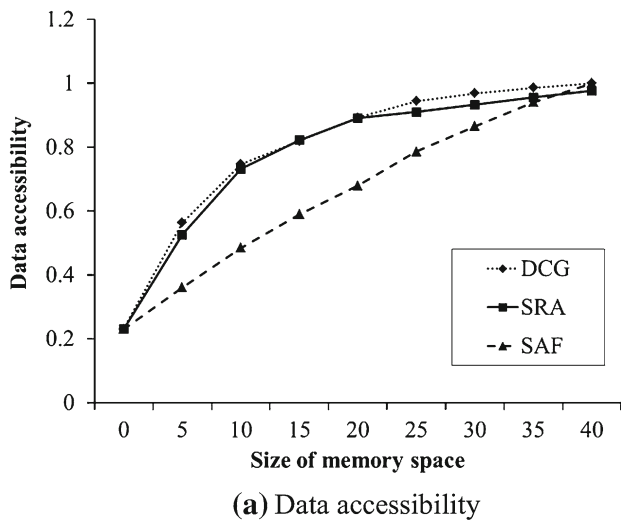


Fig. 8 Effect of the size of the memory space

ory size since each node allocates greedily according to its own access frequency and there are many duplicates among the data items in the nodes. SRA improves the average data accessibility by 15.0% over SAF. Figure 8b shows that as the memory size increases, the communication cost of SRA and SAF increases. This is because the number of the allocated replicas increases as the memory space increases. The average communication cost of SRA is 190.0 and 86.2% less than that of DCG and SAF, respectively. Interestingly, the communication cost of DCG increases up to a memory size of 25, but the cost decreases thereafter. The reason for this is that each node already holds most of the allocated data items. Therefore, the relocation does not significantly affect the communication cost.

A similar curve without a high peak is observed for SRA. The effect is marginal in SRA because the replica relocations occur less frequently than in DCG since DCG should relocate replicas whenever the network topology changes. We per-

Fig. 9 Effect of mobility-based allocation

form an in-depth analysis of this simulation result. In SRA, about 45% of the nodes serve as allocators in our simulation environments. If the allocators are not changed frequently, then the replicas held by the allocators do not change frequently. In other words, the allocators act like the nodes in SAF. Consequently, in SRA, replicas are relocated relatively less frequently than in DCG. In SAF, a node needs to allocate replicas to its memory space and the allocation does not change because the data access frequencies of the nodes do not change in our simulation. Thus, the communication cost is proportional to the size of the memory space.

5.2.5 Comparison with mobility-based allocation

We compare SRA that use the round-robin allocation method with the mobility-based replica allocation method, call it *M-SRA*. Figure 9a, b show the data accessibility and the com-

munication cost, respectively. For M-SRA, the parameter δ is set to 2.248 m/s which is the average speed of each node; the average speed was obtained through extensive experiments.

We expect that both the data accessibility and the communication cost of M-SRA would be higher than that of SRA because the mobility is one of the most important factors in a MANET. However, interestingly, the performance gap between SRA and M-SRA is marginal as we can see in Fig. 9a, b. It proves that grouping scheme is very important in the replica allocation in a MANET but the allocation of replicas in a group is not so important in this environment.

6 Conclusions

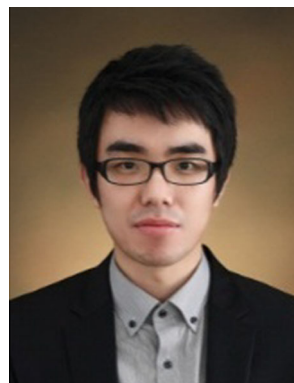
Although grouping in the existing replica allocation scheme is an effective way to achieve high data accessibility, it may lead to a very high communication cost. To resolve this problem, we propose the neighborhood-based replica allocation scheme with mobility-based grouping and role-based allocation according to estimated frequencies. The analytical evaluation and simulation results show that the proposed scheme is quite effective in terms of the communication cost, while achieving high data accessibility. Moreover, our proposed scheme is scalable because it maintains a similar performance in terms of communication cost as the number of nodes increases. We plan to analyze the proposed scheme and other replica allocation schemes in more practical environments considering the energy of devices along with various data item access patterns, more realistic mobility patterns and various data item sizes.

Acknowledgments This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0002851).

References

- Hara, T., & Madria, S. K. (2006). Data replication for improving data accessibility in ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(11), 1515–1532.
- Derhab, A., & Badache, N. (2009). Data replication protocols for mobile ad-hoc network. *IEEE Communications Surveys and Tutorials*, 11(2), 33–51.
- Choi, J.-H., Shim, K.-S., Lee, S., & Wu, K.-L. (2012). Handling selfishness in replica allocation over a mobile ad hoc network. *IEEE Transactions on Mobile Computing*, 11(2), 278–291.
- Moustafa, H., & Labiod, H. (2004). Multicast routing in mobile ad hoc networks. *Telecommunication Systems*, 25(1–2), 65–88.
- Badonnelradu, R., State, R., & Festic, O. (2005). Self-organizing monitoring in ad-hoc networks. *Telecommunication Systems*, 30(1–3), 143–160.

- Padmanabhan, P., Gruenwald, L., Vallur, A., & Atiquzzaman, M. (2008). A survey of data replication techniques for mobile ad hoc network databases. *The International Journal on Very Large Data Bases*, 17(5), 1143–1164.
- Xu, K., Hong, X., & Gerla, M. (2003). Landmark routing in ad hoc networks with mobile backbones. *Journal of Parallel and Distributed Computing*, 63(2), 110–122.
- Karumanchi, G., Muralidharan, G., & Prakash, R. (1999). Information dissemination in partitionable mobile ad hoc networks. In *Proceedings of International Symposium Reliable Distributed Systems* (pp. 4–13).
- Luo, J., Eugster, P.T., & Hubaux, J. P. (2003). Route driven gossip: Probabilistic reliable multicast in ad hoc networks. In *Proceedings of IEEE INFOCOM* (pp. 2229–2239).
- Chiang, C., Wu, H., Liu, W., & Gerla, M. (1997). Routing in clustered multihop, mobile wireless network with fading channel. In *Proceedings of IEEE Singapore International Conference on Networks* (pp. 197–211).
- Lin, C. R., & Gerla, M. (1997). Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communication*, 15(7), 1265–1275.
- Bao, L., & Garcia-Luna-Aceves, J. J. (2003). Topology management in ad hoc networks. In *Proceedings of ACM MobiCom* (pp. 129–140).
- Basagni, S., Chlamtac, I., Syrotiuk V.R., & Woodward, B.A. (1998). A distance routing effect algorithm for mobility (DREAM). In *Proceedings of MobiCom* (pp. 76–88).
- Karp, B., & Kung, H.T. (2000). Greedy perimeter stateless routing (GPSR) for wireless networks. In *Proceedings of MobiCom* (pp. 243–254).
- Ko, Y.B., & Vaidya, N.H. (1998). Location-aided routing(LAR) in mobile ad hoc network. In *Proceedings of MobiCom* (pp. 66–75).
- Kim, C., & Wu, M. (2011). Leader election on tree-based centrality in ad hoc networks. *Telecommunication Systems*, online published.
- The network simulator NS-3, <http://www.nsnam.org>.
- Broch, J., Maltz, D. A., Johnson, D. B., Cun Hu, Y., & Jetcheva, J. (1998). A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of ACM MobiCom* (pp. 85–97).
- Lee, K.-H. (2011). Authentication scheme based on cluster for mobile ad hoc game. *Journal of Future Game Technology*, 1(1), 35–42.



Sun-Kyum Kim received his M.S. in computer science from Yonsei University in Korea in 2012. He is currently a Ph.D. candidate at Yonsei University. His research interests include mobile ad-hoc networks, vehicular ad-hoc networks, and opportunistic networks.



Ji-Hyeun Yoon is currently an M.S. candidate in computer science at Yonsei University in Korea. His research interests include mobile ad-hoc networks, vehicular ad-hoc networks, and opportunistic networks.



Jae-Ho Choi received his Ph.D. and M.S. in computer science from Korea University in Korea in 2011 and 2005, respectively. He has been a postdoctoral researcher at Yonsei University since 2011. His research interests include mobile data management, mobile ad-hoc networks, opportunistic networks, and social network services.



Kwang-Jo Lee received his M.S. in computer science from Yonsei University in Korea in 2009. He has been a Ph.D. candidate at Yonsei University since 2009. His research interests include mobile ad-hoc networks, vehicular ad-hoc networks, and opportunistic networks.



Sung-Bong Yang received his M.S. and Ph.D. from the Dept. of Computer Science at the University of Oklahoma in 1986 and 1992, respectively. He has been a professor at Yonsei University since 1994. His research interests include graph algorithms, mobile computing, and social network services.