Computing and Informatics, Vol. 23, 2004, 133–156

A DYNAMIC ADAPTIVE REPLICA ALLOCATION ALGORITHM IN MOBILE AD HOC NETWORKS

Zheng Jing, Lu Xieng, Yang Kan, Wang Yije

School of Computer Science
National University of Defense Technology
Changsha 410073, Hunan, China
e-mail: zhengjing621@hotmail.com

Manuscript received 24 August 2004
Communicated by Ladislav Hluchý

Abstract. In mobile ad hoc networks (MANET), nodes move freely and the distribution of access requests changes dynamically. Replica allocation in such a dynamic environment is a significant challenge. The communication cost has become a prominent factor influencing the performance of replica allocation in the MANET environment. In this paper, a dynamic adaptive replica allocation algorithm that can adapt to the nodes motion is proposed to minimize the communication cost of the object access. When changes occur in the access requests of the object or the network topology, each replica node collects access requests from its neighbors and makes decisions locally to expand the replica to neighbors or to relinquish the replica. This algorithm dynamically adjusts the replica allocation scheme towards a local optimal one. To reduce the oscillation of replica allocation, a statistical method based on history information is utilized to choose stable neighbors and to expand the replica to relatively stable nodes. Simulation results show that our algorithms efficiently reduce the communication cost of object access in MANET environment.

Keywords: MANET, replica allocation, read-write pattern, stable neighbor

1 INTRODUCTION

Manet (Mobile Ad hoc Network) is a collection of wireless autonomous mobile nodes without any fixed backbone infrastructure, in which nodes are free to move.

MANET can be used in many situations where temporary network connectivity is required, for example in battlefields and in the disaster recovery. Such a dynamic environment brings about significant challenges to the replica allocation mechanism, which is one of the key technologies to improve accessibility, reliability and performance of the system. The replica allocation algorithm proposed in this paper addresses the issue of the performance of the data access in the MANET environment.

Replica allocation for performance improvement in the field of fixed networks has been an extensive research topic. In many researches, the communication cost is used as cost function. However, because these researches are for fixed networks, they do not consider the effect on the data replication caused by the nodes mobility. In [1], a minimum- spanning- tree (MST) write policy is introduced. However, this cost model is not suitable for the MANET environment because the communication cost and the algorithm complexity of building a spanning tree are very high in MANET. In [2], nodes forward read requests to the nearest replica node and write requests to all replica nodes along the shortest path. However, this scheme requires that every node should maintain information of all replica nodes. When a replica node changes, every node must be notified. Thus it is not suitable for the mobile environment as well.

Several strategies [3, 4, 5, 6] for replicating or caching data have been proposed in traditional wireless mobile networks. Most of these strategies assume that mobile nodes access the database at sites in a fixed network, and replicate or cache data on mobile nodes. These data replication strategies emphasize reducing the one hop wireless communication cost induced by keeping consistency between the data in a base station and their replicas in mobile nodes. However, these strategies are proposed for the traditional one-hop wireless mobile network and completely different from our approach which is designed for the multi-hop MANET network without base stations.

Only a few replica allocation algorithms have been proposed for the MANET environment recently. In [7, 8], several global greedy replica allocation methods have been put forward. However, much information needs to be exchanged among nodes by these methods, especially when the topology of network changes rapidly. In [9], an algorithm is proposed to predict the network partitioning and to determine the time and the location for replica allocating to ensure the service availability. It is known that all these algorithms [7, 8, 9] only focus on improving the data accessibility during the network partitioning.

In this paper, a distributed dynamic adaptive replica allocation algorithm is proposed for the MANET environment. The communication cost is used as the cost function in the algorithm because the communication cost becomes the most important factor which influences the performance of data access in this environment. Our algorithm can dynamically adjust the replica allocation scheme towards a local optimal one according to the access requests distribution and topology changes. The concept of "stable neighbor" is proposed in our algorithm and the access requests are collected only from stable neighbors while replica nodes expanding or relinquish-

ing the replica. Thereby the replicas are stored on relatively stable nodes and the oscillation of replica allocation is reduced while nodes move rapidly.

The rest of the paper is organized as follows: in Section 2 the cost model is defined; in Section 3 a new distributed dynamic adaptive replica allocation algorithm is presented in detail; in Section 4 the simulation results are given; and finally in Section 5, the summary and some future work are presented.

2 THE DESCRIPTION OF THE PROBLEM

2.1 The Cost Model

The cost model is described as follows:

1. In our research, hops are used as the metric of the communication cost of data access. In the MANET environment, the communication cost between two nodes includes the wireless bandwidth cost, energy consumption, the delay of the communication and so on. All these factors are related to hops, for example, increase of hops may linearly increase latency due to the packet delay at each hop; the communication is the most important factor to affect battery life, and the more hops there are between the source and destination of data access the more energy is consumed due to the packet relay at each hop; the throughput of TCP attenuates exponentially with the increase of hops. So we use the hops between two nodes to measure the communication costs between these two nodes.
2. The ROWA (READ-ONE-WRITE-ALL) policy is used to ensure the consistency of the replicas.
3. Each individual access is independent.
4. The access request for the object is sent to the closest replica in the network. The read request is served by the closest replica node, but the write request is propagated from the closest replica to all other replicas along the shortest path. Therefore the information of replication allocation just needs to be maintained on the replica nodes. Being compared to [2], this method decreases the cost to maintain the information of replica set on the non-replica nodes; Being compared to [1], it avoids the algorithm complexity caused by dynamically building the spanning tree in the MANET environment.

Definition 1. The replica allocation scheme of an object O, denoted by F, is the set of nodes at which O is replicated.

The set of mobile nodes is denoted by V. For $i, j \in V$, $d(i, j)$ is the least hops between i and j. Thus the cost of a single read request by node i is $d(i, F) = \min_{j \in F} d(i, j)$. The cost of a single write request by node i is $d(i, F) + \sum_{k \in F} d(j, k)$, where j is the node which satisfies $d(i, j) = d(i, F)$. Therefore during the interval t, the total communication cost of F, denoted by $\text{cost}(F)$, can be computed as follows:

$$\begin{aligned}\mathrm{cost}(F) &= \sum_{i\in V} W(i)d(i,F) + \sum_{s\in F}\sum_{j\in F} W_{re}(s)d(s,j) + \sum_{i\in V} R(i)d(i,F)\\ &= \mathrm{cost}W_{\mathrm{forward}}(F) + \mathrm{cost}W_{\mathrm{up}}(F) + \mathrm{cost}R(F) \qquad (1)\end{aligned}$$

In this equation, $W(i)$, $W_{re}(i)$ and $R(i)$ are statistical values acquired during the interval t. $R(i)$ and $W(i)$ are the numbers of the read and write requests to O issued by i, $W_{re}(i)$ is the total number of the write requests to O that i receives from itself or its non-replica neighbors. $\mathrm{cost}W_{\mathrm{forward}}(F) = \sum_{i\in V} W(i)d(i,F)$ is the cost of forwarding write requests to replica nodes; $\mathrm{cost}W_{\mathrm{forward}}(F) = \sum_{s\in F}\sum_{j\in F} W_{re}(s)d(s,j)$ represents the cost of propagating write requests among replica nodes; $\mathrm{cost}W_{\mathrm{up}}(F) = \sum_{i\in V} R(i)d(i,F)$ refers to the total access cost of read requests.

2.2 The Replica Allocation Problem

Definition 2. The read-write pattern for an object O is the number of reads and writes to O issued by each node.

While all the nodes access the replicated object with the same read-write pattern, let the total number of accesses to the object by every node in some fixed interval be τ and write ratio be θ:

$$\theta = \frac{\text{number of write requests}}{\text{number of write requests} + \text{number of read requests}}$$

Without loss of generality we may assume that the common value of τ is 1, and for some given u the cost function (1) then becomes

$$\begin{aligned}\mathrm{cost}(R,\theta) &= \theta\sum_{i\in V} d(i,F) + \mathrm{cost}W_{\mathrm{up}}(F,\theta) + (1-\theta)\sum_{i\in V} d(i,F)\\ &= \sum_{s\in F}\sum_{j\in F} W_{re}(s)d(s,j) + \sum_{i\in V} d(i,F) = A(F) + B(F) \qquad (2)\end{aligned}$$

where $A(F) = \mathrm{cost}W_{\mathrm{up}}(F,\theta) = \sum_{s\in F}\sum_{j\in F} W_{re}(s)d(s,j)$, $B(F) = \sum_{i\in V} d(i,F)$.

If $\mathrm{cost}(F^*,\theta) = \min_{F\subseteq V}\mathrm{cost}(F,\theta)$, $0 \le \theta \le 1$, then define a set F^* is *the optimal replication allocation scheme for* θ.

Definition 3. The *REPLICA_LOCATION* problem is for a θ and a given integer k to find a replication allocation scheme F that satisfies $\min_{F\subseteq V}\mathrm{cost}(F,\theta) \le k$ in a given graph $G = (V,E)$. As a language, we define REPLICA_LOCATION = $\{<G,\theta,k>$: graph G has the replication allocation scheme with the lowest access communication cost no more than $k\}$.

For general static networks, the problem of finding an optimal replica allocation scheme (i.e., a scheme that has the minimum cost for a given read-write pattern) has been proved to be NP-complete for different cost models [10][2]. As for the cost model defined by (1), this problem is also proved to be NP-complete as below.

Theorem 1. The REPLICA_ALLOCATION problem is NP-complete. The details of the proof can be found in Appendix A.

As for the MANET environment, it is more difficult to find the optimal replica allocation. Thus a distributed asynchronous adaptive replica allocation algorithm is proposed to find the near-optimal replica allocation scheme.

3 ADAPTIVE REPLICA ALLOCATION ALGORITHMS

3.1 The ARAM Algorithm

In the fixed networks, the optimal replica allocation scheme of an object depends on the read-write pattern, but in the MANET environment it depends not only on the read-write pattern but also on the nodes motion. In the ARAM (*the Adaptive Replica Allocation Algorithm In MANET*) algorithm, each replica node collects access requests from its neighbors and makes decisions locally to update the replica allocation scheme. Thus the ARAM algorithm adapts to the dynamic MANET environment. Furthermore, it can dynamically adjust the replica allocation scheme towards a local (rather than global) optimum.

In the MANET environment, our algorithm is executed at each replica node periodically and independently. The duration of the period t is a uniform system parameter. It depends on the nature of the network, particularly, how dynamic the network topology is. The period tends to be shorter for a network with more frequent topological changes and read-write pattern changes.

Before the ARAM algorithm is introduced in detail, some variables are denoted. For each replica node $i \in F$, $q(i) = \sum_{j \in F} d(i,j)$ is denoted as the weight value of i, and for each non-replica node j, $C(j) = \{i \mid i \in F, d(j,i) = d(j,F)\}$ is denoted as the *access set of* j. In the ARAM, if there are multiple shortest paths from non-replica node j to F, select the replica node in $C(j)$ with least weight to access F, i.e. for each non-replica node j, if $|C(j)| \geq 1$ and $q(p_j) = \min q(C(j))$ $(p_j \in C(j))$, then j accesses F through p_j.

The ARAM algorithm is executed on each replica node at the end of each interval t and is shown in Table 1.

The Expansion_test, Relinquishment_test, and Switch_test operation in the ARAM algorithm will be discussed as follows.

Expansion_test. For the neighbor u of s and $u \notin F$, if the replica is expanded to u, one hop will be decreased for some nodes to access the replica, but the cost will increase for propagating write requests to the new replica node u. If the following inequation (3) is true (which means that when a replica is expanded to u, the decrease of the access cost is greater than the increase of the update cost, thus the total communication cost decreases. This conclusion will be shown in Theorem 1:

execute *Expansion_test* for each non-replica neighbor u of s
if exist node u^*, u^* satisfies the condition of expansion
  and $\Delta\text{cost}(u^*) = \max \Delta\text{cost}(u)$ then
  expand replica to u^*, $F = F \cup \{u^*\}$
  return 1
endif
execute *Switch_test* for each non-replica neighbor u of s
if exist node u^*, u^* satisfies the condition of switch
  and $\Delta\text{cost}(u^*) = \max \text{cost}(u)$ then
  switch replica from s to u^*, $F = F - \{s\} + \{u^*\}$
  return 2
endif
execute *Relinquishment_test* on s
if the condition of the relinquishment is satisfied then
  relinquish s, $F = F - \{s\}$
  return 3
endif
return 0

Table 1. ARAM algorithm

$$R_{\text{from}}(u^*) + R_{\text{change}} > (|F| - 2)(W_{\text{from}}(u^*) + W_{\text{change}}) + \sum_{i \in F}(W_{re}(i) - W_{\text{change}}(i))d(i, s) + W + \Delta\text{change} \quad (3)$$

and

$$\Delta\text{cost}(u^*) = (R_{\text{from}}(u^*) + R_{\text{change}}) - \Big((|F| - 1)(W_{\text{from}}(u^*) + W_{\text{change}}) + \sum_{i \in F}(W_{re}(i) - W_{\text{change}} + \sum_{i \in F}(W_{re}(i) - W_{\text{change}}(i))d(i, s) + W + \Delta\text{change}\Big) = \max \text{cost}(u)$$

Then a replica is expanded to u^* and $F' = F \cup \{u^*\}$.

As shown in Figure 1 (a), set $A = \{i \mid i \in V - F,\ s \in C(i)$ and u is at a site on one of the shortest paths between i and s, but u is not at a site on the access path from i to $F\}$. The nodes in set A will change the access path and access u in the next interval if a replica is expanded to u; $\Delta\text{change} = \sum_{i \in A} W(i)(q(s) - q(p_i))$, represents the change of write cost caused by the changes of the access path of A; and during the last interval, R_{change} and W_{change} are the total read and write requests from A, respectively; W is the total write requests from V; $W_{\text{change}}(i)$

is the write requests received by replica i from A; $R_{\text{from}}(u)$ and $W_{\text{from}}(u)$ are read and write requests received by s from u, respectively.

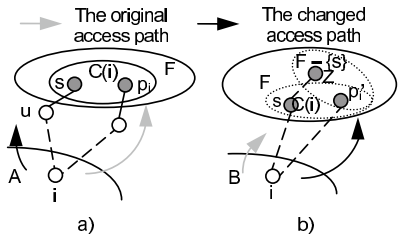


Fig. 1. Changes of access path caused by the algorithm. (a) The expansion test is executed on replica s; (b) The relinquishment test is executed on replica s.

Relinquishment_test. As for the replica node s, if all expansion and switch tests fail, the ARAM algorithm executes the *relinquishment test*. If the number of update requests received by s from other replicas is larger than that of the read and write requests received by s from itself and those non-replica nodes, then s requires to relinquish the replica. The set $B = \{i \mid i \in V - F, s = p_i, C(i) - \{s\} = \Phi\}$ is defined in Figure 1 (b). Even if s relinquishes the replica, the number of hops from the nodes of B to F will not change (but the access paths of the nodes of B will). If the following inequation is true, s relinquishes the replica.

$$(R_B + W_B)d(s,z) + \sum_{i \in F} W_{re}(i)d(i,s) > R_{re}(s) + W_{re}(s))d(s,z)$$

$$+ W_{re}(s)(q(w) - q(s) - d(s,w)) + W_B(q(p_i') - q(w) - d(s,p_i') + d(s,w)) \quad (4)$$

In this inequation, R_B and W_B are the read and write requests from B, respectively; $R_{re}(s)$ is the read requests received by s from itself and other nodes; p_i', z and w are all replica nodes. p_i' satisfies the following condition: for $i \in B$, $p_i' \in C(i)$, and $q(p_i') = \min q(j)$, $j \in C(i) - \{s\}$; z satisfies the following condition: If $D = \{i \mid i \in F, d(s, F - \{s\}) = d(s,i)\}$, $\exists z \in D$, $q(z) = \min q(D)$; and w satisfies the following condition: $w \in F - \{s\}$, $q(w) - d(w,s) = \max(q(i) - d(i,s))$, $i \in F - \{s\}$.

Switch_test. For the replica node s, if all expansion tests fail, then s executes the switch test. The switch test will allocate the replica to the neighbor node which receives more read and write requests. For each neighbor u of s and $u \in F$, if the following inequation is true,

$$R_{\text{from}}(u^*) + W_{\text{from}}(u^*) > \frac{1}{2}(R_{re}(s) + W_{re}(s) + \sum_{j \in F - \{s\}} W_{re}(j)\Delta Q) \quad (5)$$

and

$$\Delta\text{cost}(u^*) = (R_{\text{from}}(u^*) + W_{\text{from}}(u^*))$$

$$-\left(\frac{1}{2}(R_{re}(s)+W_{re}(s)+\sum_{j\in F-\{s\}}W_{re}(j)\Delta Q)\right)=\max\Delta\text{cost}(u)$$

then the replica is moved from s to u^*, and u^* becomes a replica node while s is not a replica node any more.

In the above inequation, $\Delta Q=\max(1,q(s)+|F|-1-q(j))$, $j\in F-\{s\}$.

Theorem 2. For a static network, suppose that the read-write pattern does not change and the ARAM algorithm is executed at the ends of every interval. Then the communication cost of object access will decrease once any operation of Expansion_test, Relinquishment_test, and Switch_test succeeds until the replica allocation scheme reaches a local optimal one. The details of the proof are manifested in Appendix B.

In the MANET environment, the network topology changes with the motion of nodes. It is true, however, that in many cases mobile nodes have similar mobility behavior (e.g. nodes are in the same group) or nodes move at a low speed. In these cases, the network topology changes slowly and is relatively stable. If the interval between two executions of the algorithm is shorter than the interval of the changes in the network topology, the historical information acquired by replica nodes can predict the distribution of the read and write requests and a execution of the algorithm will enable the communication cost to decrease. Therefore the replica allocation schemes image changes of the network topology.

Example 1. In this example, the operation of the ARAM algorithm is demonstrated. Suppose that the network of Figure 2 is static and the read-write pattern is fixed. In each interval t, 20 read and 2 write requests are issued by node 4; 60 read and 5 write requests by node 6; 30 read and 5 write requests by node 8; 4 read and 1 write requests by each node of the rest. Suppose further that requests are serviced in the time period in which they are issued and the access path of 6 is 6-3-1 initially.

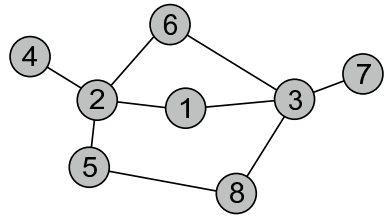


Fig. 2. The network topology

Suppose that the initial replica allocation scheme $F_1=\{1\}$, then $\text{cost}(F_1)=274$. At the end of the first interval, execute the ARAM algorithm, and 2 satisfies the expansion condition, thus $F_2=\{1,2\}$, $\text{cost}(F_2)=193$. The access path of 6 changes to 6-2. At the end of the second interval, 1 expands replica to 3 and 2 expands

replica to 6, then $F_3 = \{1, 2, 3, 6\}$, $\text{cost}(F_3) = 132$. At the end of the third interval, 1 relinquishes the replica, thus $F_4 = \{2, 3, 6\}$, $\text{cost}(F_3) = 118$. The access path of 1 is 1-3. Starting from the fourth interval, the replica allocation scheme will stabilize at $\{2, 3, 6\}$, and it will not change further. According to the assumption mentioned above, the optimal replica allocation scheme is $\{2, 6, 8\}$ or $\{6, 8\}$, $\text{cost}(F^*) = 103$. In the ARAM algorithm, the condition (3) is too strict to expand 8, therefore the relinquishment test on 3 cannot be executed.

3.2 The EARAM Algorithm

In the ARAM algorithm, information of all access paths is collected, which makes the algorithm complex and much state information need to be maintained. Meanwhile, in the ARAM the replica expansion and relinquishment operation are executed only when the strict conditions are satisfied. Such a policy can ensure the communication cost decreasing each time, but the chances to achieve a more optimal result are lost. Therefore, an improved algorithm based on ARAM-EARAM is proposed. The EARAM (The Enhanced ARAM Algorithm) algorithm ignores the changes of the access path caused by the changes in replica allocation scheme, i.e., it neglects the effect of set A when the replica is expanded and the effect of set B when the replica is relinquished. Thus, the replica expansion condition (3) can be simplified as follows: for neighbor u of s ($s \in F$).

If s is at a site on all the shortest paths between u and every node of F, the replica expansion condition is the strictest, as shown in the inequality (6).

$$R_{\text{from}}(u) > (|F| - 2)W_{\text{from}}(u) + \sum_{i \in F} W_{re}(i)d(i, s) + W \tag{6}$$

If u is at a site on all the shortest paths between s and every node of F, the replica expansion condition is the most relaxed, as shown in the inequality (7).

$$R_{\text{from}}(u) > W_{re}(s) - |F|W_{\text{from}}(u) + \sum_{i \in F} W_{re}(i)d(i, s) - W \tag{7}$$

The condition (6) is so strict that it may hamper some proper replica expansions; the condition (7) is much more relaxed but may execute some mistaken replica expansions. Therefore a tradeoff between conditions (6) and (7) is made, and the inequality (8) is acquired.

$$R_{\text{from}}(u) > W_{re}(s) - |F|W_{\text{from}}(u) + \sum_{i \in F} W_{re}(i)d(i, s) \tag{8}$$

Similarly, the relinquishment condition (5) can be simplified as follows:

$$\sum_{i \in F} W_{re}(i)d(i, s) > (R_{re}(s) + W_{re}(s))d(s, z) + W_{re}(s)(q(w) - q(s) - d(s, w)) \tag{9}$$

Similarly, a tradeoff switch condition is achieved as follows:

$$R_{\text{from}}(u) + W_{\text{from}}(u) > \frac{1}{2}(R_{re}(s) + W_{re}(s)) \tag{10}$$

$$\Delta\text{cost}(u) = (R_{\text{from}}(u) + W_{\text{from}}(u)) - \frac{1}{2}(R_{re}(s) + W_{re}(s))$$

The EARAM algorithm is the same as the ARAM algorithm except that the condition (3), (4) and (5) in the ARAM algorithm are replaced by the condition (8), (9) and (10), respectively. The information collected by the EARAM algorithm is not sufficient to ensure that the communication cost of data access decreases once the EARAM algorithm is executed, and replica may be mistakenly expanded or relinquished. However, the mistaken operations may be corrected by the operations later and the total communication cost tends to decrrease.

Execute the EARAM algorithm with the assumption given in Example 1. The changes in replica allocation schemes are: $F_1 = \{1\}$ and $\text{cost}(F_1) = 274$; 1 expands replica to 3, then $F_2 = \{1, 3\}$, and $\text{cost}(F_2) = 181$; 1 expands replica to 2 and 3 expands replica to 6, then $F_3 = \{1, 2, 3, 6\}$, and $\text{cost}(F_3) = 135$; switch replica from 2 to 4, and 1 relinquishes replica while 3 expands replica to 8, therefore $F_4 = \{4, 3, 6, 8\}$ and $\text{cost}(F_4) = 114$; 3 and 4 relinquish replica, thus $F_5 = \{6, 8\}$ and $\text{cost}(F_3) = 103$; 6 expands replica to 2, then $F_6 = \{2, 6, 8\}$ and $\text{cost}(F_6) = 103$.

3.3 The EARAM_SN Algorithm

In the MANET environment, the changes of network topology caused by nodes motion may cause the replica allocation scheme to be oscillated. The main idea of the EARAM_SN(The EARAM Algorithm Based On The Stable Neighbor) algorithm is to find the relatively stable neighbors of replica nodes in a distributed way and to expand replicas only to stable neighbors. Also in the EARAM_SN algorithm the access requests are collected only from stable neighbors while expanding or relinquishing the replica. The algorithm enables the replicas to be stored on relatively stable nodes, thus the oscillation of replica allocation is reduced while nodes move rapidly.

Now the details of this algorithm are discussed. The distance between two neighbors is used to measure their neighborhood stability (the location of neighbors can be achieved by GPS and the distance between neighbors can be computed). Suppose that the effective wireless communication area of the mobile node h is a circle with center h and radius r, and the area is divided into n sub-areas, i.e. n cirques H_1, H_2, H_3, ..., H_n, according to their distance from h (H_1 is the furthest from h and H_n is the nearest to h).

Definition 4. Neighbor g's vicinity on node h, denoted by $R_d(h, g)$. If node g is the neighbor of h and g is in area H_i of h, then $R_d(h, g) = i$; else if node g is not the neighbor of h, then $R_d(h, g) = -n$.

For each neighbor g of node h, g's vicinity to h can be estimated by its history information. Denoting $r_k(h,g)$ as the estimated value of g's vicinity at the k interval, and $Rd_{k-1}(h,g)$ as the actual value of g's vicinity at the $k-1$ interval, we can get the value of $r_k(h,g)$ from the estimated value and the actual value at the $k-1$ interval, shown as follows:

$$r_k(h,g) = \frac{r_{k-1}(h,g)\alpha + Rd_{k-1}(h,g)}{\alpha+1}, r_1(h,g) = Rd_1(h,g);$$

In this equation, a is a smooth factor and $a > 0$.

If $r_k(h,g) > \tilde{C}$ ($\tilde{C}$ is a threshold), g can be regarded as the *stable neighbor* of node h. $S(h)$ is denoted as the *stable neighbor set* of h, i.e., $S(h) = \{g \mid r_n(h,g) > \tilde{C}\}$.

Definition 5. Stable Path, the path Path(i,j) which is comprised of nodes i, $c1$, $c2$, $\ldots$, c_k, j ($k \geq 0$) is called a stable path when $i \in S(c_1)$, $c_1 \in S(c_2)$, $\ldots$, $c_k \in S(j)$.

Definition 6. Stable neighbor group, the set $T(h)$ is a stable neighbor group of node h if for each $i \in T$ there is at least a stable path between h and i.

The EARAM_SN algorithm improves the EARAM algorithm by replacing the neighbor set with the stable neighbor set while every node selects the stable path as its access path if there is a stable path to the replica node. In the EARAM_SN algorithm, all the replica nodes and their stable neighbor groups form a relatively stable topology. Only the access requests issued by nodes in the stable path can have impact on the replica allocation scheme and hence the oscillation of replica allocation caused by nodes motion can be reduced.

4 SIMULATION AND ANALYSIS

In this section, simulation results are shown to evaluate the performance of our algorithms. The program is written in C++ with event-driven method.

4.1 Simulation Model

The simulation parameters are presented in Table 2.

The mobile nodes are initially in the area of $1000 \cong 1000\,\mathrm{m}^2$ area with a Gaussian distribution. They move in Random Gauss-Markov Mobility Model [11]. In Figures 2 and 3, the horizontal axis indicates the interval of algorithms execution, and the vertical axis indicates the average communication cost of object access. Each experiment is performed 10 times to acquire the average value.

4.2 The Performance Comparison of Algorithms in Static Networks

In this experiment, our main concern is the effect of read-write pattern on algorithms. We compare four algorithms: the ARAM, the EARAM, the ADR_G [1],

parameter	value
area of motion	$1000\,m \cong 1000\,m$
number of mobile nodes	100
velocity of motion	$0\,m/s - 10\,m/s$
range of motion director	0–2π
communication radius of nodes	$200\,m$
number of object to be replicated	1
interval of algorithm executed	$0.1\,s$
initial number of replica	10
ratio between reads and writes	5 : 1

Table 2. Environment paramenteras

and the Static Replica Allocation algorithm (i.e. SRA, replicas are distributed on m nodes, and the replica allocation scheme does not change during the whole process of simulation). In the simulation, the read-write pattern does not change during each 10 intervals. In the simulation, the read (write) request issued by every node conforms to the Gaussian distribution and does not change during each 10 intervals. Both the ARAM algorithm and the EARAM algorithm begin with the replica allocation scheme in SRA algorithm. The simulation result is presented in Figure 3.

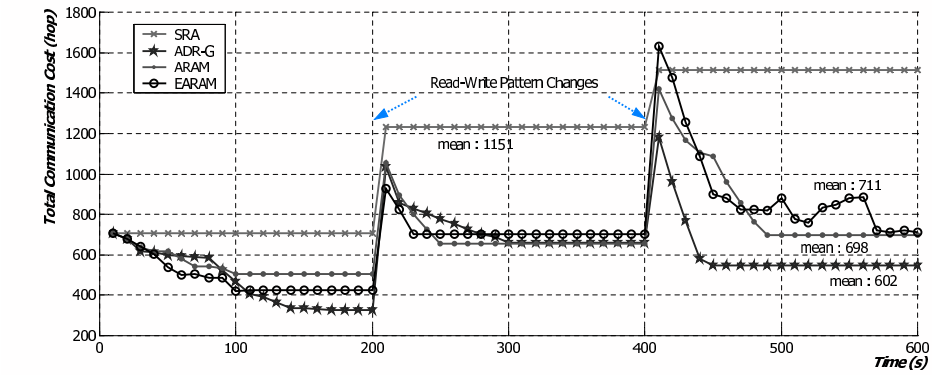


Fig. 3. Performance comparison in static networks

Figure 3 shows that when the read-write pattern is fixed, the communication cost of object access keeps decreasing until itreaches a stable value in the ARAM and ADR_G algorithm. When the read-write pattern changes, because the current read-write patterns of nodes cannot be estimated by the statistical values of write and read requests in the last interval, the communication cost increases rapidly. However, in the following 10 intervals, the read-write pattern is fixed, and the cost decreases once the ARAM algorithm or the ADR_G algorithm is executed. The similar process is repeated. This result validates the conclusion of Theorem 2. The communication

cost of object access does not decrease monotonously in the EARAM algorithm. The reason is that the EARAM algorithm relaxes the conditions of expanding and relinquishing replicas, thus replicas may be mistakenly expanded or relinquished which may lead to increase of cost.

From Figure 2, it is inferred that the mean communication cost in the ADR_G is the lowest. The reason is that the write requests are propagated among replica nodes along the MST in the ADR_G algorithm and the communication cost is $|F| - 1$.

4.3 The Performance Comparison of Algorithms in the MANET Environment

We compare the ARAM, the EARAM, the EARAM_SN, and the ADR_G algorithms with the SRA algorithm in mobile ad hoc networks. Now our main concern is the effect of nodes mobility on algorithms. In the simulation, the read request issued by every node conforms to the random distribution and is fixed during the whole process. In the EARAM_SN algorithm, $n = 10$, $a = 0.5$, and $C = 0.25 \cdot 10$.

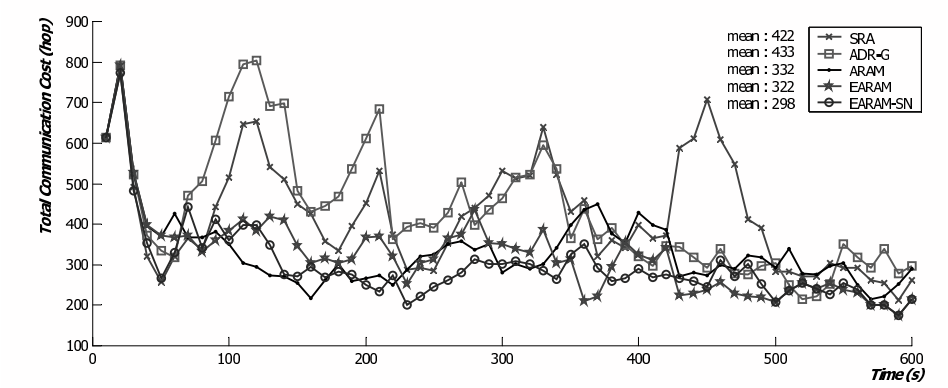


Fig. 4. Performance Comparison in MANET Environment

In Figure 4, nodes move in Random Waypoint Mobility Model [11]. It shows that compared with the ADR_G algorithm, the mean cost is 23 % less in the ARAM algorithm, 25 % less in the EARAM algorithm and is 31 % less in the EARAM_SN algorithm. The reason is that in the ADR_G algorithm, the write requests are propagated among replica nodes along the MST needed to reach all the replicas and the resulting optimal configuration tends to locate replicas in nodes adjacent to each other. While nodes move and the network topology changes, the replica nodes are no longer adjacent physically, however the write requests are propagated along the MST built in the last interval. Therefore the data access communication cost in the ADR_G algorithm increases rapidly when the network topology changes. From Figure 4 we know that the communication cost of object access in the MANET environment is greatly reduced in our algorithms. The simulation result also indicates

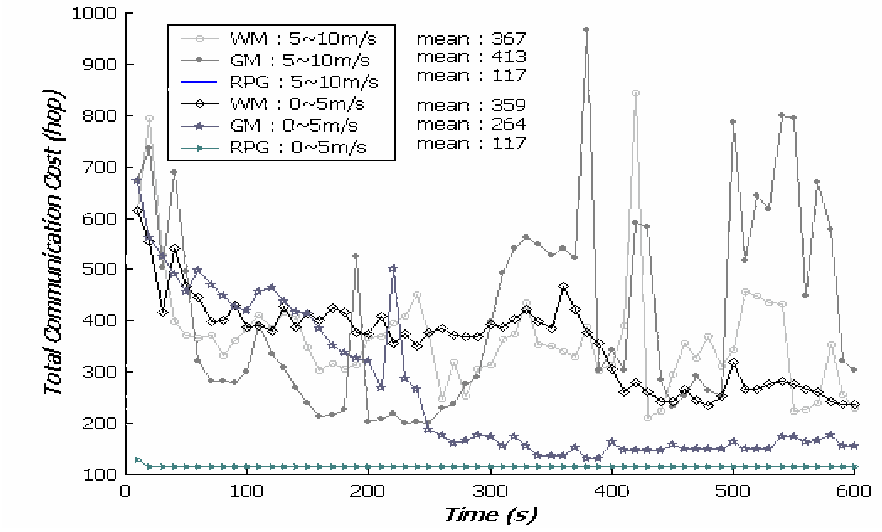


Fig. 5. The Effect of Mobile Pattern on the EARAM Algorithm

that the communication cost is the most stable in the EARAM_SN algorithm among that in other algorithms.

For analyzing the effect of velocity on the algorithm for different mobile patterns, we select three mobile patterns: Random Waypoint, Gauss-Markov, and Reference Point Group [11] (denoted by RM, GM, and PRG respectively in Figure 5). Figure 5 indicates that for the Reference Point Group mobile pattern, because all nodes are in a single mobility group and the network topology is stable, the velocity hardly influences the replica allocation scheme. However, for Random Waypoint and Gauss-Markov mobile patterns, the node velocities influence the replica allocation scheme greatly and the higher the node velocities are the more rapidly the communication cost of data access changes.

Table 3 indicates that the replica allocation scheme generated by the EARAM_SN algorithm is more stable than that generated by the EARAM algorithm. The number of nodes that are expanded to be replica node in one interval and relinquish replica in the next interval is reduced. Therefore the oscillation of replica allocation is cut down in the EARAM_SN algorithm.

5 CONCLUSION

In this paper, a new distributed dynamic adaptive replica allocation algorithm for the MANET environments is proposed. The algorithm aims at reducing communication cost and improving system performance. It can also adaptively adjust replica allocation scheme according to node mobility and the read-write pattern. The si-

No.	EARAM		EARAM_SN	
	cost(F)	F(No. of node	cost(F)	F(No. of node
1	266	0 81	371	2 81 23 30
2	329	0 81	367	2 77 23 30
3	442	0 81	372	2 5 7 30
4	341	0 9 81	333	5 77 9 30
5	411	0 9 26	361	2 77 9 30
6	360	9 19 26	385	77 9 11 30
7	398	9 26 37	412	77 9 11 15 30
8	398	9 26 33	384	77 9 11 13 30
9	349	77 26 33	418	77 9 11 20 30
10	276	18 26 33	410	9 11 18 20 30

Table 3. Changes of replica allocation scheme

mulation results show that the communication cost of object access in the MANET environment is reduced efficiently in our algorithms. As a part of our future work, these algorithms should be improved to deal with replica conflict resolution and reconciliation problem during network partitioning in the MANET environment. The replica consistency protocol should also be investigated.

REFERENCES

[1] Wolfson, O.—Jajodia, S.—Huang, Y.: An Adaptive Data Replication Algorithm. ACM Transactions on Database Systems, Vol. 22, 1997, No. 4, pp. 255–314.

[2] Cook, S. A.—Pachl, J. K.—Pressman, I. S.: The Optimal Location of Replicas in a Network Using a Read-One-Write-All Policy. Distributed Computing, Vol. 15, 2002, No. 1, pp. 7–17.

[3] Barbara, D.—Imielinski, T.: Sleeper and Workholics: Caching Strategies in Mobile Environment. In Proceedings of ACM SIGMOD'94, 1994, pp. 1–12.

[4] Cay, J.—Tan, K. L.—Ooi, B. C.: On Incremental Cache Coherency Schemes in Mobile Computing Environments. In Proceedings of IEEE ICDE'97, 1977, pp. 114–123.

[5] Huang, Y.—Pistla, S.—Wolfson, O.: Data Replication for Mobile Computer. In Proceedings of ACM SIGMOD'94, 1994, pp. 13–24.

[6] Pitoura, E.—Bhargava, B.: Maintaining Consistency of Data in Mobile Distributed Environments. In Proceedings of IEEE ICDCS'95, 1995, pp. 404–413.

[7] Hare, T.: Effective Replica Allocation in Ad hoc Networks for Improving Data Accessibility. In Proceeding of IEEE Infocom 2001, 2001, pp. 1568–1576.

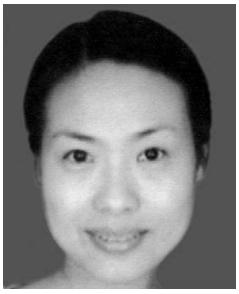
[8] Hare, T.: Replica Allocation in Ad hoc Networks with Periodic Data Update. In Proceedings of Int'l Conference on Mobile Data Management(MDM 2002), 2002, pp. 79–86.

[9] Wang, K.—Li, B.: Efficient and Guaranteed Service Coverage in Partitionable Mobile Ad-hoc Networks. In IEEE Joint Conference of Computer and Commu-

nication Societies (INFOCOM'02), New York City, New York, 2002, June 23–27, pp. 1089–1098.

[10] Wolfson, O.—Milo, A.: The Mulicast Policy and Its Relationship to Replicated Data Placement. ACM Transaction on Database Systems, Vol. 16, 1991, No. 1, pp. 181–205.

[11] Camp, T.—Boleng, J.—Davies, V.: A Survey of Mobility Models for Ad Hoc Network Research. Wireless Communication & Mobile Computing (WCMC): Special Issue On Mobile Ad Hoc Networking: Reach, Tends and Applications, Vol. 2, 2002, No. 5, pp. 483–502.



Zheng Jing received her BE degree in computer engineering from Beijing Institution of Command and Technology in 1991. She received her MS degree in computer science from National University of Defense Technology in 1998.She currently is a PHD candidate in the Department of Computer Science at National University of Defense Technology. Her research interests include mobile computing, replication.

Lu Xieng received his BE degree in computer engineering from Harbin Institution of Military Engineering in 1970. He is currently a professor in the Department of Computer Science at National University of Defense Technology. His interests are in the field of computer networks and parallel distributed processing.

Yang Kan received his BE degree in computer engineering from Northwest Uninversity of Industry in 1998. He is a candidate master of the National University of Defense Technology, China. His research interests include mobile computing and database systems.

Wang Yijie is received her BE degree, PHD degree in computer science from National University of Defense Technology in 1989 and in 1999. She is currently an associate professor in the Department of Computer Science at National University of Defense Technology. Her research interests include object-oriented database, mobile computing.

APPENDICES

A PROOF OF THEOREM 1

Proof. First we show that REPLICA_LOCATION $\in NP$. Suppose we are given a graph $G(V, E)$, $0 \leq \theta \leq 1$, and an integer k. The certificate we choose is the replica allocation scheme $F \subseteq V$. The verification algorithm affirms that $\text{cost}(F, \theta) \leq k$, and then it checks, for the F and the input $<G, \theta, k>$, whether $\text{cost}(F, \theta) \leq k$. This verification can be performed straightforwardly in polynomial time.

Second we prove that the REPLICA_LOCATION problem is NP-hard by showing that VERTEX_COVER $\leq_p$ REPLICA_LOCATION. The VERTEX_COVER is to find a vertex cover of the minimum size in a given graph. As a language, it is defined as VERTEX_COVER = $\{<G, q>$: graph G has a vertex cover of size $k\}$.

We reduce the vertex_cover problem to the replica_location problem. The reduction algorithm takes as input an instance $<G, q>$ of the vertex_cover problem. The output of reduction algorithm is the instance $<G', \theta, k>$. Given the undirected graph $G = (V, E)$, where $E = \{a_1, \ldots, a_m\}$, and $V = \{Y_1, \ldots, Y_t\}$, we define the undirected graph $G' = (V', E')$, where vertex set $V' = E \cup V$, and $V' = \{a_1, \ldots, a_m, b_1, \ldots, b_t\}$, where b_i corresponds to Y_i; $E' = \{(i, j) \mid$ either i and j are elements of $\{b1, \ldots, b_t\}$, or vertex j (or i) cover edge j (or i) in $G\}$.

Each a_i is connected to two b_i, therefore G' is connected, and $|V'| = m + t$.

Now we will prove that if the instance $<G, q>$ of the vertex_cover problem has a solution $C \subseteq V$ with $|C| \leq q$ which is a vertex cover in G, and if $F = C$ is selected, then $\min \text{cost}(F, \theta) \leq k$; if the instance of the vertex_cover problem has no solution, however, then $\text{cost}(F, \theta) > k$ for each choice of F.

Define the following parameters:

$$\theta = \frac{1.5}{|V'|} = \frac{1.5}{m + t}$$

$$A_0 = |V'|\theta(q - 1) = 1.5(q - 1)$$

$$B_0 = m + t - q$$

$$k = A_0 + B_0 = 1.5(q-1) + m + t - q = 0.5q + m + t - 1.5$$

Suppose that the instance $< G, q >$ of the vertex_cover has a solution $C = \{Y_{i_1}, \ldots, Y_{i_q}\}$. Select $F = \{b_{i_1}, \ldots, b_{i_q}\} = C$. The distance between any two nodes in F is 1, thus for each $s \in F$, $\sum_{j \in F} d(s, j) = q - 1$. Hence, by (2)

$$B(F) = \sum_{i \in V} d(i, F) = |V'| - q = m + t - q = B_0.$$

By (2), $\text{cost}(F, \theta) = A(F) + B(F) = A_0 + B_0 = k$, therefore $\min \text{cost}(F, \theta) \leq k$.

Conversely we show that if the instance $<G, q>$ of vertex_cover has no solution then $\min \text{cost}(F, \theta) > k$. Consider any subset $F \subseteq V'$, and let $r = |F|$. For each $i \in F$, i is in $\{a_1, \ldots, a_m\}$ or is in $\{b_1, \ldots, b_t\}$, then $\sum_{j \in F} d(i, j) \geq r - 1$. Therefore:

$$A(F) = \sum_{i \in F} \sum_{j \in F} W_{re}(i) d(i, j) \geq |V'| u (r-1) = 2(r-1) \tag{11}$$

Now partition V' into three sets of sizes n_0, n_1, and n_2 corresponding to those nodes of distance 0, 1, and 2 or more from F, respectively. Then $n_0 = |F| = r$, $n_0 + n_1 + n_2 = |V'| = m + t$, and $n_1 = m + t - r - n_2$. Therefore:

$$B(F) \geq n_1 + 2n_2 = m + t - r - n_2 + 2n_2 = m + t - r + n_2 \tag{12}$$

We consider three cases, and show in each case that $\min \text{cost}(F, \theta) > k$. Since $F \subseteq V'$ is arbitrary, it follows that $\min \text{cost}(F, \theta) > k$.

Case 1: $r = |F| < q$

In this case, $n_2 > q - r$ is true. It is proved as follows:

If $F \subseteq \{a_1, \ldots, a_m\}$, according to the definition of the edge in G', $m - r$ vertices in G' have a distance of 2 or more from F. Because m edges in G can be covered by no more than m vertices, and resulting from the assumption above that the instance $<G, q>$ of vertex_cover has no solution, $m > q$ is true. Thus, $n_2 \geq m - r > q - r$ is true.

Otherwise, assume that x vertices of F are in $\{b_1, \ldots, b_t\}$ of G', those vertices are $b_1, \ldots, b_x$; and other $r - x$ vertices are in $\{a_1, \ldots, a_m\}$ of G'. Because there is a edge between any two vertices of $\{b_1, \ldots, b_t\}$ in G', vertices which have a distance of 2 or more from F must be some n_2 vertices of $\{a_1, \ldots, a_m\}$ in G'. They are assumed as $a_1, \ldots, a_{n_2}$. Corresponding to G, edges in $\{a_1, \ldots, a_m\}$ of G can be parted into three sets: one part of edges of size n_2 correspond to vertices of distance of 2 or more from F in G', $a_1, \ldots, a_{n_2}$, and they can be covered by no more than n_2 vertices in G. The second part of edges correspond to $r - x$ vertices of F in G'. They can be covered by no more than $r - x$ vertices in G. The third part of edges correspond to vertices in $\{a_1, \ldots, a_m\}$ which have distance of 1 from F in G'. There is at least one edge between these vertices

and vertices $b_1, \ldots, b_x$. As a result, these edges are covered by vertices $Y_1, \ldots, Y_x$ in G corresponding to $b_1, \ldots, b_x$. Therefore, all edges in G can be completely covered by no more than $n_2 + r - x + x = n_2 + r$ vertices. Because of the assumption above that the instance $<G, q>$ of the vertex_cover has no solution, thus the inequations $n_2 + r > q$ then $n_2 > q - r$ are true.

Similarly, it can be proved that in the case of $F \subseteq \{b_1, \ldots, b_t\}$ (in this case $x = r$) the inequation $n_2 > q - r$ is also true. Therefore:

By (12), $B(F) > m + t - r + q - r = m + q + t - 2r$.

By (11) and (2), $\text{cost}(F, \theta) = A(F) + B(F) > 1.5(r-1) + m + t - r + q - r = m + q + t - 2r = 0.5q + m + t - 1.5 + 0.5(q - r) > k$.

Case 2: $r > q$

By (12), $B(F) \geq m + t - r$, thus:

$$\text{cost}(F, \theta) = A(F) + B(F) \geq 1.5(r-1) + m + t - r = 0.5r + m + t - 1.5 > k$$

Case 3: $r = q$

Because $r = q$ and it is assumed that the instance $<G, q>$ of the vertex_cover has no solution, edges in G cannot be covered by F. Therefore there is at least one node in $\{a_1, \ldots, a_m\}$ which has the distance of 2 or more from F. Thus, $n_2 \geq 1$. By (12), $B(F) \geq m + t - r + 1$.

By (11) and (2),

$$\text{cost}(F, \theta) = A(F) + B(F) \geq 1.5(r-1) + m + t - r + 1 = 0.5r + m + t - 1.5 + 1 > k.$$

□

B PROOF OF THEOREM 2

Proof. In the static networks, suppose that the read-write pattern doesn't change and the ARAM algorithm is executed at the end of the x^{th} interval. If no condition of the Expansion_test, Relinquishment_test, and Switch_test operation is satisfied for all replica nodes, no condition of these three operations is satisfied at the end of the $x + 1^{\text{st}}$ interval for all replica nodes. In this case, the replica allocation scheme does not change any more and the communication cost of data access is stable.

It will be verified that the data access communication cost decreases by the Expansion_test, Relinquishment_test, and Switch_test operation. Assume that the replica allocation scheme in the x^{th} interval is F and that in the $x + 1^{\text{st}}$ interval is F'.

Case 1: Expansion_test. If s expands replica to u at the end of the x^{st} interval, $F' = F \cup \{u\}$. Now all nodes are parted into three sets. The first part of nodes, denoted as Ψ, accesses replica node s through u in the x^{th} interval, and accesses

replica u in the $x+1^{\text{st}}$ interval; the second part of nodes are denoted as Ω. In the $x+1^{\text{st}}$ interval, for each node i of Ω, i accesses a replica node that is not u, i.e. $pi \neq u$; and the third part of nodes are nodes of set A which has been defined in Section 3. Nodes of A change the access path and access replica node u in the $x+1^{\text{st}}$ interval.

$W_{\text{change}}(i)$, p_i, $W_{\text{from}}(u)$, $R_{\text{from}}(u)$, Δchange, W, R_{change}, and W_{change} have been described in Section 3. The subscript x and $x+1$ represents the x^{th} interval and the $x+1^{\text{st}}$ interval respectively. The subscripts Ω, Ψ, A, and B represent the statistical value issued by nodes of the set Ω, Ψ, A, and B respectively.

1. For any node $i \in \Psi$, $d(i, F') = d(i, F)$. Therefore, by (1),

$$\text{cost}R(F')_\Psi = \text{cost}R(F)_\Psi - R_{\text{from}}(u)_x$$

$$\text{cost}W_{\text{forward}}(F')_\Psi = \text{cost}W_{\text{forward}}(F)_\Psi - W_{\text{from}}(u)_x$$

   In the $x+1^{\text{st}}$ interval, all write requests issued by nodes of Ψ are propagated to all the replica nodes of F' from u; therefore the update cost changes. While s is at a site on all the shortest paths between u and every node of F, the increase of update cost of F' is the greatest. In this case, for each $j \in F$, $d(u, j) = d(s, j) + 1$, thus $q(u)_{x+1} = q(s)_x + |F|$. By (1),

$$\begin{aligned}\text{cost}W_{\text{up}}(F') &= W_{\text{from}}(u)_x q(u)_{x+1} \leq W_{\text{from}}(u)_x (q(s)_x + |F|) \\ &= \text{cost}W_{\text{up}}(F)_\Psi + W_{\text{from}}(u)_x |F|\end{aligned}$$

   By (1),

$$\text{cost}(F')_\Psi \leq \text{cost}(F)_\Psi - R_{\text{from}}(u)_x + (|F| - 1) W_{\text{from}}(u)_x \tag{13}$$

2. For nodes of Ω, the distance to F does not change while expanding replica to u, hence by (1),

$$\text{cost}R(F')_\Omega = \text{cost}R(F)_\Omega$$

$$\text{cost}W_{\text{forward}}(F')_\Omega = \text{cost}W_{\text{forward}}(F)_\Omega$$

   In the $x+1^{\text{st}}$ interval, all write requests issued by nodes of Ω are propagated to the new replica node u, beside to replica nodes of F. While s is at a site on all the shortest paths between u and every node of F, the increase of update cost of F' is the greatest. In this case, for each $i \in F$, $d(u, i) = d(s, i) + 1$. The write requests propagated to u from other replica nodes are

$$\text{cost}W_{\text{up}}(F')_\Omega = \text{cost}W_{\text{up}}(F)_\Omega + \text{cost}W_{\text{up}}(u)_\Omega$$

   For $i \in F$, in the x^{th} interval, some write requests received by i are potentially from nodes of A, and these write requests are denoted as $W_{\text{change}}(i)$.

Therefore:

$$\begin{aligned}\text{cost}W_{\text{up}}(F')_\Omega &= \text{cost}W_{\text{up}}(F)_\Omega + \sum_{i\in F}(W_{re}(i)_x - W_{\text{change}}(i))d(i,u) - W_{\text{from}}(u)_x \\ &\le \text{cost}W_{\text{up}}(F)_\Omega + \sum_{i\in F}(W_{re}(i)_x - W_{\text{change}}(i))(d(i,s)+1) - W_{\text{from}}(u)_x \\ &= \text{cost}W_{\text{up}}(F)_\Omega + \sum_{i\in F}(W_{re}(i)_x - W_{\text{change}}(i))d(i,s) \\ &\quad + W_x - W_{\text{change}} - W_{\text{from}}(u)_x\end{aligned}$$

By (1),

$$\begin{aligned}\text{cost}(F')_\Omega &\le \text{cost}(F)_\Omega + \sum_{i\in F}(W_{re}(i)_x - W_{\text{change}}(i))d(i,s) \\ &\quad + W_x - W_{\text{change}} - W_{\text{from}}(u)_x \qquad (14)\end{aligned}$$

3. For each $i \in A$, according to the definition of A, $d(i,F') = d(i,F) - 1 = d(i,u)$. Thus, by (1),

$$\text{cost}R(F')_A = \text{cost}R(F)_A - R_{\text{change}}$$

$$\text{cost}W_{\text{forward}}(F')_A = \text{cost}W_{\text{forward}}(F)_A - W_{\text{change}}$$

For $i \in A$, $W(i)q(u)_{x+1} \le W(i)(q(s)_x + |F|)$. Because $\text{cost}W_{\text{up}}(R)_A = \sum_{i\in A} w_i q(p_i)_x$, and $\Delta\text{change} = \sum_{i\in A} W(i)(q(s)_x - q(p_i)_x)$, $\text{cost}W_{\text{up}}(F')_A = \sum_{i\in A} W(i)q(u)_{x+1} \le \sum_{i\in A} W(i)(q(s)_x + |F|) = \Delta\text{change} + \text{cost}W_{\text{up}}(F)_A + W_{\text{change}}|F|$.
By (1),

$$\text{cost}(F')_A \le \text{cost}(F)_A - R_{\text{change}} + (|F|-1)W_{\text{change}} + \Delta\text{change} \qquad (15)$$

By (13), (14), and (15),

$$\begin{aligned}\text{cost}(F') &\le \text{cost}(F) - R_{\text{from}}(u)_x - R_{\text{change}} + (|F|-2)(W_{\text{from}}(u)_x + W_{\text{change}} \\ &\quad + \sum_{\imath\in F}(W_{re}(i) - W_{\text{change}}(i))d(i,s) + W_x + \Delta\text{change}\end{aligned}$$

Because the condition (3) is satisfied,

$$\text{cost}(F') < \text{cost}(F).$$

Case 2: Relinquishment_test. If replica node s relinquishes replica at the end of the x^{th} interval, then $F' = F - \{s\}$. All nodes are parted into three sets: the first part of nodes are nodes of set B which have been defined in Section 3; the second part of nodes, denoted as Ψ, are nodes which access s in the x^{th} interval,

but this part of nodes does not include nodes of B; and the remaining nodes are denoted as Ω.

Replica node z, p'_i, and w, together with access request R_B , W_B and $R_{re}(s)$ have been described in the section III.

1. According to the definition of B, in the $x+1^{\text{st}}$ interval, $\text{cost}R(F')_B = \text{cost}R(F)_B$, and $\text{cost}W_{\text{forward}}(F')_B = \text{cost}W_{\text{forward}}(F)_B$. Nodes of B access s in the x^{th} interval; however, if s relinquishes replica, nodes of B change access path and access replica p'_i in the $x+1^{\text{st}}$ interval. Therefore:

$$\text{cost}W_{\text{up}}(F)_B = W_B q(s)_x = W_B \sum_{i\in F} d(i,s)$$

$$\text{cost}W_{\text{up}}(F')_B = W_B q(p'_i)_{x+1} = W_B \sum_{i\in F-\{s\}} d(i,p'_i)$$

$$= W_B \left(\sum_{i\in F} d(i,p'_i) - d(s,p'_i) \right)$$

By (1),

$$\text{cost}(F')_B = \text{cost}(F)_B + W_B \left(\sum_{i\in F} d(i,p'_i) - d(s,p'_i) \right) - W_B \sum_{i\in F} d(i,s)$$

$$= \text{cost}(F)_B + W_B(q(p'_i)_x - q(s)_x - d(s,p'_i)) \tag{16}$$

2. For each node $j \in \Psi$, $d(j,F) = d(j,s)$. As the description of z, $d(j,F') \le d(j,s) + d(s,z)$. By (1)

$$\text{cost}R(F')_\Psi \le \text{cost}R(F)_\Psi + (R_{re}(s)_x - R_B)d(s,z)$$

$$\text{cost}W_{\text{forward}}(F')_\Psi \le \text{cost}W_{\text{forward}}(F) + (W_{\text{re}}(s)_x - W_B)d(s,z)$$

$$\text{cost}W_{\text{up}}(F)_\Psi = (W_{re}(s)_x - W_B)q(s)_x = (W_{re}(s)_x - W_B)\sum_{i\in F} d(i,s)$$

$$\text{cost}W_{\text{up}}(F')_\Psi \le (W_{re}(s)_x - W_B)q(w)_{x+1}$$

$$= (W_{re}(s)_x - W_B) \sum_{i\in F-\{s\}} d(i,w) = (W_{re}(s)_x - W_B) \left(\sum_{i\in F} d(i,w) - d(s,w) \right)$$

Therefore:

$$\text{cost}(F')_\Psi \le \text{cost}(F)_\Psi + (R_{re}(s)_x + W_{re}(s)_x - R_B - W_B)d(s,z)$$

$$+ W_{re}(s)_x(q(w)_x - q(s)_x - d(s,w))$$

$$+ W_B(q(s)_x) - d(s,w)) + W_B(q(s)_x - q(w)_x + d(s,w)) \tag{17}$$

3. In the $x+1^{\text{st}}$ interval, the read access cost issued by nodes of Ω does not change, and the cost of propagating write request to s is subtracted from the total write access cost. By (1),

$$\begin{aligned}\text{cost}(F')_\Omega &= \text{cost}(F)_\Omega - \sum_{i\in F-\{s\}} W_{re}(i)_x d(i,s) \\ &= \text{cost}(F)_\Omega - \sum_{i\in F} W_{re}(i)_x d(i,s) \end{aligned} \tag{18}$$

By (16), (17), (18),

$$\begin{aligned}\text{cost}(F') \le \text{cost}(F) + (R_{re}(s)_x + W_{re}(s) - R_B - R_B)d(s,z) \\ + W_{re}(s)_x(q(w)_x - q(s)_x - d(s,w)) + W_B(q(p'_i)_x - q(w)_x - d(s,p'_i) + d(s,w)) \\ - \sum_{i\in F}(i)_x d(i,s)\end{aligned}$$

Because the condition (4) is satisfied, $\text{cost}(F') < \text{cost}(F)$

Case 3: Switch_test. If the switch_test operation on neighbor node u of replica node s is executed and the switch succeeds, then $F' = F - \{s\} + \{u\}$. All nodes are parted into three sets: the first part of nodes, denoted as Ψ, access replica node s through u in the x^{th} interval, and access replica node u in the $x+1^{\text{st}}$ interval; the second part of nodes, denoted as Ω, access replica node s, but do not pass through u in the x^{th} interval; and the rest nodes, denoted as δ, access the other replica node except s in the x^{th} interval.

1. For each $i \in \Psi$, $d(i,F') = d(i,F)-1$, thus the decrease of communication cost is $R_{\text{from}}(u)_x + W_{\text{from}}(u)_x$. When s is at a site on all the shortest paths between u and every node of F, $q(u)_{x+1} = q(s)_x + |F| - 1$, and the upper bound of the increase of communication cost is acquired, which is $W_{\text{from}}(u)_x(|F|-1)$. Therefore:

$$\begin{aligned}\text{cost}(F')_\Psi &\le \text{cost}(F)_\Psi - (R_{\text{from}}(u)_x + W_{\text{from}}(u)_x) + W_{\text{from}}(u)_x(|F|-1) \\ &= \text{cost}(F) - R_{\text{from}}(u)_x + W_{\text{from}}(u)_x(|F|-2)\end{aligned} \tag{19}$$

2. For each $i \in \Omega$, when the access path of i is $i, \ldots, s, u$ in the $x+1^{\text{st}}$ interval and s is at a site on all the shortest paths between u and every node of F, $d(i,F') = d(i,F)+1$ and $q(u)_{x+1} = q(s)_x + |F| - 1$. Then the upper bound of the increase of communication cost is acquired. Hence,

$$\begin{aligned}\text{cost}(F')_\Omega \text{cost}(F)_\Omega + (R_{re}(s)_x - R_{\text{from}}(u)_x + W_{re}(s)_x - W_{rmfrom}(u)_x) \\ + (W_{re}(s)_x - W_{\text{from}}(u)_x)(|F|-1) \\ = \text{cost}(F)_\Omega + (R_{re}(s)_x - R_{\text{from}}(u)_x) + ((W_{re}(s)_x - W_{\text{from}}(u)_x)|F|\end{aligned} \tag{20}$$

3. For each $i \in \delta$, if i does not change the access path in the $x+1^{\text{st}}$ interval, when s is at a site on all the shortest paths between u and every node of F, the communication cost of each write request from i increases 1 and that of read request from i does not change; if i changes the access path in the $x+1^{\text{st}}$ interval (i.e. the accessed replica node is u in the $x+1^{\text{st}}$ interval and p_i in the x^{th} interval), the upper bound of the increase of communication cost is acquired when s is at a site on all the shortest paths between u and every node of F. The increased cost issued by every write request is not more than $q(s)_x + |F| - 1 - q(p_i)_x$. Therefore the upper bound of the total increase of communication cost is $\sum_{j \in F-\{s\}} W_{re}(j)\Delta Q$, in which $\Delta Q = \max(1, q(s)_x + |F| - 1 - q(j)_x)$, $j \in F - \{s\}$. Thus

$$\text{cost}(F')_\delta \le \text{cost}(F_\delta) + \sum_{j \in F-\{s\}} \tag{21}$$

By (19), (20) and (21),

$$\text{cost}(F') \le \text{cost}(F) - 2(R_{\text{from}}(u)_x + W_{\text{from}}(u)_x) + R_{re}(s)_x + W_{re}(s)_x + \sum_{j \in F-\{s\}} W_{re}(j)\Delta(Q) \tag{22}$$

Because the condition (5) is satisfied, $\text{cost}(F') < \text{cost}(F)$.

□