01 Jan 2006

Data Replication for Improving Data Accessibility in Ad Hoc Networks

Sanjay Kumar Madria
Missouri University of Science and Technology, madrias@mst.edu

Takahiro Hara

Recommended Citation

S. K. Madria and T. Hara, "Data Replication for Improving Data Accessibility in Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, Institute of Electrical and Electronics Engineers (IEEE), Jan 2006. The definitive version is available at https://doi.org/10.1109/TMC.2006.165

Data Replication for Improving Data Accessibility in Ad Hoc Networks

Takahiro Hara, *Member*, *IEEE*, and Sanjay K. Madria, *Senior Member*, *IEEE*

Abstract—In ad hoc networks, due to frequent network partition, data accessibility is lower than that in conventional fixed networks. In this paper, we solve this problem by replicating data items on mobile hosts. First, we propose three replica allocation methods assuming that each data item is not updated. In these three methods, we take into account the access frequency from mobile hosts to each data item and the status of the network connection. Then, we extend the proposed methods by considering aperiodic updates and integrating user profiles consisting of mobile users' schedules, access behavior, and read/write patterns. We also show the results of simulation experiments regarding the performance evaluation of our proposed methods.

Index Terms—Ad hoc networks, replication scheme, data accessibility, data update.

1 Introduction

Recently, there has been an increasing interest in *ad hoc networks*, which are constructed by mobile hosts [2], [5]. In ad hoc networks, every mobile host plays the role of a router, and they communicate with each other. Even if the source and the destination mobile hosts are not in the communication range of each other, data packets are forwarded to the destination mobile host by relaying transmission through other mobile hosts which exist between the two mobile hosts. Since no special infrastructure is required, in various fields such as military and rescue affairs, many applications are expected to be developed for ad hoc networks.

In ad hoc networks, since mobile hosts move freely, disconnections occur frequently, and this causes frequent network partition. If a network is partitioned into two networks due to the migrations of mobile hosts, mobile hosts in one of the partitions cannot access data items held by mobile hosts in the other. Thus, data accessibility in ad hoc networks is lower than that in conventional fixed networks. In ad hoc networks, it is very important to prevent the deterioration of data accessibility at the point of network partition. A possible and promising solution is the replication of data items at mobile hosts which are not the owners of the original data.

Since mobile hosts generally have poor resources, it is usually impossible for them to have replicas of all data items in the network. For example, let us suppose a situation where a research project team engaged in excavation work constructs an ad hoc network on a mountain. The results obtained from the investigation may consist of various types of data such as numerical data, photographs, sounds, and videos. In this case, although it is useful to have the data that other members obtained, it seems difficult for a mobile host to have replicas of all the data.

In this paper, we assume an environment where each mobile host has limited memory space for creating replicas and each data item is not updated. First, we propose three replica allocation methods for improving data accessibility. Then, we extend these three methods by considering aperiodic data updates since, in a real environment, updates do occur aperiodically. These extended methods also accommodate user schedule and emergency objects. In this paper, a mesoscale ad hoc network consisting of a few dozen mobile hosts is assumed, and our proposed methods are mainly designed for such a mesoscale network. We verify the effectiveness of our proposed methods by simulation experiments. Note that the mobile hosts that are connected to each other by one-hop or multihop wireless links are simply called connected mobile hosts. That is, connected mobile hosts do not only represent those connected by one-hop links (i.e., neighbors), but also represent those connected by multihop links.

The remainder of this paper is organized as follows: In Section 2, we introduce some related works. In Section 3, we propose replica allocation methods. In Section 4, we extend the methods proposed in Section 3 to adapt to environments with data update. In Section 5, we show the results of simulation experiments. In Section 6, we discuss how we can reduce the number of accesses to old replicas. Finally, in Section 7, we summarize this paper. We note that some of the results of this paper have been reported in [8] and [10].

2 Related Works

Many routing protocols in ad hoc networks have been proposed in IETF (Internet Engineering Task Force) and other research groups [16], [20], [21], [22]. These protocols improve connectivity among mobile hosts at the network level. Thus, these protocols are useful for applications where users equipped with mobile hosts directly communicate

- *T. Hara is with the Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan. E-mail: hara@ist.osaka-u.ac.jp.*
- *S.K. Madria is with the Computer Science Department, University of Missouri-Rolla, 325 Computer Science Building, 1870 Miner Circle, Rolla, MO 65409-0350. E-mail: madrias@umr.edu.*

Manuscript received 6 Aug. 2005; revised 13 Jan. 2006; accepted 26 Jan. 2006; published online 15 Sept. 2006.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0234-0805.

with each other, e.g., video conference systems. However, in ad hoc networks, there are also other applications where mobile hosts access data items held by other mobile hosts. In these applications, the above protocols can only improve the connectivity among mobile hosts which are connected to each other, but cannot do anything when the network is partitioned. On the other hand, our proposed replica allocation methods specialize in such applications and appropriately allocate replicas of data items before a network partition. Thus, our proposed methods can improve data accessibility.

In the research field of databases, many strategies for data replication have been proposed [6], [26], [27], [29]. In distributed database systems, data replication offers the benefits of shortening response time for database operations and improving data availability. In traditional systems, shortening response time is usually considered the most important issue and, thus, most replication strategies address this issue. This issue includes the propagation of update operations to replicas. The latter is achieved by replicating databases and using the replicas when the site which holds the original database fails. This approach is considered to be similar to our approach because both approaches address improving data availability (accessibility). However, since system failures do not frequently occur in distributed database systems in fixed networks, it is usually sufficient to create a few replicas of a database. On the other hand, since frequent partitioning of the network is a special characteristic in ad hoc networks, our approach takes it into account and is completely different from that in distributed database systems. Shortening the response time and propagating update operations are also significant issues in ad hoc networks. We plan to address them in our future work.

In the research field of mobile computing, several strategies for replicating or caching data have been proposed so far [3], [14], [15], [23], [30]. Most of these strategies assume an environment where mobile hosts access databases at sites in a fixed network and replicate/cache data on the mobile hosts because wireless communication is more expensive than wired communication. They address the issue of keeping consistency between the original data and its replicas or cached data with low communication costs. These strategies assume only one-hop wireless communication and, thus, they are completely different from our approach, which assumes multihop communication in ad hoc networks.

A few studies have been conducted to improve data accessibility by replicating or caching data in ad hoc networks [7], [17], [24], [25], [28]. In [7], the authors discussed the issues associated with data communication with database systems in ad hoc networks. However, they did not propose a concrete algorithm for replica management. In [17], the authors proposed a few consistency management methods based on the quorum system that has been proposed for distributed database systems. In [24], the authors defined two new consistency levels among replicas and proposed methods that disseminate updated data to maintain consistency among replicas. In [28], the authors proposed a method for predicting the time when a network partition occurs in ad hoc networks and creates replicas at each mobile host before the network partition. These are all considered similar to our methods because data items are replicated or cached on mobile hosts in an ad hoc network. However, these methods assume that replicas are allocated to mobile hosts holding unlimited memory space, whereas our methods effectively improve data accessibility in an environment where there are mobile hosts with limited memory space. In [25], the authors proposed a method that cooperatively caches Web data in ad hoc networks. This method does not take into account the network topology and its dynamic change.

To the best of our knowledge, there have never been studies on replicating data items on mobile hosts holding limited memory space in ad hoc networks aside from our previous work [8], [9], [10], [11], [12], [13]. In [9], we discussed data replication in an environment where each data item is aperiodically updated. This is considered similar to our approach in this paper because both approaches aim to replicate data items by considering aperiodic updates. In [9], we assumed that the probability that an update to each data item occurs within an interval since the previous update is represented by a certain probability-density function and the host holding the original data precisely knows this function. However, this assumption is usually unrealistic because it is very difficult to detect specific characteristics of update occurrences even if each host holding an original records the log of the update occurrences. On the contrary, the extended methods in this paper use only the ratio of read to write probabilities; this is easy to calculate from the log of read and write occurrences.

3 Replica Allocation Methods in Ad Hoc Networks

In this section, the basic system model is described. Then, we propose replica allocation methods for improving data accessibility in ad hoc networks.

3.1 System Model

The system environment is assumed to be a mesoscale mobile ad hoc network in which mobile hosts (users) collaboratively share data items for the same purpose such as rescue and military affairs and shopping in a shopping mall. A few dozen mobile hosts exist in the ad hoc network and they access data items held by other mobile hosts as the originals. Each mobile host creates replicas of the data items and maintains the replicas in its memory space. A good example of data items shared among mobile hosts (users) is the information on the work progress and the task results of the users who are engaged in the collaborative work. Each host holds the information on its own work progress and the task results as the original, and replicates that of other users. These replicas are used for efficient collaborative work, where data access requests do not have to, but are preferred to, succeed. Therefore, the main objective of data replication is the improvement of data accessibility.

In our proposed replica allocation methods, there is no central server that determines the allocation of replicas, but

mobile hosts autonomously determine the allocation in a distributed manner. Some of our proposed replica allocation methods need a mobile host as the coordinator which is chosen dynamically. Here, since we assume a mesoscale ad hoc network, data broadcasts of small messages such as access requests, information about access frequencies, and schedules of mobile hosts do not cause broadcast storms. More specifically, we neglect the effects of broadcasting messages in the network because their sizes are much smaller than the sizes of data items.

When a mobile host issues an access request to a data item, the request is successful if either 1) the mobile host holds the original/replica of the data item or 2) at least one mobile host which is connected to the requesting host with a one-hop/multihop link holds the original/replica. Thus, the requesting host checks whether or not it holds the original/replica of the target data item. If it does, the request succeeds on the spot. If it does not, it broadcasts the request of the target data item to connected mobile hosts. Then, if it receives a reply from another host(s) which holds the original/replica of the target data item, the request is also successful. Otherwise, the request fails. Here, it should be noted that in an environment where each data item is updated, data accesses to replicas remain tentative until the requesting hosts connect to the hosts holding the original items. If the tentative accesses lead to dirty reads of old replicas, the accesses result in aborts. If not, the accesses succeed. Mobile hosts that hold old replicas refresh the replicas as needed when they connect to the hosts holding the originals.

In this system environment, we also make the following assumptions:

- We assign a unique *host identifier* to each mobile host in the system. 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.
- Each mobile host, M_i, has a memory space of C_i data items for creating replicas excluding the space for the original data items that the host holds.
- The access frequencies to data items from each mobile host are known and do not change.
- Data is handled as a data item, which is a collection of data. We assign a unique *data identifier* to each data item located in the system. 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. All data items are of the same size and each data item is held by a particular mobile host as the original. The assumption that all data items are of the same size is only for the purpose of simplicity, but not too strong or unrealistic. We can easily consider environments where data items have different sizes. A simple and popular way is to consider different sizes of data items when replacing replicas. For example, if the criteria for replica replacement is the access frequency, a data item with access frequency p and data size s is given priority according to p/s.

3.2 Replica Allocation Methods in an Environment without Update

In this section, we assume an environment where each data item is not updated. Thus, the following assumption is added to those mentioned in the previous section:

- The data items are not updated. In the above mentioned example of a digging investigation, new data is inserted but no updates occur. There are also other applications where the newness of data is not a serious problem, such as in weather information. It is also assumed that the data items are not deleted.

In order to determine the optimal assignments among all possible combinations of replica allocation, we must analytically find a combination which gives the highest data accessibility considering the following parameters:

1. the access frequency from each mobile host to each data item,
2. the probability that each mobile host will participate in the network and will disappear from the network,
3. the probability that two mobile hosts connected by a (one-hop) link will be disconnected, and
4. the probability that two disconnected mobile hosts will be connected by a one-hop link.

Even if some lopping is possible, the computational complexity is very high, and this calculation must be done every time the network topology changes due to mobile host migration. Moreover, among the above four parameters, the three latter ones cannot be formulated in practice because mobile hosts move freely. For these reasons, we take the following heuristic approach:

- Replicas are relocated in a specific period (*relocation period*).
- During every relocation period, replica allocation is determined based on the access frequency from each mobile host to each data item and, optionally, the network topology at the moment.

Based on this approach, we propose three replica allocation methods which differ in the emphasis put on access frequency and network topology:

1. The *SAF (Static Access Frequency)* method: Only the access frequency to each data item from each host is taken into account.
2. The *DAFN (Dynamic Access Frequency and Neighborhood)* method: The access frequency to each data item and the neighborhood among mobile hosts are taken into account.
3. The *DCG (Dynamic Connectivity based Grouping)* method: The access frequency to each data item and the whole network topology are taken into account.

3.2.1 The SAF Method

In the SAF method, each mobile host M_i allocates replicas of C_i data items in descending order of its own access frequencies to the data items. That is, in this method, each host allocates replicas for its selfish end and does not take into account which data items are replicated by other hosts.

TABLE 1
Access Frequencies to Data Items

Data	Mobile host					
	M_1	M_2	M_3	M_4	M_5	M_6
D_1	0.65	0.25	0.17	0.22	0.31	0.24
D_2	0.44	0.62	0.41	0.40	0.42	0.46
D_3	0.35	0.44	0.50	0.25	0.45	0.37
D_4	0.31	0.15	0.10	0.60	0.09	0.10
D_5	0.51	0.41	0.43	0.38	0.71	0.20
D_6	0.08	0.07	0.05	0.15	0.20	0.62
D_7	0.38	0.32	0.37	0.33	0.40	0.32
D_8	0.22	0.33	0.21	0.23	0.24	0.17
D_9	0.18	0.16	0.19	0.17	0.24	0.21
D_{10}	0.09	0.08	0.06	0.11	0.12	0.09

At the time of replica allocation, a mobile host may not be connected (have a path) to another mobile host which has an original or a replica of a data item that the host should allocate. In this case, the memory space for the replica is retained. The replica is created when a data access to the data item succeeds or when the mobile host connects to another mobile host which has the original or a replica at a later relocation period.

Now, let us suppose that six mobile hosts ($M_1, \cdots, M_6$) exist and M_i $(i = 1, \cdots, 6)$ holds D_i as an original. The access frequency from each mobile host to each data item is shown in Table 1. Fig. 1 shows the result of executing the SAF method. In this figure, a circle denotes a mobile host, a straight line denotes a wireless link, a gray rectangular denotes an original data item, and a white rectangular denotes a replica allocated.

In the SAF method, mobile hosts do not need to exchange information with each other for replica allocation. Moreover, replica relocation does not occur after every mobile host allocates all necessary replicas. As a result, this method allocates replicas with low overhead and low traffic. On the other hand, since each mobile host allocates replicas based only on its access frequencies to data items, mobile hosts with the same access characteristics allocate the same replicas. Since a mobile host can access data items or replicas held by other connected mobile hosts, it is more effective to share many kinds of replicas among them. Therefore, the SAF method gives low data accessibility when many mobile hosts have the same or similar access characteristics.

3.2.2 The DAFN Method

To solve the problem with the SAF method, the DAFN method eliminates the replica duplication among neighboring mobile hosts. First, this method preliminarily determines the replica allocation in the same way as the SAF method. Then, if there is replica duplication of a data item between two neighboring mobile hosts, the mobile host with the lower access frequency to the data item changes the replica to another replica. Since the neighboring status changes as mobile hosts move, the DAFN method is executed during every relocation period. The algorithm of this method is as follows:

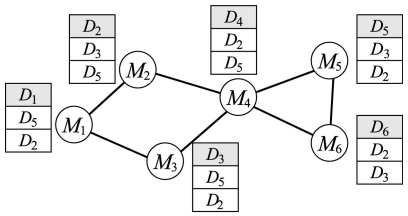


Fig. 1. An example of executing the SAF method.

1. During a relocation period, each mobile host preliminarily determines the allocation of replicas based on the SAF method.
2. Then, each mobile host broadcasts its host identifier and information on access frequencies to data items in the entire network. After all mobile hosts complete the broadcasts, from the received host identifiers, every host shall know its connected mobile hosts. Note that the network may be partitioned into several sets of connected mobile hosts.

   In each set of all the connected mobile hosts, the mobile host with the lowest suffix (i) of host identifier (M_i) becomes the first coordinator for replica relocation. Since mobile hosts know their connected hosts, the host with the lowest suffix can recognize that it has the responsibility to be the first coordinator.
3. The coordinator performs the procedure of eliminating replica duplication with its neighboring hosts that have never been coordinators. In doing so, the coordinator first sends a message to each neighbor, where it requests the information on replicas that are or were allocated to the neighbors during this relocation period. Then, from the replied information, the coordinator determines the change in replica allocation and notifies the neighbor of the result. The detailed procedure is as follows:

   When there is duplication of a data item (original/replica) between the two neighboring mobile hosts (the coordinator and its neighbor), and if one of them is the original, the host which holds the replica changes it to another replica. If both of them are replicas, the host whose access frequency to the data item is lower than the other one changes the replica to another replica. When changing the replica, among data items whose replicas are or were not allocated at either of the two hosts during this relocation period, a new data item replicated is selected where the access frequency to the item is the highest among the possible items.
4. The neighbors of the coordinator become new coordinators if they have never been coordinators and processing goes back to step 3. If all mobile hosts have been coordinators, the algorithm stops.

In the DAFN method, as shown in the above process, the procedure of eliminating replica duplication is repeated through a breadth first search from the mobile host with the lowest suffix of host identifiers and stops when all the

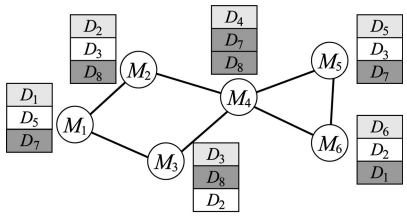


Fig. 2. An example of executing the DAFN method.

connected mobile hosts have been traversed. Thus, the execution of this method finishes in a limited time and does not cause the thrashing of the replica allocation.

After executing the above algorithm, each mobile host tries to get replicas that the host should allocate during this relocation period. This is done by requesting the replicas from the connected mobile hosts. However, at this time, the mobile host may not be connected (have a path) to another mobile host which has an original or a replica of a data item that the host should replicate. If the memory space for a replica that should be allocated but is not available at the relocation time is retained, the data accessibility in the entire network becomes low. To alleviate this problem, in the DAFN method, another replica is temporarily allocated to the memory space until the proper replica is available. This temporarily allocated replica is selected among replicas that have been allocated since the previous relocation period, but are not currently selected for allocation. This is because these replicas are available at the host and can be allocated without extra overhead. When a data access to the data item whose replica should be allocated succeeds, the memory space is filled with the proper replica.

Fig. 2 shows an example of executing the DAFN method in the environment given by Table 1 and Fig. 1. In Fig. 2, a dark gray rectangle denotes a replica which is allocated for eliminating replica duplication. In more detail, the following changes of replicas allocated occur between every combination of two neighboring mobile hosts:

$$
\begin{aligned}
&M_1\text{-}M_2:\ D_2 \rightarrow D_7\ (M_1), D_5 \rightarrow D_8(M_2)\\
&M_1\text{-}M_3:\ D_5 \rightarrow D_8\ (M_3)\\
&M_2\text{-}M_4:\ D_2 \rightarrow D_7\ (M_4)\\
&M_3\text{-}M_4:\ \text{No duplication}\\
&M_4\text{-}M_5:\ D_5 \rightarrow D_8\ (M_4)\\
&M_4\text{-}M_6:\ \text{No duplication}\\
&M_5\text{-}M_6:\ D_2 \rightarrow D_7\ (M_5), D_3 \rightarrow D_1(M_6).
\end{aligned}
$$

Here, M_i-M_j denotes that the replica duplication is eliminated between two mobile hosts M_i and M_j, and $D_k \rightarrow D_l\ (M_i)$ denotes that the replica allocation is changed from D_k to D_l at host M_i. While six kinds of replicas are allocated in the whole network in the SAF method, eight kinds of replicas are allocated in the DAFN method. This is because the DAFN method eliminates replica duplication, and thus more kinds of replicas can be shared among the neighboring mobile hosts. As a result, the data accessibility is expected to be higher than that in the SAF method.

However, the DAFN method does not completely eliminate replica duplication among neighboring hosts because it only executes the elimination process by scanning the network once based on the breadth first search. In the example in Fig. 2, we can see the replica duplication of D_8 between M_2 and M_4 and between M_3 and M_4, and of D_7 between M_4 and M_5. Moreover, if the network topology changes during the execution of this method, the replica relocation cannot be done at mobile hosts over disconnected links. Both the overhead and the traffic are higher than the SAF method because, during each relocation period, mobile hosts exchange information and relocate replicas.

3.2.3 The DCG Method

The DCG method shares replicas in larger groups of mobile hosts than the DAFN method that shares replicas among neighboring hosts. In order to share replicas effectively, each group should be stable, i.e., the group is not easily partitioned due to changes of network topology. From this viewpoint, the DCG method creates groups of mobile hosts that are *biconnected components* [1] in a network. A biconnected component denotes a maximal partial subgraph which is connected (not partitioned) if an arbitrary node in the graph is deleted. By grouping mobile hosts as biconnected components, the group is not partitioned even if one mobile host disappears from the network or one link is disconnected in the group and, thus, it is considered that the group has high stability. Here, a typical algorithm to find biconnected components from a given graph is based on the depth first search [1] and its computational complexity is $O(l)$, where l is the number of edges (wireless links) in the graph.

The DCG method is executed at every relocation period. The algorithm is as follows:

1. During a relocation period, each mobile host broadcasts its host identifier and information on access frequencies to data items in the entire network. Here, we assume that when a mobile host receives the broadcast message, it adds its own host identifier to the received message and forwards the message to its neighbors. Thus, after all mobile hosts complete the broadcasts, from the received host identifiers, every host knows the connected mobile hosts and the network topology, i.e., the graph constructed by all the connected mobile hosts (nodes) and their wireless links (edges).
2. In each set of all the connected mobile hosts, the mobile host with the lowest suffix (i) of host identifier (M_i) becomes the first coordinator. The coordinator executes an algorithm to find biconnected components within the network topology, where the coordinator becomes the root of the topology. Here, the coordinator can execute this algorithm locally (centrally) because it precisely knows the network topology from the received messages in step 1. Each biconnected component is considered as a group. If a mobile host belongs to more than one biconnected component, i.e., the host is an *articulation point*, it belongs to only one group in which the corresponding biconnected component is first found while executing the algorithm. Then, the coordinator informs the mobile host with the lowest

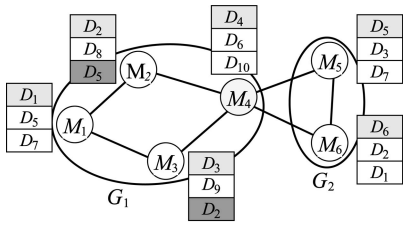


Fig. 3. An example of executing the DCG method.

suffix of host identifier in each group of the group construction result. After this, the host that is informed of the construction result becomes a new coordinator.

3. In each group, the coordinator calculates an access frequency of the group to each data item as a summation of access frequencies of mobile hosts in the group to the item. Then, the coordinator determines the replica allocation in the group by the following procedure:
   a. In the order of the access frequencies of the group, replicas of data items 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 access frequency to the data item is the highest among hosts that have free memory space to create it.
   b. After allocating replicas of all kinds of data items, if there is still free memory space at mobile hosts in the group, replicas are allocated in the order of access frequencies until the memory space is full. Each replica is allocated at a mobile host whose access frequency to the data item is the highest among hosts that have free memory space to create it and do not hold the replica or its original. If there is no such mobile host, the replica is not allocated.
4. The coordinator informs the mobile hosts in its responsible group of the replica allocation result.

Here, it should be noted that the strategy that a mobile host as an articulation point belongs to only one group in step 2 does not degrade the stability of the other groups corresponding to biconnected components that the host is included in. This is because the articulation point mobile host can relay messages and data from mobile hosts in the other groups while the host does not cooperatively allocate replicas with these hosts. That is, each of these groups is not partitioned even if one arbitrary host or one link is disconnected, i.e., the characteristic of a biconnected component is still preserved.

After executing the above algorithm, each mobile host tries to get replicas that the host should allocate during this relocation period. However, at this time, the mobile host may not be connected (have a path) to another mobile host which has an original or a replica of a data item that the host should replicate. In this case, in the same way as the DAFN method, the memory space for the replica is temporarily filled with another replica, and it is filled with the proper one when a data access to the corresponding data item succeeds.

TABLE 2
Access Frequencies of Groups

Data	Mobile host						Group	
	M_1	M_2	M_3	M_4	M_5	M_6	G_1	G_2
D_1	0.65	0.25	0.17	0.22	0.31	0.24	1.29	0.55
D_2	0.44	0.62	0.41	0.40	0.42	0.46	1.87	0.88
D_3	0.35	0.44	0.50	0.25	0.45	0.37	1.54	0.82
D_4	0.31	0.15	0.10	0.60	0.09	0.10	1.16	0.19
D_5	0.51	0.41	0.43	0.38	0.71	0.20	1.73	0.91
D_6	0.08	0.07	0.05	0.15	0.20	0.62	0.35	0.82
D_7	0.38	0.32	0.37	0.33	0.40	0.32	1.40	0.72
D_8	0.22	0.33	0.21	0.23	0.24	0.17	0.99	0.41
D_9	0.18	0.16	0.19	0.17	0.24	0.21	0.70	0.45
D_{10}	0.09	0.08	0.06	0.11	0.12	0.09	0.34	0.21

Fig. 3 shows an example of executing the DCG method in the environment given by Table 1 and Fig. 1. In this example, two groups consisting of M_1, M_2, M_3, M_4 (G_1) and M_5, M_6 (G_2) are created. Table 2 shows the access frequencies of the two groups which are calculated from Table 1. In this figure, a dark gray rectangle denotes a replica which is allocated in the second cycle. By executing the DCG method, all 10 kinds of replicas are allocated in the whole network.

Compared with the DAFN method that collaboratively replicates data items with neighboring hosts, the DCG method shares replicas in larger groups of mobile hosts which have high stability. Since many kinds of replicas can be shared, the data accessibility is expected to be higher. However, since the DCG method consists of three steps, 1) broadcasting host identifiers, 2) determining the replica allocation, and 3) notifying it to all hosts in the group, this method takes the longest time among the three methods to relocate replicas. Therefore, the probability is high that the network topology changes during the execution of this method and, in this case, the replica relocation cannot be done at mobile hosts over disconnected links. Moreover, both the overhead and the traffic are higher than the other two methods because, during each relocation period, mobile hosts exchange information and relocate replicas in a wide range.

4 Extensions Considering Data Update

The methods proposed in the previous section assume that data items are not updated. In a real environment, it is more likely that data items are randomly updated. In this section, we describe the three extended replica allocation methods that are adapted to an environment where each data item can be randomly updated.

4.1 Basic Idea

We first define the *Read/Write Ratio (RWR)* as follows:

$$RWR = R_{ij}/W_j. \quad (1)$$

Here, R_{ij} denotes the probability that an access request for data item D_j from mobile host M_i is issued at a unit of time (Read operation); W_j denotes the probability that an update for data item D_j from the mobile host who owns the original is issued at a unit of time (Write operation). RWR denotes the ratio of Read probability to Write probability for data item D_j at a unit of time. If the RWR value is higher, more Read and less Write operations have occurred for the data item at a unit time and, consequently, we should replicate the data item as the data accessibility is expected to be higher. If the RWR value is low, less Read and more Write operations have occurred for the data item, and then we should avoid the replication of data items with lower RWR values as they will be updated very often and, therefore, may be stale sooner than data items with high RWR values. Thus, replicating data items with high RWR values will allow a consistency management protocol such as lazy updates to run in the background to make the data consistent and avoid executing the consistency protocol very often at the nodes. There are many ways replicas can be made consistent; in our recent paper [12], we have explored that in a slightly different environment (Mobile P2P). The consistency management discussion is out of scope with respect to the focus of this paper.

We also use profiles [31] where users' mobility schedules, access behaviors, and read/write patterns are recorded and can actively reconfigure the replicas to adjust to the changes in user locations and system status. Thus, our algorithm can be tailored to satisfy each individual user's information requirement as closely as possible. For predicting future access patterns, each user can specify a time schedule as hints of the user's mobility pattern and data requirement. This is possible since mobile users do not move at random. They often come to a location at a predetermined time with a specific purpose in mind. Additionally, the work a user is currently engaged in has a strong relationship with the data required. This is why a user's time schedule can serve as a valuable hint for predicting the future. However, a user may not follow a schedule strictly. In such a case, we can resort to past statistics for making replication decisions.

4.1.1 User Profiles

User profiles keep schedules and maintain per-user access statistics. A schedule consists of daily activities. Each schedule entry has the time, location, and activity a user plans to do with optional data objects requirement. A mapping is performed to determine the default data requirement of a schedule entry. Considering, for example, characteristics of the user, location, and activity, can do this. The read/write histories are a user's access statistics on data objects in the system. Since a user accesses only part of the data objects at a time and location, we capture this characteristic by defining the concept of *open objects* (as in [31]) to be the set of objects accessed since now, and for a user on schedule, the set of data objects declared explicitly or implicitly in the current schedule entry. Open objects represent current and near future data access range. The information is used in making replication decisions.

Each user maintains his/her own user profile. At every relocation period, each user calculates his/her access frequencies to data items at that time based on his/her user profile and broadcasts them to connected mobile hosts. Since not all information stored in user profiles, but only access frequencies are broadcast, the network traffic caused by these exchanges is not high.

4.1.2 Emergency Events and Objects

We allow the declaration of emergency events and objects that require fast access. An emergency event is a situation that demands quick response. Such events can be identified in advanced in many application domains. An emergency event usually has a set of default information requirements that can also be identified in advance. Whenever such an event occurs, the system unconditionally replicates all emergency objects associated with that event. A user can also declare emergency objects in the schedule entry. Here, it should be noted that emergency objects are a kind of open objects.

4.2 System Model

The following assumptions are added to those mentioned in Section 3.1:

- Each data item is randomly updated. This is done by the mobile host which holds the original. After a data item is updated, all its replicas become invalid if mobile hosts holding them are not connected to the host that holds the original with one-hop/multihop links. This is because the updates cannot be propagated. This assumption represents that replicas are valid if and only if their version is same as that of the original and that all valid replicas are treated equally, i.e., there is no precedence relation among valid replicas. A possible example is a situation where each mobile host has a role of measuring some data which are useful for other mobile hosts. In case of disconnection, one can allow reading of old values. Later, when the host connects to the original holder, version numbers can be compared [19] to know whether the earlier read was dirty. In case a given application is affected by the old read, it can be aborted.
- The access frequencies to data items from each mobile host and the update frequencies of data items are known and will change. The access frequencies and update frequencies of data items can usually be known by recording the log of access requests and update operations at each host and periodically calculating the values.
- Emergency objects are always unconditionally replicated first. In terms of access frequency and access process, emergency objects are handled in the same way as normal open objects. The access frequencies of emergency objects differ with each other.

  In case there is no space to hold all the emergency objects, then the data object with the lowest RWR can be selected as the candidate to be replaced with the new one. Some other priority schemes can be used.
- Owner copies, i.e., original data items, are considered as primary copies and other copies are considered as secondary. To make the copies consistent in case of updates and disconnection, one can adopt to an existing replica consistency

protocol such as lazy replication [18]. In the case of multiple versions of replicas, hosts can read the latest version available at its own site or one of its connected hosts. Updates can be applied to only the primary copy and then propagated to secondary copies. Secondary copies can also be considered as read-only. These issues are independent of our replication scheme.

4.3 Extended Methods

Based on the basic idea and the system model, we can extend the three methods, SAF, DAFN, and DCG, by mainly changing their algorithms to use RWR values instead of access frequencies. We call the three extended methods, the *Extended SAF plus (E-SAF+)* method, the *Extended DAFN plus (E-DAFN+)* method, and the *Extended DCG plus (E-DCG+)* method, respectively.

The E-SAF+ method: In the E-SAF+ method, each mobile host, M_i, allocates replicas of C_i data items (first considering the open objects) in descending order of the RWR values.

1. During a relocation period, each mobile host that is the owner of originals broadcasts its information on write frequencies to its own data items. After all the mobile hosts complete the broadcasts, every host shall know the RWR value to each data item.
2. Based on a user's profile, if there exists an emergency object O_j which will be requested by M_i in the near future (less than the relocation period), then set $\mathrm{RWR}_{ij} = \mathrm{MAXVALUE}/\mathrm{Happen\ Time}$ (MAXVALUE is large enough to guarantee that emergency objects' RWR_{ij} values are all greater than other objects), then sort open objects by their RWR values. If the number of open objects (including emergency objects) is less than C_i, then also sort nonopen objects.
3. Each mobile host, M_i, allocates replicas of C_i data items (first considering the open objects and then considering the nonopen objects if the number of open objects is less than C_i) in descending order of the RWR values.

The E-DAFN+ method: The algorithm of the E-DAFN+ method is as follows:

1. During a relocation period, each mobile host broadcasts its host identifier and information on write frequencies to data items. After all mobile hosts complete the broadcasts, from the received information, every host shall know its connected mobile hosts and their RWR values to each data item.

   In each set of connected mobile hosts, the mobile host with the lowest suffix (i) of host identifiers (M_i) becomes the first coordinator for replica relocation.
2. Each mobile host, M_i, preliminarily allocates replicas of C_i data items (first considering the open objects and then considering the nonopen objects if the number of open objects is less than C_i) in descending order of the RWR values.
3. The coordinator performs the procedure of eliminating replica duplication with its neighboring hosts that have never been coordinators. The detailed procedure is same as that of the DAFN method, except that RWR values are used for replica relocation instead of access frequencies.
4. The neighbors of the coordinator become new coordinators if they have never been coordinators and processing goes back to step 3. If all mobile hosts have been coordinators, the algorithm stops.

The E-DCG+ method: The algorithm of the E-DCG+ method is as follows:

1. During a relocation period, each mobile host broadcasts its host identifier and information on write frequencies to data items. After all mobile hosts complete the broadcasts, from the received host identifiers, every host shall know the connected mobile hosts, their RWR values for each data item, and the network topology.
2. In each set of connected mobile hosts, the mobile hosts are clustered into groups of biconnected components in the same way as the DCG method. The mobile host with the lowest suffix of host identifier in each group becomes the coordinator of the group.
3. In each group, the coordinator calculates the RWR value of the group to each data item as a summation of RWR values of mobile hosts in the group to the item. Then, the coordinator determines the replica allocation in the group by the following procedure:
   a. In the order of the RWR values of the group, replicas of data items are allocated until 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 RWR value to the data item is the highest among hosts that have free memory space.
   b. After allocating replicas of all kinds of data items, if there is still free memory space at mobile hosts in the group, replicas are allocated in the order of RWR value until the memory space is full.
4. The coordinator informs mobile hosts in its responsible group of the replica allocation result and the mobile hosts allocate the replicas according to the result.

5 Simulation Experiments

In this section, we show results of simulation experiments regarding performance evaluation of our proposed methods.

5.1 Simulation Model

The number of mobile hosts in the entire system is 40 ($M = M_1, M_2, \cdots, M_{40}$). Mobile hosts exist in an area of 50×50. There are eight different types of events in the simulation environment and each event has a home position, an emergency object, and four open (but not emergency) objects. The home position of each event is randomly determined in the 50×50 area and the emergency object and the four open objects are also randomly chosen from all data items. This represents that there exist

correlations between an event (schedule entry) and a position and between an event and objects. Each mobile host engaged in an event issues access (read) requests to the emergency and open objects of the event. In our simulation model, mobile hosts are engaged in the same eight types of events. A good example of an event is "shopping at store A," where the home position is the store location and the objects are price lists of the store and other stores, and so on.

Each mobile host (user) has its own schedule that consists of these eight events where each event appears once in the schedule and the occurrence order of the events is randomly determined at the beginning of the simulation. Thus, mobile hosts have different schedules with each other but their schedules consist of the same event set (the eight events). It takes 3,600 units of time for mobile hosts to complete one event. Here, the 3,600 units of time include the time duration for moving to the home position and each mobile host issues access and write requests to the objects during the moving duration as well as after reaching the home position. When the movement speed of a mobile host is very low, the host may not be able to reach the home position within the 3,600 units of time (though this is a very rare case). Even in such a case, the host finishes the event when the 3,600 units of time have passed in the same way as those that reach the home position. After finishing the last event, the host transits to the first event in the schedule again and repeats this process.

Every time an event finishes, each host is engaged in the next event with the probability of "E." Such a host is called a *host on schedule* and E is called the *on-schedule probability*. Each host on schedule moves to the home position of the event in which the host is currently engaged and stops at the position during the event. The movement speed is determined randomly between 0 and V. On the other hand, each *host off schedule* skips the event and spends the 3,600 units of time for the skipped event by moving according to the random waypoint model [5]. That is, each host remains stationary for a pause time, S. Then, it selects a random destination in the 50×50 area and moves to the destination at a speed determined randomly between 0 and V. After reaching the destination, it again stops for a pause time and repeats this behavior. After the 3,600 units of time pass, the mobile host off schedule behaves in the same way as that on schedule, i.e., it is engaged in the next event with the probability of E.

The communication range of each mobile host is a circle with radius R. The number of data items in the entire network is 40. M_i $(i = 1, \cdots, 40)$ holds D_i as the original. Data items are updated at inconstant intervals and the intervals are determined based on the exponential distribution with mean $1/W$, i.e., W is the write frequency. This represents that all data items have the same write frequency whether they are specified as the emergency or open objects or not. Each mobile host has the same size of memory for replication and creates up to C replicas. This assumption is not only for simplicity, but also for evaluating replica allocation methods properly. Replicas are periodically relocated based on the relocation period T. The access frequency of each mobile host, M_i, to D_j is p_{ij} $(j = 1, \cdots, 40)$ in either of the following two cases:

$$\begin{array}{ll} \textbf{Case 1 :} & p_{ij} = 0.125\{1 + (j - 20)/400\} \quad (\text{off schedule}), \\ \textbf{Case 2 :} & p_{ij} = 0, 1.0 \qquad\qquad\qquad\qquad\quad (\text{on schedule}). \end{array}$$

Case 1 is for mobile hosts off schedule and represents a situation where every mobile host has the same access characteristics and access frequencies vary in a small range. Case 2 is for mobile hosts on schedule. $p_{ij} = 1.0$ if D_j is included in the current event as the emergency object or an open object and $p_{ij} = 0$ otherwise. Case 2 represents a situation where a user on schedule issues access requests frequently to a small number of data items included in the current event. In both cases, the average number of access requests for all data items issued by a mobile host during a unit of time is 5. In case 1, the access frequencies range between 0.125×0.9525 and 0.125×1.05, but these numbers themselves have no special meaning. These are chosen so that the average total number of access requests issued at a unit of time becomes five. In a real mobile environment, it may be unrealistic that each mobile host issues five access requests during a unit of time. However, in our simulation experiments, the values of access (read) frequencies themselves do not affect the performance characteristics of the proposed and other existing methods, but the ratio of write frequencies and the access frequencies affects them. Thus, we unintentionally determine the values of the access frequencies used in our simulations, but we can change them arbitrarily.

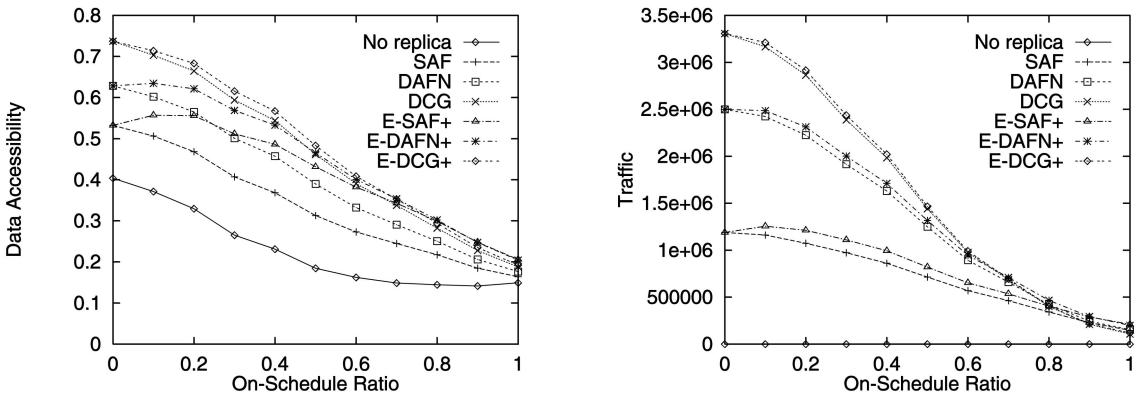
The happen time of all emergency objects, which is used for calculating the RWR values, is one unit of time. This represents that each mobile host on schedule issues access requests to the emergency object even during the time duration for moving to the home position.

Table 3 shows parameters and their values used in the simulation experiments. The parameters are basically fixed to constant values specified by numbers at the left side of the parenthetic values. The five parameters, E, W, T, R, and C, are chosen to examine their impacts on the performance of our proposed methods. In each of the following sections, the performance of our methods is evaluated when varying each of the five parameters. In doing so, each parameter is varied by the value specified by the former of the two parenthetic elements in Table 3 in the range specified by the latter. In the simulation experiments, we examine the average data accessibility and the total traffic of each of the six proposed methods during 500,000 units of time. Note that, on average, each host is engaged in an event 500,000 (simulation time)/3,600 (event execution time) $\times$ 0.2 (on schedule probability) $\simeq$ 28 times. Also, 40 (number of mobile hosts)/8 (number of events) $\times$ 0.2 = 1 host is engaged in each event in average. All the mobile hosts that engaged in the same event go to the common home position according to the event. The pause time 1 for mobile hosts off schedule represents a situation where users keep moving for 3,600 units of time without stopping, e.g., walking around a shopping center for one hour without a special objective.

The average data accessibility is defined as the ratio of the number of successful access requests to the number of all access requests issued during the simulation time, and the traffic is defined as the total hop count of data transmissions for allocating/relocating replicas that are performed during the simulation time. Here, we assume

TABLE 3
Parameter Configuration

Parameter	Meaning	Constant Values	(Increment, Range)
E	on-schedule probability	0.2	(0.1, 0∼1)
W	write frequency	0.005	(0.002, 0.001∼0.021)
T	relocation period	100	(500, 1∼5001)
R	radio communication range	7	(2, 1∼19)
C	memory size	10	(5, 0∼40)
S	pause time	1	
V	maximum movement speed	1	



(a) (b)

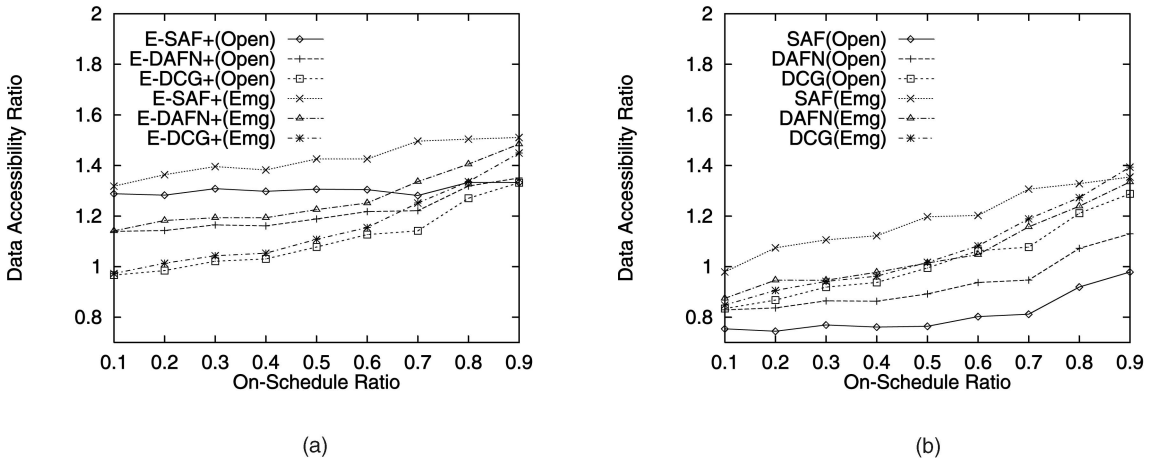
Fig. 4. Effects of on-schedule probability. (a) On-schedule probability and data accessibility. (b) On-schedule probability and traffic.

that access failures are caused only by network partitioning, but not caused by communication failures in the network. Therefore, an access request succeeds if and only if at least one of the connected mobile hosts holds the original or a valid replica of the target data item. In the three non-extended methods, SAF, DAFN, and DCG, replica allocation is determined based on access frequencies for mobile hosts off-schedule. For the purpose of comparison, we also evaluate the performance of the case without data replication, denoted by "No replica." We assume that the network topology (graph constructed by the connected mobile hosts) does not change during the algorithm execution of the proposed methods. The probability that the network topology changes, i.e., connections or disconnections of wireless links occur, during the algorithm execution is so small that we can neglect its impact on performance, since control message exchanges are usually done in less than 1 second. Thus, we include this assumption for the purpose of simplicity. Similarly, we also neglect traffic caused by the message exchanges.

5.2 Effects of the On-Schedule Probability

First, we examine the effects of the on-schedule probability, E, on each of the seven methods. Fig. 4 shows the simulation results. In both graphs, the horizontal axis indicates the on-schedule probability E. The vertical axes indicate data accessibility in the case of Fig. 4a and traffic in the case of Fig. 4b. From Fig. 4a, it can be seen that the proposed six methods give much higher data accessibility than "No replica" and this shows the effectiveness of data replication in ad hoc networks. As the on-schedule probability gets higher, data accessibility of the proposed six methods basically gets lower. In our simulation environment, a mobile host on schedule issues accesses requests only for five data items included in the current event and stays at a particular position during the event. When the on-schedule probability is high, most mobile hosts do not move and they have little chance to get necessary replicas from other mobile hosts. This is the reason why higher on-schedule probability gives lower data accessibility. This fact is also shown in Fig. 4b. When the on-schedule probability is high, the traffic caused by the proposed methods is low because mobile hosts rarely succeed to allocate the necessary replicas. From these results, it is shown that some mobility is required to improve data accessibility in an ad hoc network in which network partitioning occurs due to the low density of mobile hosts.

The reason why the data accessibility of the E-SAF+ and the E-DAFN+ methods gets higher as the on-schedule probability gets higher at first is as follows: In our simulation environment, mobile hosts off schedule have the same access characteristics and, thus, there are many replica duplications among the mobile hosts in the E-SAF+ and E-DAFN+ methods. Since the access characteristics of mobile hosts on schedule are different from that of mobile hosts off schedule, the existence of a little number of mobile hosts on schedule reduces such data duplications.



(a) (b)

Fig. 5. Effectiveness of giving priority to mobile hosts on schedule. (a) On-schedule probability and data accessibility ratio (extended methods). (b) On-schedule probability and data accessibility ratio (original methods).

Of the proposed methods, the extended three methods, E-SAF+, E-DAFN+, and E-DCG+, give higher data accessibility than the corresponding nonextended methods, especially for the E-SAF+ and E-DAFN+ methods. Since the increase in traffic is not so large in the extended methods, it can be seen that the three extended methods are more effective than the nonextended methods. The reason why the difference in data accessibility between the DCG and E-DCG+ methods is not so large is that both methods share many kinds of replicas in a stable group.

Of the three extended methods, the E-DCG+ method gives the highest data accessibility and the E-DAFN+ method follows in most cases. This result shows the effectiveness of sharing replicas among connected mobile hosts. However, the E-DCG+ method produces the highest amount of traffic in most cases. When the on-schedule probability is higher than 0.7, the E-DCG+ method gives lower data accessibility than the other two methods. Since mobile hosts on schedule request only five kinds of data items, in the E-DCG+ method, the mobile hosts are forced to replicate many kinds of data items that are never accessed. This causes the deterioration of data accessibility. In addition, since such unnecessary data items nearly always fail to be replicated, the traffic caused by this method is also low.

Consequently, the results in Fig. 4 show the trade-off relation between data accessibility and the amount of traffic. Since the E-DCG+ method gives about 50 percent higher data accessibility but three times higher traffic than the SAF method, it is an undesirable trade. Actually, reduction of traffic is usually very significant in mobile environments where mobile hosts have poor battery power and low network bandwidth. However, in some situations, such as intervehicle information sharing in high-speed wireless networks, the traffic caused by replica relocation is not a serious problem if the available network bandwidth is high enough. Moreover, the traffic in the entire network is caused not only by replica relocation, but also by data access, and the latter is generally much higher than the former. Therefore, the trade-off shown in Fig. 4 is not always undesirable. In a real environment, an appropriate method should be chosen among our proposed methods according to the system requirement.

Now, we examine the effectiveness of giving priority to mobile hosts on schedule in the three extended methods. Fig. 5 shows the ratios of data accessibility for open and emergency objects requested by mobile hosts on schedule to that for other data items requested by those off schedule. We call the ratio the *data accessibility ratio*. In both graphs, the horizontal axis indicates the on-schedule probability and the vertical axis indicates the data accessibility ratio. "Open" and "Emg" denote the data accessibility ratios for open and emergency objects, respectively. From Fig. 5a, the data accessibility ratio in the three extended methods is nearly always higher than 1 and, thus, we can see the effectiveness of giving priority to mobile hosts on schedule. On the contrary, from Fig. 5b, the data accessibility ratio in the three nonextended methods is lower than that in the extended methods and is lower than 1 when the on-schedule probability is low. The effectiveness is more conspicuous in the E-SAF+ and the E-DAFN+ methods than the E-DCG+ method. This is because the data accessibility for data items requested by mobile hosts off schedule is lower in the two methods than the E-DCG+ method due to heavy replica duplication. Comparing open objects and emergency objects, the data accessibility ratio for emergency objects is higher than that for open objects in each of the three extended methods.

5.3 Effects of the Write Frequency

Next, we examine the effects of the write frequency on each of the seven methods. Fig. 6 shows the simulation results. In both graphs, the horizontal axis indicates the write frequency W. The vertical axes indicate data accessibility and traffic, respectively.

Fig. 6a shows that, as the write frequency gets lower, the data accessibility of the proposed six methods gets higher. This is because, when the write frequency is low, replicas created are valid for a long time. Similar to the result in Fig. 4a, the data accessibility in the three extended methods is higher than that in the corresponding nonextended methods. Of the three extended methods, the E-DCG+ method always gives the highest data accessibility and the E-DAFN+ method follows. As the write frequency gets higher, the differences in data accessibility among the six methods get slightly smaller. When the write frequency is high, replicas become invalid soon. In most cases, access requests succeed only when the request issue hosts connect with hosts that hold the originals of the requested data

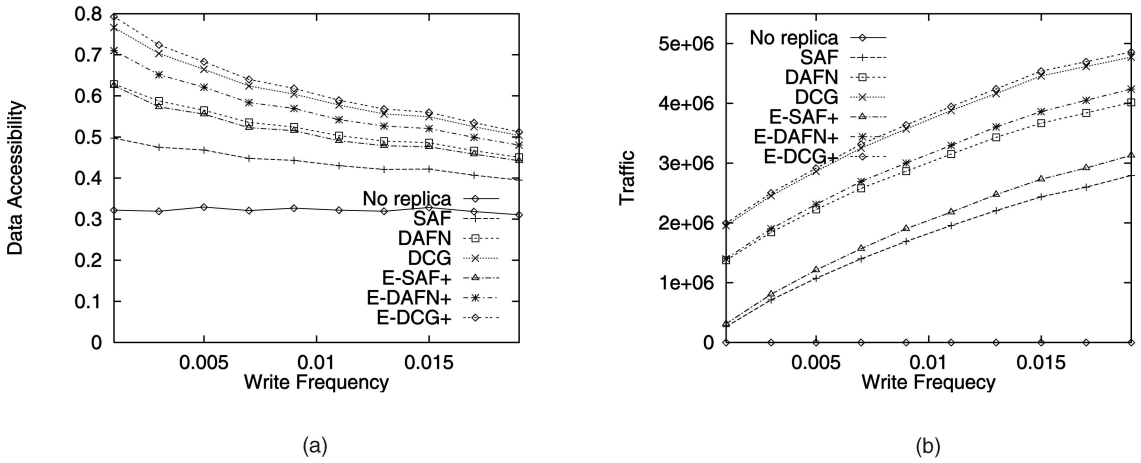


Fig. 6. Effects of the write frequency. (a) Write frequency and data accessibility. (b) Write frequency and traffic.

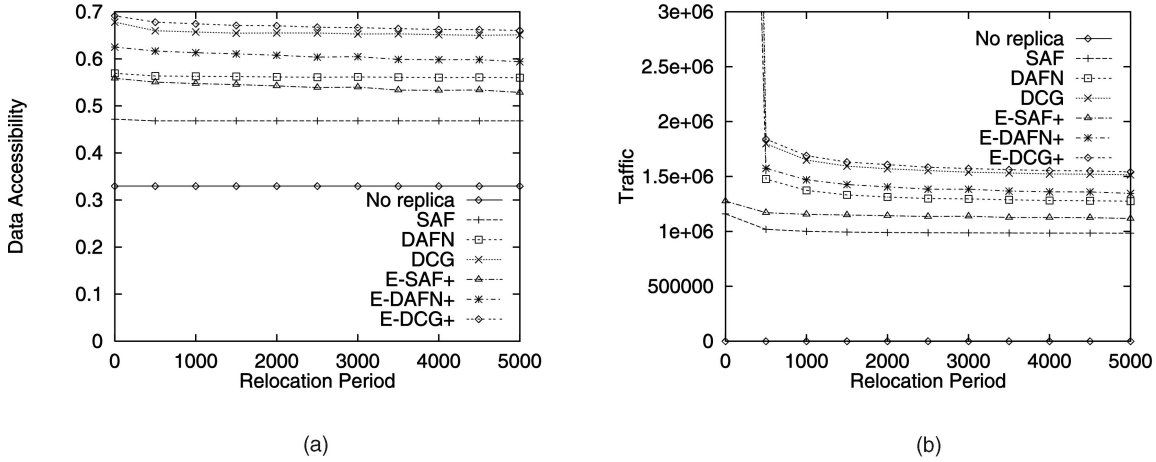


Fig. 7. Effects of the relocation period. (a) Relocation period and data accessibility. (b) Relocation period and traffic.

items. This is the reason why replication strategies have a little impact on data accessibility.

From Fig. 6b, it can be seen that the three extended methods produce a slightly higher amount of traffic than the corresponding nonextended methods. Of the three extended methods, the E-DCG+ method produces the highest amount of traffic and the E-DAFN+ produces the next highest. As the write frequency gets higher, the traffic also gets higher. This is because each mobile host must frequently refresh the replicas they hold after the originals have been updated.

5.4 Effects of the Relocation Period

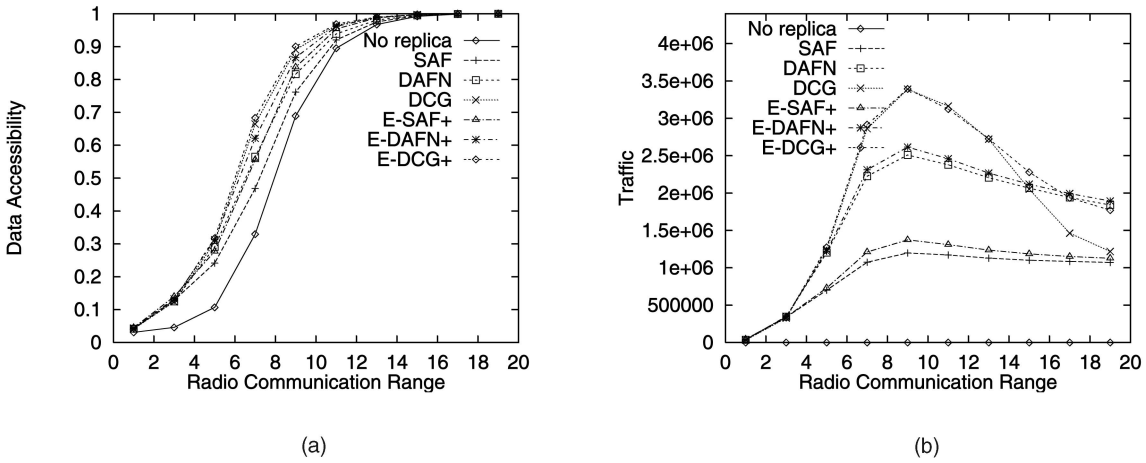
We examine the effects of the relocation period on each of the seven methods. Fig. 7 shows the simulation results. In both graphs, the horizontal axis indicates the relocation period T. The vertical axes indicate data accessibility and traffic, respectively.

From Fig. 7a, the data accessibility in the three extended methods is higher than that in the corresponding nonextended methods. Of the three extended methods, the E-DCG+ method gives the highest data accessibility, and the E-DAFN+ method gives the next highest. As the relocation period gets longer, the data accessibility of the proposed six methods gets slightly lower. This is because a shorter relocation period can detect the changes of network topology and events sensitively. From Fig. 7b, the three extended methods produce a slightly higher amount of traffic than the corresponding nonextended methods. Of the three extended methods, the E-DCG+ method produces the highest amount of traffic, and the E-DAFN+ produces the next highest. The traffic caused by the two methods is inversely proportional to the relocation period. The traffic caused by the SAF and E-SAF+ methods is little affected by the relocation period. This is because access frequencies and RWR values of data items for mobile hosts off schedule are constant and, thus, in the SAF and E-SAF+ methods, the replica relocation occurs only when the mobile hosts change the status to be on schedule.

5.5 Effects of the Radio Communication Range

We examine the effects of the radio communication range of mobile hosts on each of the seven methods. Fig. 8 shows the simulation results. In both graphs, the horizontal axis indicates the communication range, R. The vertical axes indicate the data accessibility and the traffic, respectively. This experiment corresponds to an experiment that examines the effect of the density of mobile hosts in the network. Fig. 9 shows the relationship between the radio communication range and the number of neighboring hosts of a mobile host (average, minimum, and maximum).

From Fig. 8a, as the radio communication range gets larger, the data accessibility also gets higher in every



(a) (b)

Fig. 8. Effects of the radio communication range. (a) Radio communication range and data accessibility. (b) Radio communication range and traffic.

method. In most cases, the E-DCG+ method gives the highest data accessibility. When the communication range is very small, every method gives almost the same data accessibility. This is because the number of mobile hosts connected to each other is small and, thus, replica relocation rarely occurs. When the communication range is very large, every method also gives almost the same data accessibility. This is because most mobile hosts are connected to each other and, thus, mobile hosts can access original data items in most cases.

From Fig. 8b, as the radio communication range gets larger, the traffic produced by each method also gets higher at first, but it gets lower from a certain point. When the radio communication range is very small, every method produces low traffic. This is because the number of mobile hosts connected to each other is small and, thus, replica relocation and replica refreshment do not cause high traffic. When the radio communication range is very large, the DCG and E-DCG+ methods give lower traffic than the DAFN and E-DAFN+ methods. This is because, in the DCG and E-DCG+ methods, most mobile hosts belong to one big group and, thus, replica relocation rarely occurs.

5.6 Effects of the Size of Memory Space

We examine the effects of the size of memory space on each of the seven methods. Fig. 10 shows the simulation results.

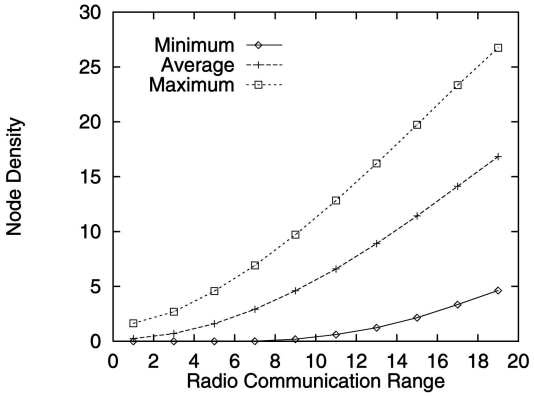


Fig. 9. Radio communication range and number of neighbors.

In both graphs, the horizontal axis indicates the memory size of a mobile host, C. The vertical axes indicate the data accessibility and the traffic, respectively.

From Fig. 10a, as the memory size gets larger, the data accessibility also gets higher in every method. Of the three extended methods, in most cases, the E-DCG+ method gives the highest data accessibility and the E-DAFN+ method gives the next. The data accessibility of the SAF method is linearly affected by the memory size.

From Fig. 10b, as the memory size gets larger, the traffic produced by the DAFN, E-DAFN+, DCG, and E-DCG+ methods also gets higher at first, but it gets lower from a certain point. When the memory is very small, replica relocation does not cause large traffic because the number of replicas created is small. When the memory size is very large, the four methods produce small traffic because each host holds replicas of most data items and, thus, replica relocation rarely occurs, i.e., traffic is mainly produced by replica refreshment.

5.7 Effects of the Skew of Access and Write Frequencies

Since the three extended methods determine replica relocation based on RWR values, these methods replicate data items whose write frequencies are very low even if their access frequencies are also low, i.e., data items with low utility. This may cause performance deterioration in the three methods. To investigate this, we examine the effect of the skew of access and write frequencies on each of these methods.

The simulation environment is same as that in the previous sections except for access and write frequencies for some data items. Specifically, p_{ij} and W of these items are changed as follows:

$$
\begin{aligned}
p_{ij} &= 0.0625\{1 + (j - 20)/400\} \quad \text{(off schedule)} \\
&\quad (\text{for } D_1, \cdots, D_{10}), \\
p_{ij} &= 0.725\{1 + (j - 20)/400\} \quad \text{(off schedule)} \\
&\quad (\text{for } D_{31}, \cdots, D_{40}), \\
W &= 0.00025 \\
&\quad (\text{for } D_1, \cdots, D_{10}).
\end{aligned}
$$

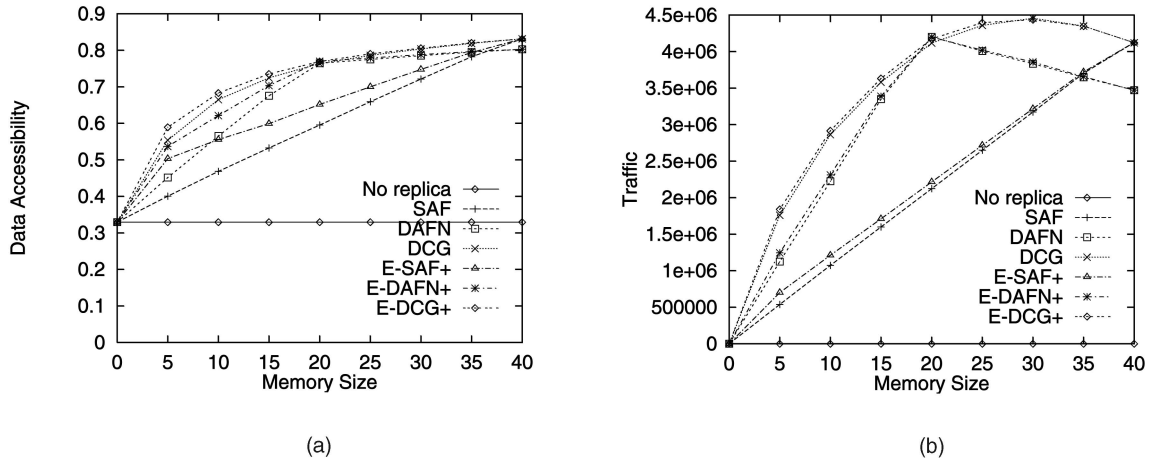


Fig. 10. Effects of the memory size. (a) Memory size and data accessibility. (b) Memory size and traffic.

This shows that 10 data items with identifiers $D_1, \cdots, D_{10}$ (*cold items*) have half of the access frequencies and 20 times lower write frequencies, i.e., 10 times higher RWR values, from the original parameter setting and 10 data items with identifier $D_{31}, \cdots, D_{40}$ (*hot items*) have five times higher access frequencies, i.e., five times higher RWR values. The access frequencies of mobile hosts on schedule to these data items are not changed from the original setting. This situation represents an extreme case where the three extended methods work badly.

In this experiment, we evaluated not only the data accessibility and the traffic for replica relocation, but also the traffic for data access. Here, the traffic for data access is defined as the total hop count of data transmission during the simulation time for accessing data items (originals/replicas) held by connected mobile hosts. Fig. 11 shows the simulation results. In all graphs, the horizontal axis indicates the memory size of a mobile host, C. The vertical axes indicate the data accessibility, the traffic, and the traffic for data access, respectively.

From Fig. 11a and Fig. 11c, the three extended methods give lower data accessibility and higher traffic for data access than the corresponding nonextended methods. This shows that the performance of the three extended methods becomes worse in this situation. The difference in data accessibility between the E-DCG+ and DCG methods is much smaller than that between the other two pairs. This shows the effectiveness of sharing many kinds of replicas in the E-DCG+ methods. Although this experiment is an extreme case, similar situations may occur in a real environment. We can solve this problem by setting a threshold (lower bound) of access frequency to prevent mobile hosts from replicating data items with low access frequencies.

6 Invalidation of Old Replicas

From the simulation results in the previous section, it is confirmed that our proposed methods achieve high data accessibility in ad hoc networks where each data item is updated at inconstant intervals. However, mobile hosts may access invalid replicas whose originals have been updated. If a mobile host accesses an invalid replica, a roll back may occur when it connects with the mobile host holding the original. Such invalid accesses consume the power of mobile hosts and this is a serious problem for mobile hosts that usually have poor resources. To solve this problem, the following two approaches could be considered:

- A lifetime is assigned to each replica when it is allocated to a mobile host and the host automatically discard the replica when the life time has expired.
- The invalidation report for invalidating an old replica is broadcast when the holder of the original updates it.

The first approach is effective because this needs no message exchange among mobile hosts and does not cause any extra overhead. However, the effectiveness of the scheme heavily depends on the propriety of the lifetime assigned to the replicas. In a real environment, it is very difficult to determine the appropriate lifetime of each replica because the data updates usually occur randomly. Thus, the second approach is more effective where the occurrence of data update does not have prominent characteristics. In this section, we discuss the impact of replica invalidation on the number of accesses to old replicas. For this aim, we employ two replica invalidation methods, the *Update Broadcast (UB) method* and the *Connection Rebroadcast (CR) method*, which we proposed in [13].

6.1 Outline of the Replica Invalidation Methods

We assume that each mobile host holds a table in which the information on time stamps of all data items in the entire network is recorded. A time stamp is the latest update time of the corresponding data item which the mobile host knows. This might be different from the actual latest update time. This information table is called the *time stamp table*.

6.1.1 UB Method

In the UB method, a mobile host holding an original data item broadcasts the invalidation report to connected mobile hosts every time it updates the data item. The invalidation report includes the following information:

- The data identifier.
- The update time (time stamp).

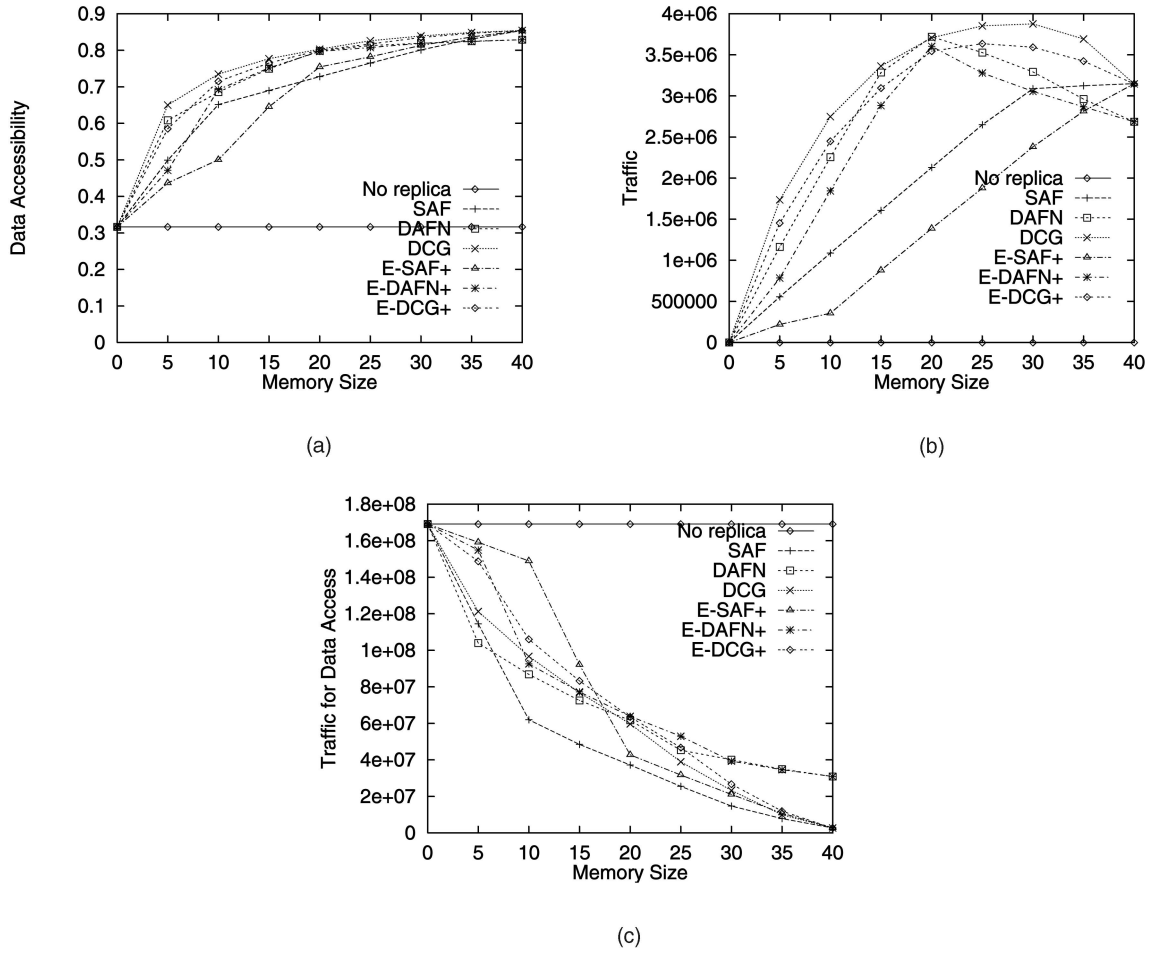


Fig. 11. Effects of the memory size (p_{ij} and W are skewed). (a) Memory size and data accessibility. (b) Memory size and traffic. (c) Memory size and traffic for data access.

When a mobile host receives the invalidation report, it refers to and updates its own time stamp table and checks whether the replica held by the host is valid. If the host holds an invalid replica of the corresponding data item, the host discards the replica. The memory space for the discarded replica is kept free and the new replica of the data item is allocated again on the free memory space when it accesses the original/replica of the data item with the same or larger time stamp than that in its own time stamp table.

6.1.2 CR Method

In the CR method, similar to the UB method, a mobile host holding an original data item broadcasts the invalidation report every time it updates the data item. In addition, every time two mobile hosts are newly connected with each other, they rebroadcast invalidation reports that they have previously received. The algorithm of this method is as follows:

1. When two mobile hosts M_i and M_j ($i < j$) are newly connected with each other, the mobile host with larger suffix (j) of host identifier (M_j) transmits its own time stamp table to the other one.
2. Mobile host M_i compares the entry for each data item in its own time stamp table with that in the time stamp table received from mobile host M_j and updates its own time stamp table. Then, the following processes are executed:
   - M_i broadcasts the invalidation reports for data items whose time stamps held by M_i are smaller than those held by M_j to its connected mobile hosts except for M_j.
   - M_i sends the information on the updated time stamps for data items whose time stamps held by M_j are smaller than those held by M_i to M_j. Then, M_j broadcasts the invalidation reports for these items to its connected mobile hosts except for M_i.

Mobile hosts that receive the invalidation reports invalidate their replicas in the same way as the UB method.

In this method, connected mobile hosts hold the same time stamp table because invalidation reports are rebroadcast whenever two mobile hosts are newly connected. Moreover, invalidation reports spread among a large number of mobile hosts and the replica invalidation is performed at these mobile hosts even if they are not connected to the mobile hosts holding the original data items. Thus, this method can further reduce the number of accesses to invalid replicas than the UB method. However, when the network topology frequently changes, the traffic caused by broadcasting invalidation reports is much higher

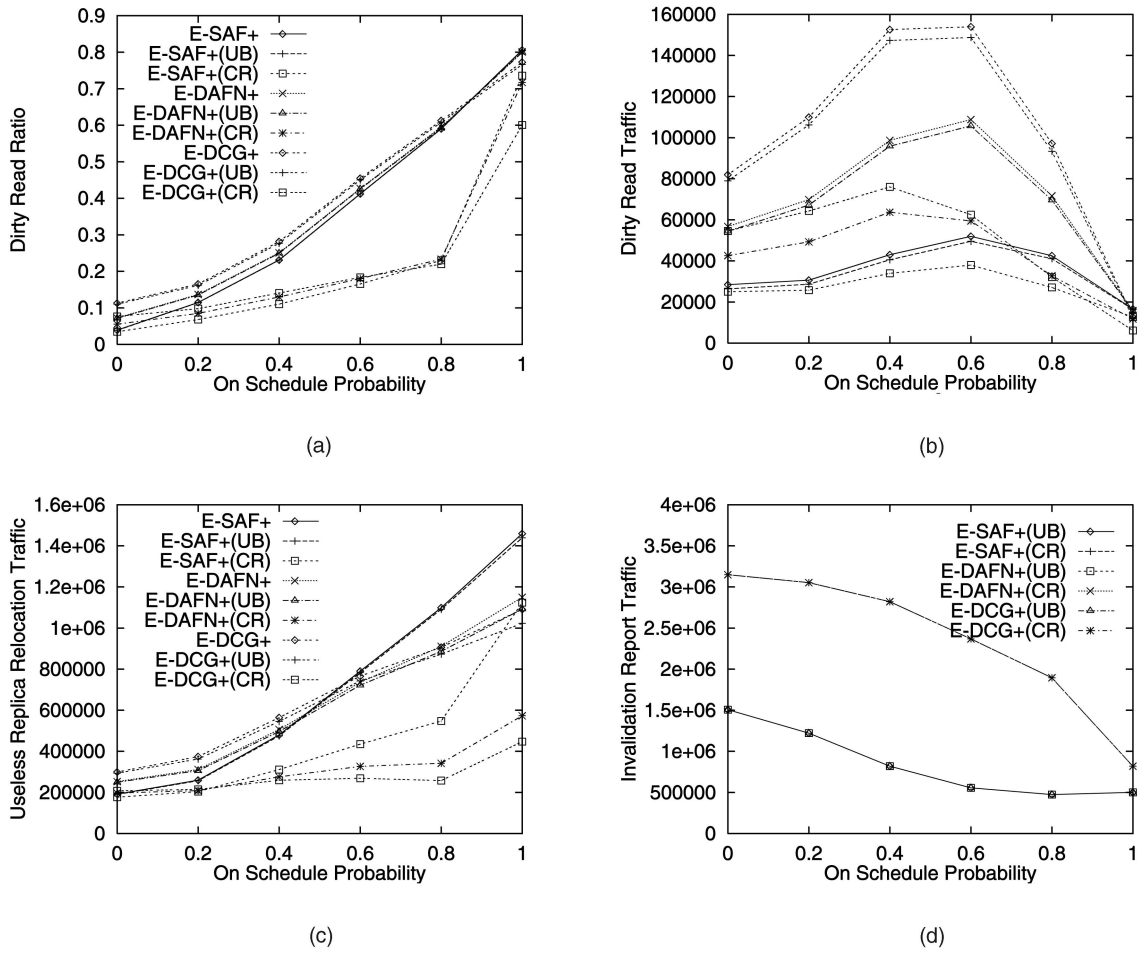


Fig. 12. Effects of replica invalidation. (a) On-schedule probability and dirty read ratio. (b) On-schedule probability and dirty read traffic. (c) On-schedule probability and useless replica. (d) On-schedule probability and invalidation report traffic.

than that in the UB method due to the frequent rebroadcast of the reports.

6.2 Effectiveness of Replica Invalidation

We examine the effectiveness of the two replica invalidation methods, UB and CR, using a simulation experiment. Fig. 12 shows the result of the experiment in which the same parameter configuration is applied as that in Section 5. In all graphs, the horizontal axis indicates the on-schedule probability E. The vertical axes indicate *dirty read ratio* in the case of Fig. 12a, *dirty read traffic* in the case of Fig. 12b, *useless replica relocation traffic* in the case of Fig. 12c, and *invalidation report traffic* in the case of Fig. 12d.

The dirty read ratio is defined as the ratio of the number of access requests that read old replicas (dirty read) to the number of all access requests issued during the simulation time. The dirty read traffic is defined as the total hop count of data transmissions for dirty reads. The useless replica relocation traffic is defined as the total hop count of data transmissions for allocating/relocating old replicas that have already become invalid. The invalidation report traffic is defined as the total hop count of message transmissions for broadcasting invalidation reports. Here, in the CR method, the traffic for broadcasting an invalidation report is calculated by the product of the total hop count for broadcasting it and the number of elements (data identifiers) included in the report. In all graphs, "E-SAF+(UB)" denotes the case where the E-SAF+ method is used as a replica relocation method and the UB method is used as a replica invalidation method and so forth. In Fig. 12b, although there is no explanation for each line and plot due to the limitation of space, it corresponds to the same method as that in the case of Fig. 12a. In Fig. 12d, we omit the results for E-SAF+, E-DAFN+, and E-DCG+ methods without replica invalidation since they produce no invalidation report traffic.

From Fig. 12a, it is shown that the replica invalidation methods are effective for reducing the dirty read ratio. The CR method drastically reduces the dirty read ratio compared with the UB method when employing the same replica allocation method. As the on-schedule probability gets higher, in all the nine patterns, the dirty read ratio gets higher. When the on-schedule probability is 1, 60 percent to 80 percent of data requests result in dirty reads. As mentioned in Section 5.2, when the on-schedule probability is high, most mobile hosts do not move and they have little chance to get the necessary replicas from other mobile hosts. Since such a mobile host issues access requests only for five data items included in the current event, it repeatedly accesses invalid replicas of the five data items that are held by the host in its own memory space. This fact is also shown in Fig. 12b. When the on-schedule ratio is nearly 1, the dirty

read traffic is very low, while the dirty read ratio is high as shown in Fig. 12a.

Fig. 12b shows that the UB method slightly reduces the dirty read traffic and the CR method drastically reduces it compared with the case without replica invalidation. As the on-schedule probability gets higher, in all the nine patterns, the dirty read traffic also gets higher at first, but it gets lower from a certain point. As mentioned above, in our simulation model, mobile hosts on schedule tend to hold invalid replicas. Thus, as the on-schedule probability gets higher, the number of invalid replicas gets higher in the entire network and mobile hosts off schedule frequently access these invalid replicas. This is why the dirty read traffic gets higher at first. However, when the on-schedule ratio is high, most mobile hosts are on schedule, and they frequently access invalid replicas that they hold. This is why the dirty read traffic is very low when the on-schedule is nearly 1. Of the replica allocation methods, the E-DCG+ method produces the highest dirty read traffic and the E-DAFN+ method follows it. This is because the E-DCG+ method has the highest possibility that connected mobile hosts hold replicas of the target data items whether they are valid or invalid.

Fig. 12c shows that the replica invalidation methods are effective for reducing the useless replica relocation traffic. The CR method drastically reduces the useless replica relocation traffic compared with the UB method when employing the same replica allocation method. As the on-schedule probability gets higher, in all the nine patterns, the useless replica relocation traffic gets higher. This is because a mobile host on schedule issues access requests only for five data items and in most cases it replicates them by copying their invalid replicas from connected mobile hosts. Of the replica allocation methods, the E-SAF+ method shows the highest useless replica relocation traffic when the on-schedule probability is high. Since there are many replica duplications in the E-SAF+ method, each host has little chance to get necessary replicas by accessing valid ones.

In Fig. 12d, three lines for the three replica allocation methods overlap for each of the two cache invalidation methods. The CR method produces a higher amount of invalidation report traffic than the UB method. This result shows that the CR method produces a much larger number of invalidation reports than the UB method. As the on-schedule probability gets higher, in both cache invalidation methods, the invalidation report traffic gets lower. When the on-schedule probability is high, most mobile hosts stop at the home position and the change in network topology rarely occurs. This is why the invalidation report traffic gets lower in the CR method as the on-schedule probability gets higher. In the UB method, the invalidation report traffic also gets lower because most connected mobile hosts stay at the home position and an invalidation report can be sent to them by a one-hop transmission.

From Fig. 12, it is shown that the replica invalidation methods cause the invalidation report traffic but decrease the dirty read and useless replica relocation. Although in Fig. 12 the invalidation report traffic shows higher values than the dirty read traffic and useless replica relocation traffic, we evaluate only hop counts, but do not consider the sizes of invalidation reports and data items. Generally, an invalidation report is much smaller than a data item, e.g., few bytes versus a few hundred kilobytes or few megabytes. Thus, in a real environment, we can neglect the impact of increase in traffic caused by broadcasting invalidation reports.

7 Conclusion

In this paper, we have discussed replica allocation in ad hoc networks to improve data accessibility. First, we have proposed three replication methods in an environment where each data item is not updated. Then, we extended these three methods by considering aperiodic data updates since, in a real environment, updates do occur aperiodically. These extended methods also accommodate user schedules and emergency objects. By using the Read/Write Ratio (RWR), the extended methods can handle data updates at each mobile host. We allow emergency objects to be unconditionally replicated. This is targeting safety and time-critical application domains.

The simulation results showed that the three extended methods work well in an environment where each data item is randomly updated and mobile users behave based on their schedules. Since there is a trade-off relationship between the improvement of data accessibility and the reduction of traffic, there cannot be one universal optimal method for replica relocation. For example, if the traffic caused by replica relocation is not a serious problem, e.g., intervehicle information sharing in a high-speed wireless network, the E-DCG+ method should be the optimal method. On the other hand, if the traffic is a very serious problem, e.g., information sharing in a shopping center using a narrow-band cellular phone network, the SAF or E-SAF+ method may be the optimal method. Therefore, in a real environment, an appropriate method should be chosen among our proposed methods according to the situation. The simulation results also showed that the three extended methods give poor performance when some data items have very low write frequencies and not high access frequencies. This drawback is due to that RWR is defined as the ratio of read frequencies to write frequencies. Thus, as part of our future work, we plan to address this problem by setting a threshold (lower bound) of access frequency to prevent mobile hosts from replicating data items with low access frequencies. Moreover, we discussed how we can reduce the number of accesses to old replicas and have reported the simulation experiment to verify the effectiveness of broadcasting replica invalidation reports.

The implementation of our scheme calls for careful consideration of several issues such as the maintenance of access histories and user profiles. In this paper, we assume a mesoscale ad hoc network and, thus, we neglect the effect of broadcasting small messages. In order to apply our proposed methods in a large scale ad hoc network, some extensions can be incorporated to prevent broadcast storms, e.g., using gossiping protocols [4] and setting TTL (Time To Live) for broadcasting. In addition, the replica allocation algorithms in our proposed methods can be modified to apply them in a much larger scale network because they require message exchanges among all connected mobile hosts and relocate replicas.

ACKNOWLEDGMENTS

The authors would like to express their sincere appreciation to Professor Shojiro Nishio of Osaka University for invaluable comments on this work.

REFERENCES

[1] A.V. Aho, J.E. Hopcroft, and J.D. Ullman, *The Design and Analysis of Computer Algorithms.* Addison-Wesley, 1974.
[2] D.J. Baker, J. Wieselthier, and A. Ephremides, "A Distributed Algorithm for Scheduling the Activation of Links in a Self-Organizing, Mobile, Radio Network," *Proc. IEEE Int'l Conf. Comm. (ICC '82),* pp. 2F6.1-2F6.5, 1982.
[3] D. Barbara and T. Imielinski, "Sleepers and Workholics: Caching Strategies in Mobile Environments," *Proc. ACM SIGMOD '94,* pp. 1-12, 1994.
[4] K.P. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky, "Bimodal Multicast," *ACM Trans. Computer Systems,* vol. 17, no. 2, pp. 41-88, 1999.
[5] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *Proc. MobiCom '98,* pp. 159-164, 1998.
[6] M.J. Carey and M. Livny, "Distributed Concurrency Control Performance: A Study of Algorithm, Distribution, and Replication," *Proc. 14th Very Large Data Bases Conf.,* pp. 13-25, 1988.
[7] L.D. Fife and L. Gruenwald, "Research Issues for Data Communication in Mobile Ad-Hoc Network Database Systems," *ACM SIGMOD Record,* vol. 32, no. 2, pp. 42-47, 2003.
[8] T. Hara, "Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility," *Proc. IEEE INFOCOM 2001,* pp. 1568-1576, 2001.
[9] T. Hara, "Replica Allocation Methods in Ad Hoc Networks with Data Update," *ACM-Kluwer J. Mobile Networks and Applications,* vol. 8, no. 4, pp. 343-354, 2003.
[10] T. Hara and S.K. Madria, "Dynamic Data Replication Schemes for Mobile Ad-Hoc Network Based on Aperiodic Updates," *Proc. Int'l Conf. Database Systems for Advanced Applications (DASFAA '04),* pp. 869-881, 2004.
[11] T. Hara, N. Murakami, and S. Nishio, "Replica Allocation for Correlated Data Items in Ad-Hoc Sensor Networks," *ACM SIGMOD Record,* vol. 33, no. 1, pp. 38-43, 2004.
[12] T. Hara and S.K. Madria, "Consistency Management Among Replicas in Peer-to-Peer Mobile Ad Hoc Networks," *Proc. Int'l Symp. Reliable Distributed Systems (SRDS '05),* pp. 3-12, 2005.
[13] H. Hayashi, T. Hara, and S. Nishio, "Cache Invalidation for Updated Data in Ad Hoc Networks," *Proc. Int'l Conf. Cooperative Information Systems (CoopIS '03),* pp. 516-535, 2003.
[14] Y. Huang, P. Sistla, and O. Wolfson, "Data Replication for Mobile Computer," *Proc. ACM SIGMOD '94,* pp. 13-24, 1994.
[15] J. Jing, A. Elmagarmid, A. Helal, and R. Alonso, "Bit-Sequences: An Adaptive Cache Invalidation Method in Mobile Client/Server Environments," *ACM/Baltzer Mobile Networks and Applications,* vol. 2, no. 2, pp. 115-127, 1997.
[16] D.B. Johnson, "Routing in Ad Hoc Networks of Mobile Hosts," *Proc. IEEE Workshop Mobile Computing Systems and Applications,* pp. 158-163, 1994.
[17] G. Karumanchi, S. Muralidharan, and R. Prakash, "Information Dissemination in Partitionable Mobile Ad Hoc Networks," *Proc. Int'l Symp. Reliable Distributed Systems (SRDS '99),* pp. 4-13, 1999.
[18] R. Ladin, B. Liskov, L. Shrira, and S. Ghemawat, "Providing High Availability Using Lazy Replication," *ACM Trans. Computer Systems,* vol. 10, no. 4, pp. 360-391, 1992.
[19] S.K. Madria, "Timestamps to Detect R-W Conflicts in Mobile Computing," *Proc. Int'l Workshop Mobile Data Access (MDA '98),* pp. 242-253, Nov. 1998.
[20] M.R. Pearlman and Z.J. Haas, "Determining the Optimal Configuration for the Zone Routing Protocol," *IEEE J. Selected Areas in Comm.,* vol. 17, no. 8, pp. 1395-1414, 1999.
[21] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Proc. ACM SIGCOMM '94,* pp. 234-244, 1994.
[22] C.E. Perkins and E.M. Royer, "Ad Hoc on Demand Distance Vector Routing," *Proc. IEEE Workshop Mobile Computing Systems and Applications,* pp. 90-100, 1999.
[23] E. Pitoura and B. Bhargava, "Maintaining Consistency of Data in Mobile Distributed Environments," *Proc. IEEE Int'l Conf. Distributed Computing Systems (ICDCS '95),* pp. 404-413, 1995.
[24] K. Rothermel, C. Becker, and J. Hahner, "Consistent Update Diffusion in Mobile Ad Hoc Networks," Technical Report 2002/04, Computer Science Dept., Univ. of Stuttgart, 2002.
[25] F. Sailhan and V. Issarny, "Cooperative Caching in Ad Hoc Networks," *Proc. Int'l Conf. Mobile Data Management (MDM '03),* pp. 13-28, 2003.
[26] M. Stonebraker, "Concurrency Control and Consistency in Multiple Copies of Data in Distributed INGRES," *IEEE Trans. Software Eng.,* vol. 3, no. 3, pp. 188-194, 1979.
[27] R.H. Thomas, "A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases," *ACM Trans. Database Systems,* vol. 4, no. 2, pp. 180-209, 1979.
[28] 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.
[29] O. Wolfson and A. Milo, "The Multicast Policy and Its Relationship to Replicated Data Placement," *ACM Trans. Database Systems,* vol. 16, no. 1, pp. 181-205, 1991.
[30] K.L. Wu, P.S. Yu, and M.S. Chen, "Energy-Efficient Caching for Wireless Mobile Computing," *Proc. IEEE Int'l Conf. Data Eng. (ICDE '96),* pp. 336-343, 1996.
[31] S.-Y. Wu and Y.-T. Chang, "An Active Replication Scheme for Mobile Data Management," *Proc. Int'l Conf. Database Systems for Advanced Applications (DASFAA '99),* pp. 143-150, 1999.
[32] J. Xu, Q. Hu, D.L. Lee, and W.-C. Lee, "SAIU: An Efficient Cache Replacement Policy for Wireless On-Demand Broadcast," *Proc. ACM Int'l Conf. Information and Knowledge Management (ACM CIKM '02),* pp. 46-53, 2000.

Takahiro Hara received the BE, ME, and DrE degrees in information systems engineering from Osaka University, Osaka, Japan, in 1995, 1997, and 2000, respectively. Currently, he is an associate professor in the Department of Multimedia Engineering at Osaka University. He is currently serving as a program chair of the IEEE International Conference on Mobile Data Management (MDM '06) and as a PC vice-chair of the IEEE International Conference on Advanced Information Networking and Applications (AINA '06). He guest edited the *IEEE Journal on Selected Areas in Communications* special issues on peer-to-peer communications and applications. He was a PC vice-chair of the IEEE International Conference on Data Engineering (ICDE '05) and the IEEE International Conference on Parallel and Distributed Systems (ICPADS '05). He has served and is serving for various international conferences such as DASFAA, ACM MobiHoc, and ACM SAC. His research interests include distributed databases, peer-to-peer systems, mobile networks, and mobile computing systems. He is a member of four learned societies, including the IEEE and the ACM.

Sanjay K. Madria received the PhD degree in computer science from the Indian Institute of Technology, Delhi, India, in 1995. He is an associate professor in the Department of Computer Science at the University of Missouri-Rolla. Earlier, he was a visiting assistant professor in the Department of Computer Science, Purdue University, West Lafayette. He has published more than 100 journal and conference papers in the areas of Web data warehousing, mobile databases, and nested transaction management and performance issues. He guest edited the *WWW Journal* and *Data and Knowledge Engineering* special issues on Web data management and data warehousing. He coauthored the book *Web Data Management: A Warehouse Approach* (Springer-Verlag) and served as a program cochair for the ECWEB '00 and '01 conferences held in the United Kingdom and Germany. He is serving as a PC member of various database conferences, cochair of the XSDM workshops, and reviewer for many reputed database journals published by the ACM, the IEEE, and Springer, among others. His research is supported by grants from the US National Science Foundation, the US Department of Energy, the UM research board, and from industry. He is a senior member of the IEEE.