Neighborhood-Based Scalable Replica Allocation Scheme in Mobile Ad-hoc Networks

Sun-Kyum Kim, Ji-Hyeun Yoon, Kwang-Jo Lee, Jae-Ho Choi and Sung-Bong Yang

Department of Computer science, Yonsei University
Seoul, Korea
E-mail: {skyum, y_chick, kjlee5435, redcolor25, sbyang}@yonsei.ac.kr

*Corresponding author: Sung-Bong Yang

Abstract

In mobile ad-hoc networks, data accessibility is lower than that in traditional fixed networks due to frequent network divisions. Several schemes have been proposed to improve the data accessibility by replicating data items. These schemes utilize grouping to achieve high data accessibility. Although the grouping is an effective way to obtain high data accessibility, it may lead to huge amount of communication cost. In this paper, we propose a scalable replica allocation scheme to reduce the communication cost while maintaining high data accessibility. The proposed scheme builds groups of nodes in the network in such a way that each group has the star network topology. The central node in each group plays a role of a replica allocator who distributes replicas to the group nodes based on the estimated data access frequencies that are computed periodically. We evaluate the proposed scheme by simulations using the network simulator NS-3 as well as analytical evaluations. The results of simulations and analytical evaluations demonstrate that our proposed scheme reduces the communication cost significantly over the existing replica allocation schemes as the number of users increases in the network, while achieving comparable data accessibility.

Keywords: Replica allocation, MANET, Grouping, Data accessibility, Scalability

1. Introduction

With the advances in wireless communication technologies and the popularity of mobile devices, a mobile ad-hoc network (MANET) has attracted a lot of attentions [1][2][3]. A MANET is a self-organizing, rapidly deployable network which consists of wireless nodes without a fixed infrastructure. A large variety of MANET applications have been introduced [4]. 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 a MANET, network divisions do 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][2][3]. The proposed 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 schemes for replica allocation may lead to a great amount amount of communication cost. Consequently, the traditional schemes can be used only in mesoscale ad-hoc networks [1].

In this paper, we propose a scalable replica allocation scheme called the neighborhood-based replica allocation scheme (NRA) 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 to reduce the communication cost and to achieve high data accessibility as well. With this motivation in mind, we devise a grouping method in which each group has the star network topology. For grouping, NRA utilizes the mobility values of the nodes that are acquired during a replica relocation period.

For replica allocation, the central node in each group plays a role of the replica allocator who distributes replicas to the group nodes based on the estimated access frequencies that are computed by the allocator at the replica relocation time. Grouping with mobility and utilizing the estimated access frequency information enhance the stability of the topology of each group as well as the data accessibility.

The technical contributions of this paper can be summarized as follows.

- **Grouping method**: We devise a novel grouping for replica allocation with the communication among one-hop neighbor nodes. After grouping is done, the topology of the network is a set of star networks. The central node in a group has the lowest mobility among the group nodes.

- **Replica allocation scheme**: We propose an effective replica allocation based on the roles of the nodes and the estimated access frequencies. The central node of each group plays role of the replica allocator. The estimated data access frequencies are updated dynamically at each relocation time.

- **Analytical evaluations and the conducted simulations**: We verify the proposed scheme by analytical evaluations as well as the simulation using the network simulator NS-3. The results demonstrate that our 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 describes a brief overview of the related works. In Section 3, we show the employed system model. In Section 4, the proposed scheme is described in more detail and the analytical evaluations are provided. In Section 5, we give the simulation results of the proposed scheme. We conclude the paper in Section 6.

2. Related Work

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

In DCG, groups are constructed by finding biconnected components. Note that a biconnected component is a maximal biconnected subgraph in a graph. The network is considered as a graph in
which the nodes are the points and the communication links are the edges. Each biconnected component is a group. After grouping, the lowest suffix node—the node whose ID is lexicographically smallest—in a group is selected as the replica allocator of the group. Although DCG can achieve higher data accessibility than SAF, the communication cost for grouping is very high due to 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 [5]. The scheme targets for applications where inaccuracy is preferred to no information at all. Luo et al. also proposed a set of protocols that use a gossip-based multicast protocol [6]. The objective of this technique is to maximize the probability of accessing the up-to-date values of data items. Different from the aforementioned schemes, we focus on reducing the communication cost with high data accessibility for scalable replica allocation in a MANET.

In the aspect of data accessibility, grouping is a very important tool to achieve low communication cost. Recently, several grouping schemes were proposed. The least cluster change algorithm maintains groups in an event-driven way [7]. When a node moves out of the predefined range, the algorithm reconstructs groups. An adaptive clustering for mobile wireless network has also been proposed [8]. This algorithm reduces the communication cost for building groups, but it creates too many groups as time goes by. Consequently, almost every node forms a single-node group. The three-hops between adjacent cluster-heads scheme has also been introduced [9]. Although these schemes can make 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 bring down the communication cost as the number of nodes 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 like 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 also adopt the processing in this paper. Further details of our assumptions are listed below:

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

All data items are of an equal size. Each data item has a unique identifier. The data items are denoted by $D = \{D_1, D_2, D_3, \ldots, D_n\}$, where $n$ is the total number of data items.

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 feature are introduced in the previous researches [1][2]. Data items are reallocated periodically at a regular interval except for the static allocation scheme.

Each node moves freely within the maximum speed and is aware of its own speed (or velocity). To know the speed, GPS (or some other positioning systems) is equipped with each mobile node. Each node measures its mobility periodically. Node’s maximum speed varies from 0.1 m/s to 9 m/s.
4. Proposed scheme

4.1 An Overview

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

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

4.1 Grouping Method

In the proposed scheme, each node makes its neighborhood list at each relocation time. For the grouping algorithm we define a neighborhood list of a node is a set of tuple (node ID, mobility value). Each node \( N_i \) measures its mobility value \( m(t_s, t_i) \) between the previous relocation time \( t_s \) and the current relocation time \( t_i \) with the following equation:

\[
m_i(t_s, t_i) = \sqrt{(x_s - x_i)^2 + (y_s - y_i)^2} \quad (4.1)
\]

In Equation 4.1, \( x_s \) and \( y_s \) are the x- and y-coordinates of \( N_i \) at time \( t_s \), similarly \( x_i \) and \( y_i \) are the x- and y-coordinates at time \( t_i \). Hence the equation computes the mobility value of \( N_i \) by dividing the distance between the position at time \( t_s \) and the position at time \( t_i \) by the length of the relocation period.

At every relocation time, each node \( N_i \) sends \((ID_i, m(t_s, t_i))\) to every other node within one hop. Hence, each node will receive \((ID_j, m(t_s, t_i))\) from each node \( N_j \) who is within one hop distance. After each node has exchanged ID and mobility information, if a node \( N_i \) finds that it has the lowest mobility value, then \( N_i \) becomes the group allocator and tells each node \( N_j \) in its neighborhood list that \( N_j \) now becomes a group member. Algorithm 1 outlines how to construct groups for the proposed scheme.

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 \( \omega_i \), where \( \omega_i \) is the expected maximum time taken to exchange on round of send-receive message with neighbor nodes.

![Fig. 1. Grouping example of NRA](image-url)
Algorithm 1. Pseudo code for constructing groups

01: At relocation time \( t_r \) // the previous relocation time is \( t_c \).
02: **constructing_group()**
03: Each node \( N_i \) sends \((ID_i, m(t_c, t_r))\) to each of its neighbor nodes;
04: **while** (during the predefined time \( \omega \))
05: **if** \( N_i \) receives the mobility value \( m_j(t_c, t_r) \) from \( N_j \); // \( N_j \) is a neighbor node of \( N_i \).
06: **if** \( m(i, t_c) > m_j(t_c, t_r) \) 
07: \( N_i \) becomes a member node of a group;
08: **break**;
07: **else** \( N_i \) becomes the allocator;
08: **else** \( N_i \) becomes the allocator;
09: }
10: If \( (N_i \) is the allocator) \( N_i \) sends join message to each of its neighbor nodes;
11: **else** \( N_i \) receives join message for its neighbor node; // \( N_i \) will become a member node.
12: }

Fig. 1 shows an example of the 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. Assume that \( N_1 \) and \( N_5 \) have the lowest mobility values among their own neighbor nodes. Then \( N_1 \) 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 who receive join messages become group members.

4.3 Replica Allocation Scheme

After grouping is done, each group has the star network topology. We now let the central node in each group play a role of the replica allocator. Since we use the mobility value as a criterion for grouping, an allocator could be considered as a more ‘reliable’ node than other nodes in the group. Note that if the criterion is changed into some other factors such as energy (battery power) and memory size, an allocator can be a more ‘powerful’ or ‘huge’ node (i.e., having high power or larger memory space). Since the allocators are reliable nodes in the proposed scheme, the allocators with lower mobility contribute to achieving high data accessibility. The philosophy behind distributing replicas within a group is that important (i.e., more frequently accessed) data items should be held by the important (i.e., reliable) node in a group. It is natural that the important data items are the data items accessed more frequently and the important nodes are the allocators.

For the proposed allocation scheme, 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 other nodes and then counts the number of accesses to each data item as well as the total number of queries issued during a period between two consecutive relocation times. \( N_i \) now can construct its own estimated data access frequency table called \( esTable \), like Table 1, each of whose entries 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:

\[
the \ estimated \ access \ frequency \ of \ D_j = \frac{\text{total number of accesses to } D_j}{\text{total number of queries issued}} \quad (4.2)
\]

Note that it does not matter whether \( N_i \) has the requested data item in its memory or not and that \( N_i \) can observe all the messages passing by so that \( N_i \) can acknowledge whether a request for a data item has been satisfied or not. These estimated access frequencies to the data items provide more ‘up-to-date’ access information—without any assumption on the access frequencies—and do not require additional communication cost at all. The estimated frequencies of all the data items at each node are initialized to one over the total number of data items, that is, \( 1/n \).

Since \( esTable \) is constructed by each node individually and independently, the table in each node may possess different contents. Observe that a request query to a data item may go through only a few nodes.
while many other nodes in the network do not involve with processing the query. Algorithm 2 describes how replicas are allocated at each relocation time by the proposed replica allocation scheme. In the algorithm, $esTable_i$ and $M_i$ mean the $esTable$ and the size of memory space of node $N_i$, respectively. Table 1 shows the contents of $esTables$ of the allocator nodes as examples; the values in the table entries are obtained for a specific case and hence we do not need to know how these values are computed.

Figure 2 illustrates the results of the data item allocation to the nodes in groups $A$ and $B$. In this example, the memory space of each node can hold up to three data items and $N_i$ and $N_i$ are the allocators of groups $A$ and $B$, respectively. Based on $esTable_i$ in Table 1, $N_i$ keeps $D_1$, $D_3$, and $D_5$. Then $N_i$ allocates $D_4$ to $N_2$, $D_7$ to $N_3$, $D_{10}$ to $N_5$, $D_9$ to $N_3$, $D_3$ to $N_2$, finally $D_6$ to $N_4$. For group $B$, $N_i$ keeps $D_2$, $D_{10}$, and $D_6$. $N_i$ has $D_4$, $D_8$, and $D_5$, and $N_6$ has $D_2$, $D_1$, and $D_7$.

**Algorithm 2. Pseudo code of allocating replicas**

```java
01: At each relocation time
02: /*Allocator $N_i$ allocates replicas to the group nodes based on $esTable_i$*/
03: allocating_replica (){
04:     $N_i$ sorts the data items in $esTable_i$ in descending order of data access frequency;
05:     Let the sorted data items be stored in array $A$;
07:     $N_i$ allocates the next data item with the highest estimated frequency from the array to each node in the group in the round robin fashion until the memory spaces of all the group members are full;
08: }
```

**Table 1. Examples of $esTables$: $esTable_1$ and $esTable_5$**

<table>
<thead>
<tr>
<th>Data items</th>
<th>Estimated Access Frequencies</th>
<th>Data items</th>
<th>Estimated Access Frequencies</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N_f$</td>
<td>$N_i$</td>
<td>$N_f$</td>
</tr>
<tr>
<td>$D_1$</td>
<td>0.38</td>
<td>0.00</td>
<td>$D_6$</td>
</tr>
<tr>
<td>$D_2$</td>
<td>0.16</td>
<td>0.07</td>
<td>$D_7$</td>
</tr>
<tr>
<td>$D_3$</td>
<td>0.24</td>
<td>0.00</td>
<td>$D_8$</td>
</tr>
<tr>
<td>$D_4$</td>
<td>0.12</td>
<td>0.11</td>
<td>$D_9$</td>
</tr>
<tr>
<td>$D_5$</td>
<td>0.00</td>
<td>0.40</td>
<td>$D_{10}$</td>
</tr>
</tbody>
</table>

Fig. 2. Replica allocation by NRA based on Table 1
4.4 Analytical Evaluation

We analyze the communication costs of our replica allocation scheme NRA, SAF, and DCG in this section. We formulate the communication cost for replica allocation as follows.

$$C_t = C_g + C_r + C_a \quad (4.3)$$

In Equation 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 allocation, and the cost for accessing the replicas. 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 \(C_g-NRA\) of our scheme can be estimated:

$$C_{g-NRA} \leq \sum_{c \in CC} 3E_c \quad (4.4)$$

In Equation 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 NRA is at most 3 times the number of edges in \(c\), since each edge (link) is used twice for sending the IDs and mobility values each other and once for notification of group membership by each central node.

Next, we formulate \(C_r-NRA\) of our scheme below;

$$C_{r-NRA} \leq \sum_{g \in GG} N_g \quad (4.5)$$

In the above equation, \(GG\) is a set of all groups constructed by NRA 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, \(C_{r-NRA}\) is at most \(m\). Last, we formulate \(C_a-NRA\) of our scheme in the following;

$$C_{a-NRA} \leq \sum_{c \in CC} N_c (2E_c + ph) \quad (4.6)$$

In Equation 4.6, \(p\) is the probability that each node has the data item requested by others, and \(h\) is the average number of hops from the requesting node to the node having the data item requested. Hence, \((2E_c + ph)\) is the upper bound on the expected number of hops to access a data item by a node, whether the data item has been accessed successfully or not. Then we multiply the number of nodes in a connected component \(c\) to the term for the accessing costs of all the nodes in \(c\). For \(C_{a-NRA}\), we sum up the costs for all the connected components. Now \(C_{r-NRA}\) can be written as follows:

$$C_{r-NRA} \leq \sum_{c \in CC} 3E_c + m + \sum_{c \in CC} N_c (2E_c + ph) \quad (4.7)$$

We also formulate communication costs of SAF and DCG similarly. \(C_{r-SAF}\) and \(C_{r-DCG}\) can be formulated below.

$$C_{r-SAF} \leq \sum_{c \in ECC} N_c (2E_c + ph) \quad (4.8)$$

$$C_{r-DCG} \leq \sum_{c \in ECC} (E_c + 2E_{d_{max}} Diam_c) + \sum_{g \in GG} N_g E_g + \sum_{c \in CC} N_c (2E_c + ph) \quad (4.9)$$

\(C_{r-SAF}\) is equal to \(C_{a-SAF}\), since there are no grouping and no replica reallocation in SAF. In DCG, the grouping cost \(C_{g-DCG}\) is much more expensive than \(C_{g-NRA}\), 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 the information it can get. The term \(\sum_{c \in CC} (E_c + 2E_{d_{max}} Diam_c)\) is the upper bound on the number of hops for broadcasting in DCG. For each connected component \(c\), first, each node sends its ID to its neighbor nodes. This entails \(E_c\);
is, each edge in $c$ is used at 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 for this case. Since all the nodes in $c$ should receive the message sent by each node, the upper bound of the overall cost is $2E_c d_{\text{max}} Diam_c$, where $d_{\text{max}}$ is the maximum degree of the nodes in $c$ and $Diam_c$ 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 the total cost $C_{g-DCG}$ for grouping is $O(m^2)$. But $C_{g-NRA}$ is only at most $O(E_c) = O(m^2)$.

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

### Table 2. Parameters used in the simulation

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>40~70</td>
<td>Size of memory space</td>
<td>10 data</td>
</tr>
<tr>
<td>Number of data</td>
<td>40~70</td>
<td>Speed of nodes</td>
<td>0.1 ~ 9.0</td>
</tr>
<tr>
<td>Communication</td>
<td>63 m</td>
<td>Query processing</td>
<td>10 sec</td>
</tr>
<tr>
<td>Network area</td>
<td>450x450</td>
<td>Relocation period</td>
<td>64 ~ 1,024</td>
</tr>
</tbody>
</table>

## 5. Simulation Results

### 5.1 Simulation Environments

For the experiments, we use the network simulator NS-3 v3.10 [11]. The system parameters are determined based on the parameters that are adopted in [1]. The number of nodes varies from 40 to 70. Each node has the memory space that can hold up to 10 data items. Network area is set to 450m × 450m and the communication range is 63 m. The movement of a node follows the random way point model [12] which is the most frequently used moving pattern in the MANET simulations. The maximum speed of a node varies from 0.1 to 9.0 m/sec. In the simulator, a node issues a query every 10 sec. The replicas are relocated at specific relocation time periodically [1]. The relocation period varies from 64 to 1,024 sec. 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 ID in [1]. During 6,000 seconds, we simulate and compare the proposed scheme with SAF and DCG. Table 2 summarizes the parameters of our simulation environment. We evaluate these schemes using the following two performance metrics:

1) Data accessibility: This is the ratio of the number of successful queries to the total number of issued queries.
2) Communication cost: This is the total hop count of data transmission for groupings, replica allocations and accessing data items.

### 5.2.1 Effect of the relocation period

First, we examine the effects of the relocation period on each of three schemes (i.e., NRA, DCG, and SAF). Fig. 3(a) and 3(b) show the simulation results. Since SAF does not need any grouping or reallocation, each node holds many duplicated data items. As a result, SAF shows the lowest data accessibility. In Fig. 3(a), NRA shows a similar performance to DCG. Although NRA does not build groups based on a costly grouping scheme, NRA allocates replicas based on the role-based allocation scheme with the estimated data access frequencies to achieve comparable success ratios to DCG. The
results in Fig. 3(a) confirm the effectiveness of NRA in such a way that replica allocation based on the estimated access frequencies provides high accessibility but how well the grouping is made is not so important in this environment.

Next, we examine the communication cost with varying relocation period. In each of DCG and NRA schemes, the communication cost decreases as the relocation period increases because these schemes perform relocations less and less frequently. Fig. 3(b) compares the communication costs of the schemes. We expected that the communication cost of NRA is slightly higher than that of SAF and significantly lower than that of DCG. As we can see in Fig. 3(b), the communication cost gap between NRA and DCG becomes narrower tremendously as the relocation period reduces. Such phenomenon proves that DCG requires too much communications in grouping, compared with NRA.

![Fig. 3. Effect of relocation period](image1.png)

![Fig. 4. Effect of increasing the number of nodes](image2.png)

### 5.2.2 Effect of the increasing the number of nodes

We evaluate the scalability of the schemes as the number of nodes in the network increases. In this simulation, the relocation period is fixed at 1,024 sec. Our intuition is that when the number of nodes increases the data accessibility also increases. However, interestingly, as can be seen Fig. 4(a), the data accessibility of all the schemes does not seem to increase or decrease noticeably. This is because the network simulator NS-3 models the network link problems (e.g., fading, multipath effects, congestion, interference, and so on). When the number of nodes increases such problems do occur more frequently. Consequently, a large amount of requests are failed.

Fig. 4(b) 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. Interestingly, the communication cost of NRA seems to increase minimally. This is because most communications for messages and data items are processed within one hop in NRA. Such results demonstrate that NRA supports scalability well and can be deployed in a real world environment.

5. Conclusions

Although grouping in the existing replica allocation scheme is an effective way to achieve high data accessibility, it may lead to a very large amount of communication cost. To resolve such a problem, we proposed the neighborhood-based replica allocation scheme with mobility-based grouping and role-based allocation along with the estimated frequencies. The analytical evaluation and simulation results show that the proposed scheme is proven to be quite effective in terms of the communication cost, while achieving high data accessibility. Moreover, our proposed scheme is scalable because our scheme maintains similar performances as the number of nodes increases in terms of the communication cost. We plan to analyze the proposed scheme and other replica allocation schemes in more practical environments and devise more practical allocation schemes.

References