

Stable Neighbor Based Adaptive Replica Allocation in Mobile Ad Hoc Networks*

Zheng Jing, Su Jinshu, Yang Kan, and Wang Yijie

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

Abstract. In mobile ad hoc networks (MANET), nodes move freely and the replica allocation in such a dynamic environment is a significant challenge. 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.

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 don't 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

* This research was supported by the National Natural Science Foundation of China (No. 90104001).

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 isn't suitable for the mobile environment as well.

Several strategies [3,4,5] for replicating or caching data have been proposed in traditional wireless mobile networks. 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 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 [6], much information needs to be exchanged among nodes, especially when the topology of network changes rapidly. In [7], an algorithm is proposed to predict the network partitioning and to allocate replicas to ensure the service availability. It is known, all these algorithms [6,7] 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 relinquishing 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 demonstrated; and finally in section 5, the summary and some future work are presented.

2 The Cost Model

In our research, hops is 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, so we use the hops between two nodes to measure the communication costs between these two nodes.

We suppose that the access request for an object is sent to the closest replica in the network. The read request is served by the closest replica node, and 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. The ROWA (READ-ONE-WRITE-ALL) policy is used to ensure the consistency of the replicas, and we assume that each individual access is independent.

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 $cost(F)$, can be computed as below:

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

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 number 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. $costW_{forward}(F) = \sum_{i \in V} W(i)d(i, F)$, is the cost of forwarding write requests to replica nodes; $costW_{up}(F) = \sum_{s \in F} \sum_{j \in F} W_{re}(s)d(s, j)$, represents the cost of propagating write requests among replica nodes; $costR(F) = \sum_{i \in V} R(i)d(i, F)$, refers to the total access cost of read requests.

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

For general static networks, the problem of finding an optimal replica allocation scheme has been proved to be NP-complete for different cost model [8, 2]. As for the cost model defined by (1), this problem is also proved to be NP-complete in [10]. As for the MANET environment, it is more difficult to find the optimal replica allocation. Thus a distributed adaptive replica allocation algorithm is proposed to find the near-optimal replica allocation scheme.

3 Adaptive Replica Allocation Algorithms

3.1 The Adaptive Replica Allocation 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. The *ARAM (the Adaptive Replica Allocation Algorithm In MANET)* algorithm has been proposed in [10]. In the *ARAM* 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.

The algorithm is executed at each replica node periodically and independently. The duration of the period t is a uniform system parameter. The period tends to be shorter with more frequent topological changes and read-write pattern changes.

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

Table 1. The ARAM algorithm

```

execute Expansion_test for each non-replica neighbor  $u$  of  $s$ 
if exist node  $u^*$ ,  $u^*$  satisfies the condition of expansion
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
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

```

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

1. *Expansion_test*. For the neighbor u of the replica node 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 condition of expansion is satisfied (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 declines.), a replica is expanded to u^* and $F'=F \cup \{u^*\}$.
2. *Relinquishment_test*. If the number of update request received by replica node 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.
3. *Switch_test*. The switch test will allocate the replica to the neighbor node which receives more read and write requests. Then u^* becomes a replica node while s isn't a replica node any more.

THEOREM 1. For a static network, suppose that the read-write pattern doesn't 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 [10].

In the ARAM algorithm, information of all access paths is collected, which makes the algorithm complex. 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 [10]. The EARAM (*The Enhanced ARAM Algorithm*) algorithm ignores the changes of the access path caused by the changes in replica allocation scheme. Thus the replica expansion condition can be simplified as below:

$$R_{from}(u) > W_{re}(s) - W_{from}(u) + \sum_{i \in F} W_{re}(i) d(i, s) \quad (2)$$

Similarly, the relinquishment condition can be simplified as below:

$$\sum_{i \in F} W_{re}(i) d(i, s) > (R_{re}(s) + W_{re}(s)) d(s, F - \{s\}) \quad (3)$$

The switch condition is achieved as below:

$$R_{from}(u) + W_{from}(u) > 1/2(R_{re}(s) + W_{re}(s)) \quad (4)$$

Where $R_{re}(s)$ is the read requests received by s from itself and other nodes; $R_{from}(u)$ and $W_{from}(u)$ are read and write requests received by s from u respectively.

The *EARAM* algorithm is the same as the *ARAM* algorithm except that the expansion, relinquishment, and switch conditions in the *ARAM* algorithm are replaced by the condition (2), (3) and (4) respectively. The information collected by the *EARAM* algorithm is not enough to ensure that the communication cost of data access decreases once the *EARAM* algorithm is executed, and replica may be mistakenly expanded or relinquished. But the total communication cost tends to decline.

3.2 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 the *neighborhood stability* of them (the distance between neighbors can be achieved by GPS). Suppose that the effective wireless communication area of the mobile node h is a circle with the center h and the radius r , and the area is divided into n sub-areas, i.e. n cirques $H_1, H_2, H_3, \dots, H_n$, according to their distance from h (H_1 is the furthest from h and H_n is the nearest to h).

Definition 3: Neighbor g 's vicinity on node h , denoted by $Rd(h, g)$. If node g is the neighbor of h and g is in area H_i of h , then $Rd(h, g) = i$; else if node g isn't the neighbor of h , then $Rd(h, g) = -n$.

For each neighbor g of node h , g 's vicinity on 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 below:

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

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

Definition 5. 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 by C++ with event-driven method.

The simulation parameters are presented as follows: area of motion is $1000m \times 1000m$; the number of mobile nodes is 100; the velocity of motion is 0 m/s - 10 m/s; the range of motion director is $0 - 2\pi$; the communication radius of nodes is 200m; the number of object to be replicated is 1; the interval of algorithm executed is 0.1s; the initial number of replica is 10; the ratio between reads and writes is 5:1. The mobile nodes move in Random Waypoint Mobility Model [9]. Each experiment is performed 10 times to acquire the average values.

Firstly, we compare the performance 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], and the Static Replica Allocation algorithm (i.e. *SRA*, replicas are distributed on m nodes, and the replica allocation scheme doesn't change during the whole process of simulation). In the simulation, the read-write pattern doesn't change during each 10 intervals. The simulation result is presented in Fig.1.

Fig.1 shows that when the read-write pattern is fixed, the communication cost of object access keeps decreasing until reaches a stable value in the *ARAM* and the *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 similar process is repeated. This result validates the conclusion of Theorem 1. From Fig.1, it is inferred that the mean communication costs 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 $|F|-1$.

Secondly, we compare the performance of algorithms in the MANET environment. We compare the *ARAM*, the *EARAM*, the *EARAM_{SN}*, and the *ADR_G* algorithms with the *SRA* algorithm. 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 $\tilde{C}=0.25*10$.

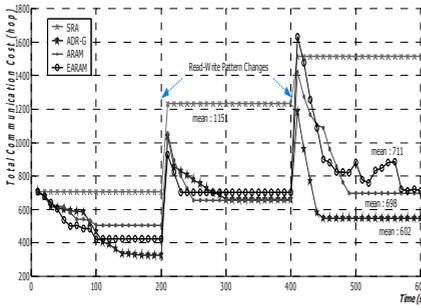


Fig. 1. Performance in Static Networks

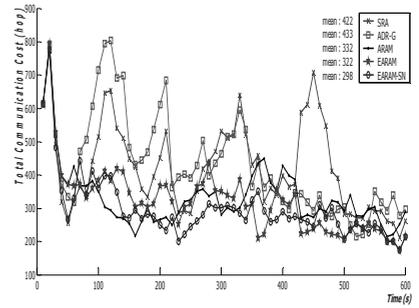


Fig. 2. Performance in MANET

Fig.2 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 of 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. Therefore the data access communication cost in the *ADR_G* algorithm increases rapidly when the network topology changes. From Fig.2 we know that the communication cost of object access in the MANET environment is greatly reduced in our algorithms. The simulation result also indicates that the communication cost is the most stable in the *EARAM_SN* algorithm among that in other algorithms.

Table 2. Changes of Replica Allocation Scheme

No.	<i>EARAM</i>		<i>EARAM_SN</i>	
	cost(<i>F</i>)	<i>F</i> (No.of node)	cost(<i>F</i>)	<i>F</i> (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

Table 2. presents 8 sequential replica allocation schemes generated by the *EARAM* algorithm and the *EARAM_SN* algorithm respectively in one experiment.

Table 2. 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. Also it can adaptively adjust replica allocation scheme according to node mobility and the read-write pattern. The simulation 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. O. Wolfson, S. Jajodia, and Y.Huang, "An Adaptive Data Replication Algorithm", *ACM Transactions on Database System*, vol. 22, no. 4, 1997, pp. 255-314.
2. S.A Cook, J.K Pahl, and I.S Pressman, "The Optimal Location of Replicas in A Network Using A Read-One-Write-All Policy", *Distribute Computing*, vol.15, no.1, 2002, pp. 7-17.
3. D Barbara, T.Imielinski, " Sleeper and Workholics: Caching Strategies in Mobile Environment", in: *Proceedings of ACM SIGMOD'94*,1994, pp. 1-12.
4. J Cay, K.L Tan, B.C.Ooi, " On Incremental Cache Coherency Schemes in Mobile Computing Environments", In *Proceeding of IEEE ICDE'97*, 1997, pp. 114-123.
5. Y Huang, S.Pistla, and O. Wolfson, "Data Replication for Mobile Computer", In *Proceedings of ACM SIGMOD'94*,1994, pp.13-24.
6. T.Hare, "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.
7. K Wang, B.Li, "Efficient and Guaranteed Service Coverage in Partitionable Mobile Ad-hoc Networks", In *IEEE Joint Conference of Computer and Communication Societies (INFOCOM'02)* , New York City, New York, 2002, June 23-27, pp.1089-1098.
8. O.Wolfson, A.Milo, "The Multicast Policy and Its Relationship to Replicated Data Placement", *ACM Transaction on Database System*, vol.16, no.1, 1991, pp.181-205.
9. T Camp, J Boleng, v.Davies, "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, no.5, September 2002, pp. 483- 502.
10. Zhengjing, et al., "An Adaptive Replica Allocation Algorithm in MANET Environment," *Tech. Rep. PDD-2003-9*, School of computer, National University of Defense Technology, 2003.