REPLICA ALLOCATION CONSIDERING DATA UPDATE INTERVALS IN AD HOC NETWORKS

Hideki Hayashi, Takahiro Hara, and Shojiro Nishio

Dept. of Multimedia Eng., Grad. Sch. of Information Science and Tech., Osaka Univ.
{hideki, hara, nishio}@ist.osaka-u.ac.jp

Abstract In ad hoc networks, since network division occurs frequently, it is effective to replicate data items. This paper proposes effective replica allocation methods in ad hoc networks where each data item is updated at irregular intervals. The proposed methods allocate replicas based on probability density functions of the update intervals of data items. Also, they invalidate replicas that have been updated with high probability because accesses to old replicas impose extra computational overhead and roll backs. As a result, the proposed methods not only improve data accessibility but also reduce the number of accessing old replicas.

Keywords: ad hoc networks, replica allocation, data accessibility, data update, mobile computing environment

1. Introduction

As one of the research fields in mobile computing environments, there has been increasing interest in *ad hoc networks* that are constructed of only mobile hosts that play the role of a router. Disconnections occur frequently in ad hoc networks, since mobile hosts move freely, and this causes frequent network division. If network division occurs due to the migration of mobile hosts, mobile hosts in one of the two divided networks cannot access data items held by mobile hosts in the other network. In Figure 1, if the radio link between two mobile hosts is disconnected at the central part, the mobile hosts on the left-hand side and those on the right-hand side cannot access data items D_1 and D_2, respectively. A key solution to this problem is to replicate data items on mobile hosts that are not the owners of the original data item.

In ad hoc networks, there also be many applications in which mobile hosts access data held by other mobile hosts. A good example is when a research project team constructs an ad hoc network and the team members refer to data obtained by other members for efficiency. Recently, ad hoc networks have attracted much attention as an infrastructure of next-generation computer environments, e.g., wearable computing en-

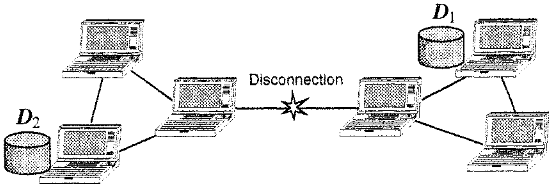

Figure 1. Network division.

vironments and sensor networks. Therefore, it will be more and more important to improve data accessibility in ad hoc networks.

In [2, 4], we proposed three replica allocation methods for improving data accessibility in ad hoc networks where a data item is not updated. These methods periodically determine replica allocation based on the access frequency from each mobile host to each data item and the network topology at that moment. In [3], we extended the three methods proposed in [2] to adapt to an environment where each data item is periodically updated. These extended methods replicate data items on mobile hosts based on the access frequency, the time remaining until each item is next updated, and the network topology.

In a real environment, it is more likely that data items are updated at irregular intervals. In this paper, we propose three replica allocation methods to improve data accessibility based on probability density functions of the update intervals of data items. In an assumed environment, mobile hosts may access invalid replicas that have been updated. Such invalid accesses cause roll backs when the hosts later connect to the mobile hosts holding the originals. Invalid accesses and roll backs consume the power of mobile hosts, causing a serious problem since mobile hosts typically have poor power resources. Thus, the proposed methods also invalidate replicas that have been updated with high probability. As a result, the proposed methods not only improve data accessibility but also reduce the number of accessing invalid replicas.

The remainder of the paper is organized as follows. In Section 2, we show some conventional works related to our work. In Section 3, we present our assumed environment. In Section 4, we propose three replica allocation methods. In Section 5, we show the results of simulation experiments. Finally, in Section 6, we summarize this paper.

2. Related Works

In the research field of ad hoc networks, a few studies have been made to improve data accessibility [7–12].

In [7] and [9], the authors have proposed methods in which replicas are allocated to a fixed number of mobile hosts which act as servers and the consistency among the replicas is keep using a strategy based on the quorum system which has been proposed for distributed database. These are considered similar to our methods because replicas are allocated to mobile hosts. However, these methods assume that replicas are allocated to only mobile hosts selected as servers and their storages have unlimited memory space. Our methods effectively allocate replicas to all mobile hosts with limited memory space.

In [8, 11], the authors have proposed caching methods in the Internet based mobile ad hoc network. This method is considered similar to our methods because each mobile host allocates replicas. However, this method differs from our methods with the point that the authors assume that some of the mobile hosts are connected to the Internet. We assume that all mobile hosts are not connected to the Internet.

In [10], the authors have defined two new consistency levels among replicas and proposed methods that disseminate updated data to keep the consistency. These methods are considered similar to our methods because replicas are allocated to all mobile hosts. However, these methods are different from our methods in which each mobile host invalidate replicas considering update intervals of the data items.

In [12], the authors have proposed a method that predicts time when a network division occurs and allocates replicas to mobile hosts before the network division occurs. This is similar to our proposed methods because the authors assume frequent network divisions. However, this method differs from our methods with the point that the authors assume a specific mobility model. Our methods work for any mobility models.

3. Assumptions and Approach

The system environment is assumed to be an ad hoc network where mobile hosts access data items held by others. In this paper, mobile hosts connected to each other by one-hop/multihop links are simply called *connected mobile hosts.* We make the following assumptions:

- We assign a unique *host identifier* to each mobile host. The set of all mobile hosts in the system is denoted by $\boldsymbol{M} = \{M_1, M_2, \cdots, M_m\}$, where m is the total number of mobile hosts and M_j $(1 \leq j \leq m)$ is a host identifier. Each mobile host moves freely.

- Data are handled as a data item which is a collection of data. We assign a unique *data identifier* to each data item. The set of all data items is denoted by $\boldsymbol{D} = \{D_1, D_2, \cdots, D_n\}$, where n is the total number of data items and D_j $(1 \leq j \leq n)$ is a data identifier. The original of each data item is held by a particular mobile host.

- Each mobile host has memory space of C data items for creating replicas, excluding the space for the original data item. Replicas are relocated in a specific period called *relocation period.*
- The access frequencies to data items from each mobile host are known. In a real environment, the access frequencies can usually be known by recording the log of access requests at each host.
- Each data item is updated by the mobile host holding the original at irregular intervals. The update interval of D_j is represented by a probability density function, $f_j(t)$. After a data item is updated, the replicas become invalid. In a real environment, $f_j(t)$ can usually be known by recording the log of the update at mobile hosts holding the originals.
- Each mobile host holds a table in which the information on the update times (time stamps) of all data items in the entire network is recorded. This information table is called the *time stamp table.*

In this environment, a request for a data item is successful only when the request issuing host accesses the original or its replica with the same time stamp as the original. That is, replicas with a time stamp different from the original are invalid. The request succeeds immediately if the request issuing host holds the original or connects to the mobile host holding the original. Otherwise, if the request issuing host or at least one connected mobile host holds the replica, the request issuing host tentatively accesses the replica. After the tentative access, when the request issuing host connects to the host holding the original, the tentative access is determined as having either succeeded or failed. This can be achieved by comparing the update logs at the host holding the original with the information on the time stamp of the accessed replica and the access time at the request issuing host. If the tentative access fails, the roll back occurs so that the request issuing mobile host recovers its state before accessing the replica. If the request issuing host and connected mobile hosts do not hold the original/replicas, the request fails.

4. Replica Allocation Methods

In this section, we first propose three replica allocation methods. Then, we explain the cache invalidation in our proposed methods.

4.1 Replica allocation

In this paper, we propose three new replica allocation methods, which are extensions of the methods proposed in [3], to adapt to an environment where data items are updated at irregular intervals.

First, we define a *PTT value* for each replica of D_j at M_i as follows:

$$p_{ij} \cdot \int_0^{to_j - t_j} f_j(t + t_j) \cdot t\,dt \tag{1}$$

Here, p_{ij} denotes the access frequency of M_i to D_j; to_j denotes the lifetime of replicas of D_j ($to_j \geq 0$); t_j denotes the time that has passed since D_j's most recent update, which is the time interval between the current time and the time stamp of D_j. If D_j has already timed out ($to_j < 0$), the PTT value is defined as 0.

The PTT value represents the average number of successful access requests until the mobile host discards the replica of D_j. Thus, by allocating replicas with high PTT values at a relocation period, not only data accessibility is expected to be higher but also the number of accessing invalid replicas is expected to be lower. The detailed decision process of to_j is explained in the next subsection.

Based on this idea, we extend the three methods proposed in [3] by mainly changing their algorithms to use PTT values instead of PT values in order to adapt to an environment where each data item is updated at irregular intervals. We call the three extended methods the *E-SAF (Extended-Static Access Frequency)α* method, the *E-DAFN (Ex.-Dynamic Access Frequency and Neighborhood)α* method, and the *E-DCG (Ex.-Dynamic Connectivity based Grouping)α* method. In the following, we describe the details of the three extended methods.

4.1.1 E-SAFα method. Each mobile host allocates replicas of data items in descending order of PTT values within the limit of its own memory space. If a mobile host issues an access request for a data item whose replica at the host has become invalid and the request is satisfied, the request issuing host again allocates the valid replica, i.e., refreshes the replica. This operation is also done in the other two methods.

4.1.2 E-DAFNα method. In the E-SAFα method, since mobile hosts with the same access characteristic allocate the same replicas and there are many replica duplications, the data accessibility is low. To solve this problem, after allocating replicas with the E-SAFα method, the E-DAFNα method eliminates replica duplications between two neighboring mobile hosts. The algorithm is as follows:

1 At a relocation period, each mobile host broadcasts its host identifier and information on access frequencies to data items. After all mobile hosts complete their broadcasts, from the received host identifiers, every host knows its connected mobile hosts.

2 Each mobile host preliminary determines the replica allocation with the E-SAFα method.

3 In each set of connected mobile hosts, starting from the mobile host with the lowest suffix (i) of host identifier (M_i), the following procedure is repeated in the order of the breadth first search. When there is duplication of a data item (original/replica) between two neighboring mobile hosts, and if one of them is the original, the host which holds the replica replaces it with another replica. If both of them are replicas, the host with the lower PTT value replaces the replica with another replica. When replacing the replica, from among data items whose replicas are not allocated at either of the two hosts, a different replicated data item is selected whose PTT value is the highest.

4.1.3 E-DCGα method. The E-DCGα method shares replicas in larger groups of mobile hosts than the E-DAFNα method. This method creates groups of mobile hosts as *biconnected components* [1] and then allocates replicas of data items on mobile hosts in each group in descending order of PTT values in the group. By grouping mobile hosts as a biconnected component, the group is not divided even if one mobile host disappears from the network or one link is disconnected in the groups. Thus, the group has high stability. The algorithm is as follows:

1 At a relocation period, each mobile host broadcasts its host identifier and information on access frequencies to data items. After that, every host knows its connected mobile hosts.

2 In each set of connected mobile hosts, starting from the mobile host with the lowest suffix (i) of host identifier (M_i), an algorithm to find biconnected components is executed. Then, each biconnected component is put in a group. If a mobile host belongs to more than one biconnected component, it can belong only to the group in which the corresponding biconnected component was found earlier.

3 In each group, the PTT value of each mobile host in the group to each item is calculated. Then, the PTT value of the group to each item is calculated as a summation of PTT values of mobile hosts in the group to the item. These calculations are done by the mobile host with the lowest suffix of host identifier in the group.

4 In descending order of PTT values in each group, replicas are allocated until the memory space of all mobile hosts in the group becomes full. Here, replicas of data items which are held as originals by mobile hosts in the group are not allocated. Each replica

is allocated at a mobile host whose PTT value to the data item is the highest among hosts that have free memory space to create it.

5 After allocating replicas of all data items that have no original in the group, if there is still free memory space at mobile hosts in the group, replicas are allocated in descending order of PTT values until the memory space is full. Each replica is allocated at a mobile host with the highest PTT value to the data item among hosts that have free memory space and do not hold the original/replica. If there is no such mobile host, the replica is not allocated.

4.2 Cache invalidation

In our proposed methods, a lifetime is assigned to each data item based on its probability density function of update intervals. Since each mobile host discards replicas whose most recent update is a long time ago, the number of accesses to invalid replicas can be reduced. In the following, we explain the procedure of cache invalidation in our methods.

The probability that D_j is updated within time τ_j since D_j's most recent update is expressed by the following expression:

$$\int_0^{\tau_j} f_j(t)\,dt. \tag{2}$$

In our proposed methods, a constant threshold α is assigned to all data items in the entire network. If the value of expression (2) becomes equal or more than α, each mobile host holding a replica of D_j discards the replica from its own cache. More specifically, when each mobile host allocates a replica of D_j or records the time stamp of D_j, the value of τ_j that satisfies the condition $\int_0^{\tau_j} f_j(t)\,dt = \alpha$ is calculated using expression (2). The value of τ_j calculated in this manner is defined as the lifetime, to_j. Then, using expression (2) and the found $\tau_j (= to_j)$, the PTT value is calculated at every relocation period.

In order to reduce the number of accessing invalid replicas, each mobile host monitors its own cache space and discards replicas whose lifetimes have passed. The cache space for the discarded replicas is kept free. When the mobile host accesses the original or a valid replica, the new replica of the data item is allocated again on the free cache space

At the time of discarding replicas, each mobile host discards the information on time stamps of replicas whose lifetimes have passed from its own time stamp table regardless of whether it holds the replica. When it accesses the original or a valid replica, it again records the information on the time stamp of the replica in its own time stamp table.

If α is set to a small value, each mobile host discards replicas whose originals have been updated with low probabilities, and thus the number of accesses to invalid replicas is reduced. However, since a small value of α also discards many valid replicas, the number of successful accesses is also reduced. Thus, the value of α should be chosen carefully according to the system characteristics and the performance requirements.

5. Simulation Experiments

In this section, we present simulation results from our performance evaluation of the proposed methods.

5.1 Simulation model

The number of mobile hosts in the entire network is 40. Mobile hosts exist in a size 500 [m] × 500 [m] flatland and randomly move in all directions at a speed randomly determined from 0 to 1 [m/sec]. The radio communication range of each mobile host is 80 [m]. The number of kinds of data items in the entire network is 40, and M_i holds D_i ($i = 1, \cdots, 40$) as the original. The size of each data item is 1 [MB]. Each mobile host creates up to 10 replicas with our proposed methods in Section 4. Replicas are periodically relocated every 400 [sec]. The access frequency of M_i to D_j is $p_{ij} = 0.5 \times (1 + 0.001j)$[1/10sec]. That is, each mobile host requests data items based on their access frequencies at every 10 [sec]. Each data item is updated with intervals based on the exponential distribution with mean U [sec].

In the simulation experiments, we randomly determine the initial position of each mobile host in the flatland and evaluate the following two criteria for our proposed methods during 1,000,000 [sec].

- *Data accessibility*:
  The ratio of the number of successful access requests to the number of all access requests issued during the simulation time.
- *Rate of accessing invalid replicas*:
  The ratio of the number of tentative data accesses that resulted in failure to the number of all access requests.

5.2 Effects of α value

First, we examine the effects of α value when the average update period U is fixed to 500. Figures 2 and 3 show the results. In both graphs, the horizontal axis indicate the value of α. The vertical axes indicate the data accessibility and the rate of accessing invalid replicas, respectively. For comparison, the performances when data replication is

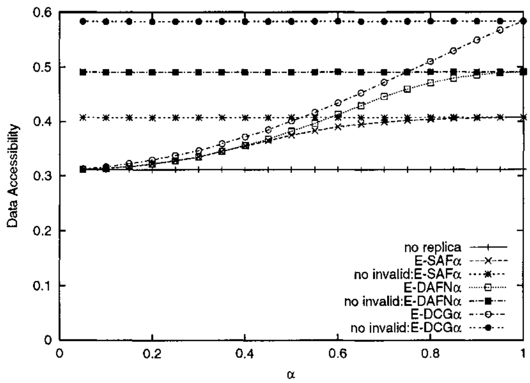

Figure 2. α and data accessibility.

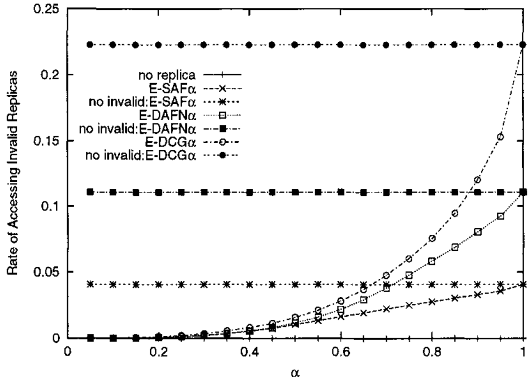

Figure 3. α and the rate of accessing invalid replicas.

not made and when replica invalidation is not made are shown as "no replica" and "no invalid:(replica allocation method)," respectively.

In Figure 2, in the cases where each mobile host invalidates replicas, as α gets larger, the data accessibility of each method gets higher. This is because the lifetime of each data item gets longer, and thus a larger number of replicas are available in the entire network. In the cases where each mobile host does not invalidate replicas, the data accessibility of each method is not affected by α. Comparing the three methods, the E-DCGα method gives the highest data accessibility. This is because mobile hosts can share many kinds of data items in stable groups.

Figure 3 shows that in the cases where each mobile host invalidates replicas, mobile hosts rarely access invalid replicas when α is very small. This is because the lifetime is very short, and thus replicas are discarded in a short time. On the contrary, as α gets larger, the rate of accessing

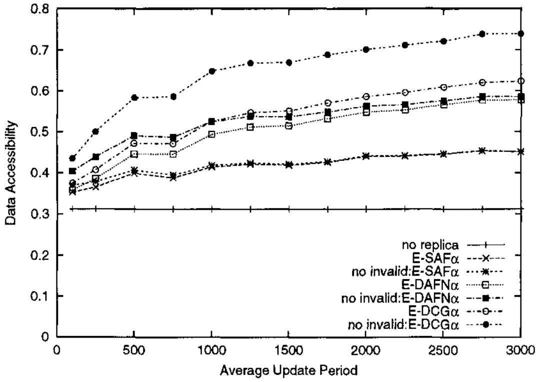

Figure 4. Average update period and data accessibility.

invalid replicas gets higher. This is because while many valid replicas are accessible in the entire network as shown in Figure 3, many invalid replicas are also accessible. Comparing the three replica allocation methods, the E-DCGα method always gives the highest rate of accessing invalid replicas. This is due to the same reason given in the case of Figure 2.

The above results show that, in this experiment, the increment of data accessibility when α gets larger is larger than that of rate of accessing invalid replicas. However, since invalid accesses and roll backs impose extra power consumption, it is meaningful to reduce the number of accessing invalid replicas. We have conducted other simulation experiments where the average update period is changed. The results show that when the average update period is long, the increment of rate of accessing invalid replicas is larger than the case in this subsection.

5.3 Effects of average update period

Next, we examine the effects of the average update period, U, on our proposed methods when α is fixed to 0.7. Figures 4 and 5 show simulation results. In both graphs, the horizontal axis indicates the average update period. The vertical axes indicate the data accessibility and the rate of accessing invalid replicas, respectively.

In Figure 4, as the average update period gets longer, the data accessibility gets higher in all cases. This is because replicas held by each mobile host are valid for a longer time. In each replica allocation method, the data accessibility in the case where each mobile host does not invalidate replicas is larger than that in the case where it invalidates replicas. This is because when each mobile host discards replicas whose lifetimes have passed, it may discard valid replicas that have not yet been updated.

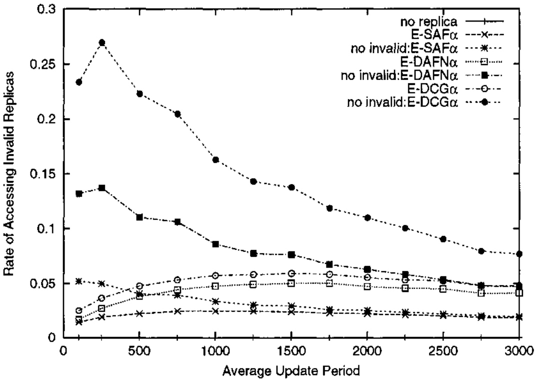

Figure 5. Average update period and the rate of accessing invalid replicas.

Figure 5 shows that in the cases where mobile hosts invalidate replicas, when the average update period is very short, the rate of accessing invalid replicas is very low. In these cases, since replicas are discarded in a very short time, each mobile host can create replicas only when connecting with mobile hosts holding the originals in a relocation period. As a result, it creates only replicas with the same version as the originals and thus rarely accesses invalid replicas. In the cases where each mobile host does not invalidate replicas, as the average update period gets longer, the rate of accessing invalid replicas of all methods gets lower. This is because replicas held by each mobile host are valid for a longer time. The above results show that invalidation of replicas is very effective in each method when the average update period is short. However, when the average update period is long, the effectiveness becomes low since replicas are valid for a long time.

6. Conclusions

In this paper, we proposed effective replica allocation methods in ad hoc networks where each data item is updated at irregular intervals. The proposed methods allocate replicas based on probability density functions of the update intervals of data items. Furthermore, in the proposed methods, each mobile host discards replicas that are updated with high probability. As a result, the proposed methods not only improve data accessibility but also reduce the number of accessing old replicas.

The results of simulation experiments according to our proposed methods show that data accessibility and the rate of accessing invalid replicas significantly depend on the setting of threshold α.

In [5], we proposed cache invalidation methods to effectively invalidate old replicas by broadcasting invalidation reports. In [6], we also proposed updated data dissemination methods to update old replicas effectively. As part of our future work, we plan to evaluate our methods in an environment where methods proposed in [5] or [6] are used together.

Acknowledgments

This research was partially supported by The 21st Century Center of Excellence Program "New Information Technologies for Building a Networked Symbiotic Environment" and Grant-in-Aid for Young Scientists (A)(1668005) of the Ministry of Education, Culture, Sports, Science and Technology, Japan, and by Tateishi Science and Technology Foundation.

References

[1] A.V. Aho, J.E. Hopcroft, and J.D. Ullman, "The Design and Analysis of Computer Algorithms," *Addison-Wesley*, 1974.

[2] T. Hara, "Effective replica allocation in ad hoc networks for improving data accessibility," *Proc. IEEE Infocom'01*, pp. 1568–1576, 2001.

[3] T. Hara, "Replica allocation methods in ad hoc networks with data update," *ACM-Kluwer Journal on Mobile Networks and Applications (MONET)*, vol. 8, no. 4, pp. 343–354, 2003.

[4] T. Hara, N. Murakami, and S. Nishio, "Replica allocation for correlated data items in ad-hoc sensor networks," *SIGMOD Record*, vol. 33, no. 1, pp. 38–43, 2003.

[5] H. Hayashi, T. Hara, and S. Nishio, "Cache invalidation for updated data in ad hoc networks," *Proc. CoopIS'03*, pp. 516–535, 2003.

[6] H. Hayashi, T. Hara, and S. Nishio, "Updated data dissemination in ad hoc networks," *Proc. Int'l Workshop on Ubiquitous Mobile Information and Collaboration Systems (UMICS'04)*, pp. 29–43, 2004.

[7] G. Karumanchi, S. Muralidharan, and R. Prakash, "Information dissemination in partitionable mobile ad hoc networks," *Proc. Symposium on Reliable Distributed Systems (SRDS'99)*, pp. 4–13, 1999.

[8] S. Lim, W-C. Lee, G. Cao, and C.R. Das, "A novel caching scheme for internet based mobile ad hoc networks," *Proc. ICCCN'03*, pp. 38–43, 2003.

[9] J. Luo, J.P. Hubaux, and P. Eugster, "PAN: Providing reliable storage in mobile ad hoc networks with probabilistic quorum systems," *Proc. ACM MobiHoc'03*, pp. 1–12, 2003.

[10] K. Rothermel, C. Becker, and J. Hahner, "Consistent update diffusion in mobile ad hoc networks," *Technical Report 2002/04, Computer Science Department, University of Stuttgart*, 2002.

[11] F. Sailhan, and V. Issarny, "Cooperative caching in ad hoc networks," *Proc. Int'l Conf. on Mobile Data Management MDM'03*, pp. 13–28. 2003.

[12] K. Wang, and B. Li, "Efficient and guaranteed service coverage in partitionable mobile ad-hoc networks," *Proc. IEEE Infocom'02*, vol. 2, pp. 1089–1098, 2002.