Replica Allocation for Correlated Data Items in Ad Hoc Sensor Networks

Takahiro Hara, Norishige Murakami, Shojiro Nishio
Department of Multimedia Engineering, Osaka University
{hara, norisige, nishio}@ist.osaka-u.ac.jp

Abstract

To improve data accessibility in ad hoc networks, in our previous work we proposed three methods of replicating data items by considering the data access frequencies from mobile nodes to each data item and the network topology. In this paper, we extend our previously proposed methods to consider the correlation among data items. Under these extended methods, the data priority of each data item is defined based on the correlation among data items, and data items are replicated at mobile nodes with the data priority. We employ simulations to show that the extended methods are more efficient than the original ones.

1 Introduction

Recent advances in wireless communication and sensing device technologies have led to an increasing interest in *sensor networks* that are constructed by low-cost, low-power, and multifunctional sensor nodes [2, 6]. Sensor nodes consist of sensing, data processing, and communication components and typically form *ad hoc networks* [3, 7, 10, 11], in which every node plays the role of a router and communicates with other nodes. Even if the source and the destination nodes are not within each other's communication range, data packets are forwarded to the destination by relaying transmission through other nodes that exist between the two nodes. Since no special infrastructure is required, many applications are expected to be developed in ad hoc sensor networks in various fields such as military affairs and commerce.

In this paper, we assume ad hoc sensor networks in which sensor nodes dynamically change their locations, i.e., mobile ad hoc networks (MANETs). That is, sensor nodes are mobile computers equipped with sensing devices. The following contains some scenarios in which sensor MANETs could be useful:

Rescue situations: Rescue workers engaged in disaster relief investigate the extent of the damage around them and collaboratively work by sharing the information on their locations and findings.

Excavations: Members of a research project team engaged in an archeological excavation collect various phenomenal data from sensors and share the obtained data with other members to streamline work.

In MANETs, since nodes move freely, network division occurs frequently. If a network is divided into two networks, nodes in one of the divided two networks cannot access data items held by nodes in the other network (Figure 1). Thus, data accessibility is lower than that in fixed networks. In MANETs, it is critical to prevent deterioration of data accessibility at the point of network division.

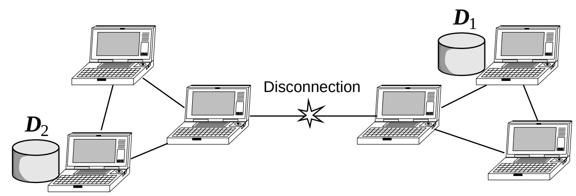

Figure 1. Network division and data access.

A possible and effective solution is to replicate data items at nodes that are not the owners of the original data. Since mobile (sensor) nodes generally have poor resources, it is usually impossible for nodes to have replicas of all data items in the network. For example, let us suppose a situation where a research project team that conducts an archeological excavation constructs a MANET 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 node to have replicas of all the data. Consequently, it is very important for system performance to decide which data items are replicated.

While there are a few studies that focus on data replication in MANETs[8, 9], they assume unlimited memory space for data replication. In [4], we assumed an environment where each node has limited memory space for creating replicas, and proposed three replica allocation methods for improving data accessibility.

Users (sensor nodes) often access certain sets of data items collectively, i.e., correlation generally exists among data items. In [4], we did not take the correlation among data items into account. However, data replication regardless of such data correlation might cause a situation whereby a node requests two correlated data items at the same time although it can access only one of them. If this situation frequently occurs, the data accessibility of the whole system deteriorates.

In this paper, we extend our previously proposed methods to consider the correlation among data items. In these extended methods, the data priority of each data item is defined based on the correlation among data items, and data items are replicated at nodes with the data priority.

The remainder of the paper is organized as follows. In section 2, we describe correlation among data items, then in section 3, we explain our extended methods. In section 4, we show the simulation results. Finally, in section 5, we summarize this paper.

2 Correlation among Data Items

MANETs are often constructed to support cooperative work in environments without network infrastructures. In such environments, there is generally a correlation among data items held by nodes. For example, in a MANET constructed by rescue parties at a disaster site, the rescue workers equipped with portable computers share the data obtained by other workers in order to work efficiently. In this situation, the rescue workers often access particular data items at the same time as follows: (i) photographs of damage spaces and numerical data that show the extent of the damage are referred, (ii) data on the extent of the damage to different spaces are compared, and (iii) data on the extent of the damage are published as Web pages. In this case, the HTML file and photographs that comprise of a Web page are collectively accessed.

The probability that the client collectively accesses a set of data items represents the strength of the correlation among those data items; the stronger the correlation, the higher the probability that a particular set of data items will be accessed together. The probability differs for each set of data items.

When a client collectively accesses correlated data items, the following two cases occur: (i) the client accesses a set of correlated data items by submitting multiple access requests at the same time, as described in the above examples, and (ii) the client accesses a set of correlated data by submitting access requests with some intervals. An example of the latter case is a situation in which each Web page is a data item and users access several Web pages that are linked to each other. In this paper, for simplicity, we assume the former case.

In a real environment, the correlation among data items can be usually known by recording the access log at each node and periodically estimating it.

3 Replica Allocation for Correlated Data

In this section, we describe the basic system model, then propose three replica allocation methods that take data correlation into account.

3.1 System model and approach

The system environment is assumed to be a MANET in which sensor nodes access data items held by other nodes as the originals. Each node creates replicas of the data items, and maintains the replicas in its memory space. When a node issues an access request to a data item, the request is successful in either case where: (i) the node holds the original/replica of the data item, or (ii) at least one node which is connected to the request-issuing node with a one-hop/multihop link holds the original/replica. Thus, first, the request-issuing node checks whether 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. Then, if it receives a reply from another node(s) that holding the original/replica of the target data item, the request is also successful. Otherwise, the request fails.

In this paper, we assume that a node simultaneously issues access requests for correlated data items. We define that the requests for the correlated data items are successful only when all of the requests are successful; that is, all the requests for the correlated data items fail even if one of the requests fails. In the following, for the sake of simplicity, we only deal with the case in which *two* correlated data items are requested at the same time. However, our proposed methods can be directly applied to the case in which more than two data items are requested simultaneously.

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

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

- 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 node as the original. The data items are not updated.

- Each node has memory space of C data items for creating replicas, excluding the space for the original data item that the node holds.

- The strength of correlation between each two data items is known, and does not change. p_{i-jk} denotes the frequency that node M_i accesses two data items j and k $(i = 1 \cdots m, j, k = 1 \cdots n)$ at the same time, i.e., the strength of correlation. Here, $p_{i-jk} = p_{i-kj}$, $p_{i-jj} = 0$

Since nodes move freely, it is impractical to determine the optimal allocation among all possible combinations of replica allocation every time when the network topology changes due to the node migration. Thus, we take a heuristic approach in which replicas are relocated in a specific period, T (*relocation period*). At every relocation period, replica allocation is determined based on the correlation among data items at each node, and the network topology at the moment.

3.2 Replica allocation methods

In [4], based on the above approach except for the assumption that there is the correlation among data items, we proposed three replica allocation methods that take into account the access frequency to each data item and the network topology. These three methods differ in emphasis placed on the two factors:

SAF (Static Access Frequency): Only the access frequency to each data item is taken into account.

DAFN (Dynamic Access Freq. and Neighborhood): The access frequency to each data item and the neighborhood among nodes are taken into account.

DCG (Dynamic Connectivity based Grouping): The access frequency to each data item and the whole network topology are taken into account.

In this paper, we extend these three methods to take into account the correlation among data items. We call the three extended methods, the C(Correlation)-SAF, C-DAFN, and C-DCG methods. In these extended methods, a data priority of each data item is defined based on the correlation among data items, and data items are replicated at nodes with the data priority.

In the following, we describe how to determine a data priority and the details of the extended methods.

3.2.1 Data Priority

The priority of data item D_j at mobile node M_i is determined as follows:

1. For every data item D_j, its access frequency is calculated as $F_{ij} = \sum_{k=1}^{n} p_{i\text{-}jk}$. Then, among possible combinations of $\lfloor n/x \rfloor$ data items with the highest access frequency, two data items with the strongest correlation are chosen. Here, n is the total number of data items, and x is a constant.
2. Among data items that are not yet chosen, a new data item, D_{j0}, is chosen where D_{j0} has the strongest correlation to data items which are already chosen or held by M_i as originals. Here, the correlation is calculated as a summation of correlations between D_{j0} and the data items.

   This process is repeated until all items are chosen.
3. The data priority is determined as follows: The priority of original data items that M_i holds is the highest, and that of the two items chosen at step 1 is the next. The priority of remaining items is determined according to the order in which each item is chosen at step 2.

3.2.2 The C-SAF method

Under the C-SAF method, each node allocates replicas of C data items according to the data priority. At the time of replica allocation, a node may not connect to another node which has the original or a replica of a data item that the node should allocate. In this case, the memory space for the replica is retained free. The replica is created when a data access to the data item succeeds or when the node connects to another node which has the original or the replica at a relocation period.

Under the C-SAF method, nodes do not need to exchange information with each other for replica allocation. Moreover, replica relocation does not occur after each node allocates all necessary replicas. As a result, this method allocates replicas with low overhead and low traffic. On the other hand, since each node allocates replicas based on only the access frequencies to data items and the data correlation, nodes with the same access characteristics allocate the same replicas. Due to the replica duplication, this method gives low data accessibility.

3.2.3 The C-DAFN method

To solve the problem with the C-SAF method, the C-DAFN method eliminates the replica duplication among neighboring nodes. Since the neighboring status changes as nodes move, the C-DAFN method is executed at every relocation period. The algorithm of this method is as follows:

1. At a relocation period, each node broadcasts its node identifier and information on access frequencies to data items. After all nodes complete the broadcasts, every node shall know its connected nodes from the received node identifiers.
2. Each node preliminarily determines the allocation of replicas based on the C-SAF method.
3. In each set of nodes that are connected to each other, the following procedure is repeated in the order of the breadth-first search from the node with the lowest suffix (i) of the node identifier (M_i). When there is duplication of a data item (original/replica) between two neighboring nodes, and if one of them is the original, the node that holds the replica changes it to another replica. If both of them are replicas, the node whose original data items have weaker correlation with the data item than those of the other node changes the replica to another replica. When changing the replica, among data items whose replicas are not allocated at either of the two nodes, a new replicated data item is selected where the data priority of the item at the node that changes the replica is the highest among the possible items.

At a relocation period, a node may not connect to another node that has the original or a replica of a data item which that node should allocate. In this case, the memory space for the replica is temporarily filled with one of replicas that has been allocated since the previous relocation period but is not currently selected for allocation. This temporary allocated replica is chosen from among the possible replicas where the data priority of the replica (data item) is the highest among them. When an attempt at data access to the data item whose replica should be allocated succeeds, the memory space is filled with the proper replica. This behavior is the same for the C-DCG method.

Since the C-DAFN method eliminates replica duplication, the accessibility to data is expected to be greater than that for the C-SAF method. However, the C-DAFN method does not completely eliminate replica duplication among connected nodes because it only executes the elimination process among neighboring nodes. Here, "connected nodes" denote the nodes which are connected to each other by wireless links of one or more hops. Furthermore, both the overhead and the traffic are higher than under the C-SAF method because at each relocation period, nodes exchange information and relocate replicas.

3.2.4 The C-DCG method

The C-DCG method shares replicas in larger groups of nodes than the C-DAFN method. In order to share replicas effectively, each group should be stable, i.e., the group is not easily divided due to changes of network topology. From this perspective, the C-DCG method creates groups of nodes that are *biconnected components*[1] in a network. By grouping nodes as a biconnected component, the group is not divided even if one node disappears from the network or one link is disconnected.

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

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

2. In each set of nodes that are connected to each other, from the node with the lowest suffix (i) of node identifier (M_i), an algorithm to find biconnected components is executed. Then, each biconnected component is put to a group. If a node belongs to more than one biconnected component, i.e., the node is an *articulation point*, it belongs to only one group in which the corresponding biconnected component is first found in executing the algorithm.

3. In each group G, the data correlation between data items D_j and D_k in the group is calculated as a summation of data correlations between the two items at all nodes in the group ($\sum_{M_i \in G} p_{i\text{-}jk}$). Then, in the same way as in section 3.2.1, the priority of each data item in the group is determined. When there is more than one original data item in the group, the priority of original data items is determined in descending order of access frequencies of the group to the items. Here, the access frequency of the group to each item is calculated as a summation of access frequencies of all nodes in the group to the item ($\sum_{M_i \in G} \sum_{k=1}^{n} p_{i\text{-}jk}$)). These calculations are performed by the node with the lowest node identifier suffix in the group.

4. In each group, replica allocation is determined as follows: If the group consists of one node, the replica allocation is determined by the C-SAF method. Otherwise, according to the data priority in the group, replicas of data items are allocated until memory space of all nodes in the group becomes full. Here, replicas of data items which are held as originals by nodes in the group are not allocated. Each replica is allocated at a node whose original data items have the strongest correlation with the data item among nodes that have sufficient free memory space to create it.

5. After allocating replicas of all types of data items, if there is still free memory space at any of the nodes in the group, replicas are allocated according to the data priority until the memory space is full. Each replica is allocated at a node whose original data items have the strongest correlation with the data item among nodes that have free memory space to create it and do not hold the replica or its original.

Since this method shares many types of replicas in larger groups of nodes with high stability, the data accessibility is expected to be higher. Moreover, by allocating replicas of data items with strong correlation to the original data items at each node, this method can reduce undesirable situations in which the node can access a data item/replica but cannot access its correlated data item.

However, both the overhead and the traffic are higher than the other two methods because at each relocation period, nodes exchange information and relocate replicas across a wide range.

4 Simulation Experiments

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

4.1 Simulation model

Mobile nodes exist in a size 50×50 flatland. Both the number of nodes and types of data items in the whole network are 40 ($\boldsymbol{M} = \{M_1, \cdots, M_{40}\}$, $\boldsymbol{D} = \{D_1, \cdots, D_{40}\}$). M_i ($i = 1, \cdots, 40$) holds D_i as the original. Each node randomly moves in all directions, and the movement speed is randomly determined from 0 to 1. The radio communication range of each node is a circle with

Table 1. Parameter configuration.

Parameter	Value	
T	256	(1∼1024)
R	7	(1∼19)
C	10	
x	2	

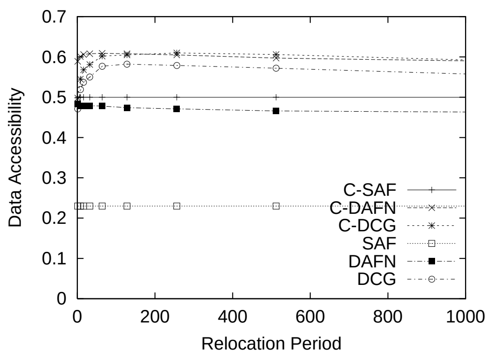

Figure 2. Relocation period and data accessibility.

the radius R. All nodes have the same access characteristics ($p_{i_1\text{-}jk} = p_{i_2\text{-}jk}$, $\forall\ M_{i_1}, M_{i_2}$). The probability that two items D_i and D_j will be simultaneously requested at a unit of time, i.e., the data correlation, is expressed by the i-j element, P_{ij} ($\sum_{i=1}^{n} \sum_{j=i}^{n} P_{ij} = 1$, and $P_{ij} = P_{ji}$), in the access probability matrix (a 40×40 symmetric matrix). For the sake of simplicity, it was assumed that $P_{ii} = 0$ for each item D_i. All elements except for P_{ii} were assigned 0 or a positive value determined randomly from 0 to 1. Non-zero elements were randomly determined so that the ratio of them to all elements was 0.2.

Table 1 shows parameters and their values used in the simulation experiments. There, x is a constant used for determining data priority. Each parameter is basically fixed to a constant value, but could change within the range represented by the parenthetic values in one of the simulation experiments.

In all simulation experiments, we initially placed each node at a random position and examined the average data accessibility and the total traffic of each of the three extended methods during 10,000 units of time. Here, we define the average data accessibility as the ratio of successful correlated two access requests to all correlated two requests, and the traffic as the total hop count of data transmission for allocating/relocating replicas. For the purpose of comparison, we also evaluate the performance of the non-extended methods proposed in [4]. Under these methods, the access frequency, F_{ij}, of node M_i to data item D_j was calculated by $F_{ij} = \sum_{k=1}^{n} p_{i\text{-}jk}$.

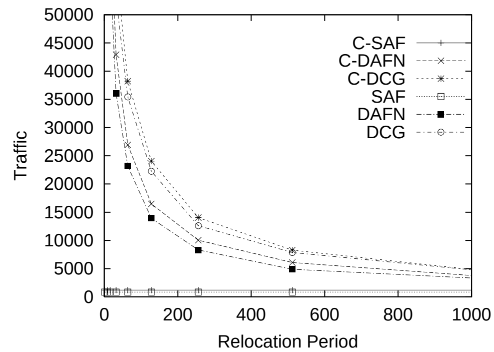

Figure 3. Relocation period and traffic.

4.2 Effects of the relocation period

First, we examine the effects of the relocation period on each of the three methods. Figures 2 and 3 show the simulation results. In both figures, the horizontal axis indicates the relocation period, T, while the vertical axis indicates the data accessibility and the traffic, respectively.

From Figure 2, it is clear that the three extended methods proposed in this paper give higher data accessibility than the corresponding non-extended method. The C-DAFN method provides the highest data accessibility when the relocation period is short, and the C-DCG method does that in other cases. Under the four methods (C-DCG, C-DAFN, DCG, and DAFN) that share replicas among nodes, under the condition where the relocation period is very short, the data accessibility decreases as the relocation period shortens. In our simulation experiments, since nodes move randomly, the network topology frequently changes. Thus, when the relocation period is very short, the changes in network topology are detected with high sensitivity, and under the four methods, replicas are shared among nodes that are temporarily connected by unstable wireless links. This degrades the access probability in the four methods.

Figure 3 shows that the extended methods proposed in this paper increase the traffic. The extended methods take into account the correlation with original data items that each node holds when determining replica allocation. Thus, replicated data items change as the neighboring nodes change, and this causes high traffic volumes. Among the three extended methods, the C-DCG method gives the highest traffic.

4.3 Effects of the radio communication range

We examine the effects of the nodes' radio communication range on each of the three methods. Figures 4 and 5 show the simulation results. In both figures, the horizontal axis indicates the communication range, R, while the vertical axis indicates the data accessibility and the traffic, respectively.

Figure 4 indicates that the C-DAFN method gives the highest data accessibility when the radio communication range is short, and the C-DCG method does so in other cases. This happens because when the radio communica-

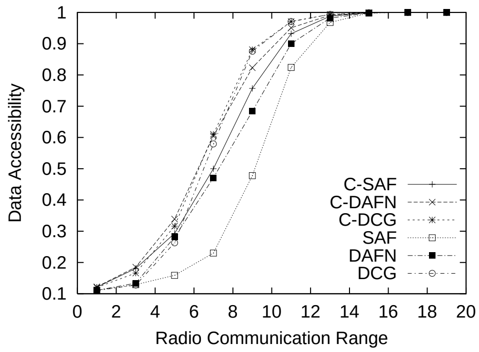

Figure 4. Radio communication range and data accessibility.

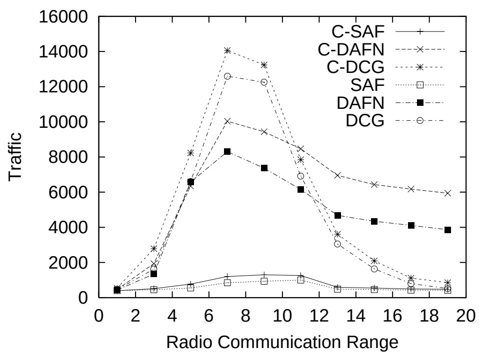

Figure 5. Radio communication range and traffic.

tion range is short, under the C-DCG method, replicas are shared among nodes that temporarily form a bi-connected component with unstable wireless links. On the contrary, as the radio communication range increases, the effectiveness of sharing data items in a stable group becomes conspicuous. When the radio communication range is very long, the difference between the C-DCG method and the DCG method is small. This is because replicas of nearly all data items are shared in a group that consists of a large number of nodes and has high stability, and thus, the effectiveness of considering data correlation becomes low.

Figure 5 shows that as the radio communication range increases, the impact of considering data correlation on traffic becomes more conspicuous under the C-DAFN method. This is because when the radio communication range is long, the number of neighboring nodes is large, and thus, this causes the increase of traffic under the C-DAFN method.

5 Conclusion

In this paper, we have assumed an environment in which nodes simultaneously issue access requests to correlated data items in sensor MANETs. We have proposed three replica allocation methods that are actually extensions of our previously proposed methods to adapt such an environment.

The simulation results show that the extended methods proposed in this paper give higher data accessibility than the corresponding non-extended methods. The results also show that the extensions cause traffic to increase under the C-DAFN and C-DCG methods.

As part of our future work, we plan to address data replication in an environment where access requests for correlated data items are issued with some intervals. We also plan to extend our proposed methods to adapt data updating. To do so, the approaches in [5] could be applied.

Acknowledgment

This research was supported in part by "The 21st Century Center of Excellence Program" and Special Coordination Funds for promoting Science and Technology of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

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

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey o sensor networks," IEEE Communication Magazine, Vol.40, No.8, pp.102-114, 2002.

[3] 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.

[4] T. Hara, "Effective replica allocation in ad hoc networks for improving data accessibility," Proc. IEEE Infocom 2001, pp.1568-1576, 2001.

[5] T. Hara, "Replica allocation methods in ad hoc networks with data update," ACM-Kluwer Journal on Mobile Networks and Applications, Vol.8, No.4, pp.343-354, 2003.

[6] C. Intanagonwiwat, R. Govindan, D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," Proc. Mobicom 2000, pp.56-67, 2000.

[7] D.B. Johnson, "Routing in ad hoc networks of mobile hosts," Proc. IEEE Workshop on Mobile Computing Systems and Applications, pp.158-163, 1994.

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

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

[10] 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.

[11] C.E. Perkins and E.M. Royer, "Ad hoc on demand distance vector routing," Proc. IEEE Workshop on Mobile Computing Systems and Applications, pp.90-100, 1999.