Managing Data Replication in Mobile Ad-Hoc Network Databases Using Content Based Energy Optimization

Mobarak Abaker Adam Hassan\textsuperscript{1}, Nayyar Ahmed Khan\textsuperscript{2} and Mohammad Ahmad Mohammad Nasim\textsuperscript{3}

\textsuperscript{1}Assistant Professor, Department of Computer Science, Shaqra University, Saudi Arabia. Email: mhassan@su.edu.sa
\textsuperscript{2}Lecturer, Department of Computer Science, Shaqra University, Saudi Arabia. Email: nayyarkhan@gmail.com
\textsuperscript{3}Lecturer, Department of Computer Science, Shaqra University, Saudi Arabia. Email: nasim.ahmad0033@gmail.com

Article Received: 09 October 2017 Article Accepted: 27 November 2017 Article Published: 14 December 2017

ABSTRACT

A Mobile Ad-hoc Network (MANET) is a collection of wireless autonomous nodes without any fixed backbone infrastructure. All the nodes in MANET are mobile and power restricted and thus, disconnection and network partitioning occur frequently. In addition, many MANET database transactions have time constraints. Data Replication technique for real-time Ad-hoc Mobile databases (Dream) is proposed that addresses all those issues. [1] It improves data accessibility while considering the issue of energy limitation by replicating hot data items at servers that have higher remaining power. It addresses disconnection and network partitioning by introducing new data and transaction types and by considering the stability of wireless link. It handles the real-time transaction issue by replicating data items that are accessed frequently by firm transactions before those accessed frequently by soft transactions. However there are certain problems that may arise in the Dream concept and in order to overcome the issue proposed as a fail back in Dream the paper represents a better approach so as to handle the scenario. This approach is a schema presented in order to handle the content based system replica for the MANET and provide a better congestion free and energy efficient distribution amongst the servers. In MANETs, mobile nodes can move randomly and autonomously. They form a temporary communication network without any prior infrastructure.

Keywords: Cloud Computing, Amazon, Salesforce, EC2 and IBM Bluemix.

1. INTRODUCTION

MANET is a collection of wireless autonomous nodes that may move unpredictably, forming a temporary network without any fixed backbone infrastructure [2]. All the nodes in MANET are mobile and, thus, restricted by power. These nodes not only can communicate with nodes that are within their communication ranges, but also can communicate with nodes that are outside their transmission ranges using multi hop communication. Since no fixed infrastructure is required, they fit well in military, rescue operations and sensor networks [3]. MANET has dynamic topology as mobile nodes move around or even enters and exit the network. Every mobile nodes of a MANET work as a router to discover and maintain routes, and forward packets to other nodes. In many ad hoc networks, each node is powered by a battery and has a limited energy supply. Over time, various nodes will deplete their energy supplies and drop out from the network.

Unless nodes are replaced or recharged, the network will eventually become partitioned. In a large network, relatively few nodes may be able to communicate directly with their intended destinations. Replication of hot data items before such a partitioning occurs, improve data accessibility in MANET. With the current MANET data replication protocols which are based on Update broadcast/ Connection broadcast methods consistency can be assured but energy saving cannot. Energy-preserving replication strategies for MANET have received ongoing attention. In general incorporating information about nodes battery levels in replication can help to preserve energy. In this paper, we are presenting On Demand Update of Replica in MANET based upon estimate of the remaining lifetime of each node. In our scheme (On Demand Update of Replica) host will update data items at multiple nodes by invalidating their replicas On Demand instead of every updating. Moreover many
applications in this environment are time critical and, hence, their transactions should be executed not only correctly, but also within their deadlines. In a distributed database system, data are often replicated to improve reliability and availability. However one important issue to be considered while replicating data is the correctness of the replicated data. Since nodes in MANET are mobile and have limited energy, disconnection may occur frequently, causing a lot of network partitioning. Data that is available in mobile hosts in one partition cannot be accessed by mobile hosts in another partition. The issues related to data replication in MANET databases are as follows:

1. **Server Power Consumption**: Servers in MANET run on battery power. If a server has low power remaining and if it is replicated with many frequently accessed data items, then the frequent data access requests for these hot data will drain its power.

2. **Server mobility**: Due to their mobility, servers might sometimes move to a place where they could not be reached by any other server or client.

3. **Client mobility**: Clients that query servers for information are also mobile. Clients usually send their transactions to the nearest servers to get a quick response. The decision to replicate data items at a particular server may be based on the access frequencies of data items from that specific server. Hence, the decision to replicate data items at appropriate servers must be dynamic and based on the current network topology.

4. **Client Power**: Clients in MANET also run on battery power. If a client waits too long for its transactions’ results, it might lose its power rapidly. The replication technique should be able to replicate data items at appropriate servers in such a way that a client might be able to access its requested data items from its nearest server which has the least workload.

5. **Time-critical applications**: Many MANET applications, like rescue networks, military operations, are time-critical. Data replication should be used to improve data accessibility and Performance of the system, thereby reducing the time to execute transactions. Transactions with short deadlines should be sent to the nearest servers that have the least workload for their execution. These transactions should also be executed before other transactions that have longer deadlines.

6. **Frequent Disconnection and Network Partitioning**: Disconnection and network partitioning is a severe problem in MANET as servers in one partition that hold required data items cannot provide services to other clients and servers in different partitions.

7. **Update transactions**: If there are no constraints in storage space and if there are read-only transactions, data accessibility and performance of MANET databases can be increased by fully replicating the database in all the servers. However, maintaining consistency in a fully replicated database is a major issue when there are frequent update transactions.

### 2. RELATED WORKS

A data replication technique that replicates data items based on their access frequencies and the current network topology is proposed in [4]. Hot data are replicated before cold data items. If the access characteristics of data
items are similar, there could be replica duplications at many mobile nodes. Hence, two other techniques to reduce replica duplication between mobile nodes are proposed in [4]. They also detect network partitioning and replicate hot data items before such a partitioning occurs to improve data accessibility. However in those techniques, when there is a replica duplication between any two connected mobile nodes, one of the duplicate replicas is replaced by another hot data item, irrespective of how high the access frequency of the replaced data item is or how low the access frequency of the new data item is.

A data replication technique that replicates data items at multiple nodes and employs quorum based strategies to update and query information is proposed in [5]. It sends the update information to nodes in such a way that other nodes while querying for this update information gets the most updated information, and thereby, reducing inconsistency and dirty read transactions. It determines the time to send the updates, where to send them and which nodes to query for the information in such a way that it mitigates the impacts of network partitioning. Another way of disseminating the update information to mobile nodes is by data broadcasting as proposed in [6]. This replication technique delays the updates to replicas for performance and bandwidth considerations. It also determines what updates to broadcast and when to broadcast them in an efficient manner.

A set of protocols that use a gossip-based multi-cast protocol to probabilistically disseminate updates in a quorum system is proposed in [7]. It achieves high reliability even when there are large concurrent update and query transactions. A new metric for evaluating wireless link robustness that is used to detect network partitioning is proposed in [7]. Its decision to replicate data items is taken not only at the time of detecting network partitioning, but also during the time when the wireless connections become bad in terms of reliability, bandwidth and delay. Roam [8] is a replication technique that attempts to provide data availability to mobile hosts irrespective of the mobility of the hosts. It models the mobility of hosts by grouping them into wards and determines periods of motion of the mobile hosts. Ward masters are elected to provide communication across wards, but hosts belonging to the same ward may directly communicate with each other. Roam maintains consistency of replicas across the network, irrespective of the locations of movements’ different hosts. None of the above replication techniques addresses the issues related to real-time database transactions and mobile hosts’ power limitation. It should also be noted that network partitioning might occur not only due to mobile hosts’ mobility, but also due to battery power drainage of some mobile hosts.

3. NON ECCENTRIC - MANET DATABASE ARCHITECTURE WITH MULTIPLE SERVER POINTS AND CLIENT POINTS

Depending on communication strength, computing power, disk and memory storage size, and mobile hosts can be classified into two groups: Clients and Servers. Clients are equipped with reduced memory and computing capabilities. They store only the Query Processing module of the Database Management System (DBMS) that allows them to submit transactions to appropriate servers and receive results. Servers are equipped with higher memory and computing capabilities; they store the complete DBMS. For example, soldiers in battlefields, who
are equipped with handy portable computers like PDA’s and smart phones, could be considered as clients, while tanks and Humvee equipped with high end portable computers like laptops could be considered as servers. This paper assumes a decentralized architecture, where clients are free to communicate (single-hop or multi-hop) and submit their transactions to any of the available servers in the network as shown in Figure 1. This architecture does not place reliance on any centralized server and, thus, improves system resilience by avoiding a single point of failure.

![Figure 1: Non Eccentric MANET Database Architecture with multiple client and server access points (SAP’s)](image)

4. PROPOSED MANET DATA REPLICATION TECHNIQUE CONTENT BASED DATA REPLICATION

A data replication for MANET databases, is proposed that extends the techniques proposed in [6] that have briefly been reviewed in [2] to consider additional issues including mobile hosts’ power limitation, real-time database transactions as well as various data and transaction types that exist in many MANET applications. The active data items at the server are having high remaining power. It handles the real-time transaction issue by giving a higher priority for replicating data items that are accessed frequently by perfect transactions than those accessed frequently by temporal transactions. It addresses disconnection and network partitioning by introducing new data and transaction types and by determining the stability of wireless links connecting servers.

The remaining energy of connecting servers is also used to measure their link stability. Each server in the design proposed holds the original copy of a data item is termed its primary copy server and the other servers that hold the replicas of the data item are termed its secondary copy servers. The scheme addresses the replica synchronization issue by maintaining two timestamps that indicate when a particular data item is updated in its primary and secondary copy servers.

The main parts for the replication mechanism are as: first part determines the data items to be replicated by obtaining the need for the items that are to be replicated and the servers in which they have to be replicated. The second part determines how the allocated replicas can be accessed for transaction processing based on their data and transaction types. The third part identifies the way to synchronize the replicas.
A. Data Types

Data items are classified into read-only and read-write data items. Read-write data items can be further classified into two types: Temporal and Persistent. The former are those data items the values of which are valid only for a certain period of time. The location of a soldier and the location of an enemy are examples of temporal data items as they move frequently in a battlefield. However, persistent data items are valid throughout their existence in the database. All read-write data items can be further classified into periodic and aperiodic update data items. Periodic update data items are those data items that are updated periodically at fixed intervals of time. For example, the location of a soldier may be updated frequently at a constant interval of time. Aperiodic-update data items are those data items that are updated at random intervals of time. For example, the location and information about an enemy may be updated at random time intervals.

1. Computing Access Frequencies based on Data and Transaction Types

Access frequency of a data item \( d \) at a particular server \( s \), \( \text{Access Frequency} \), is the number of times that \( d \) is accessed at \( s \). From the access logs, similar to [6], the access frequency of each data item at each server is computed. In addition, in DREAM, the numbers of times a data item is accessed by Firm, Soft, MRV and Non-UCV transactions at all servers are computed. These are computed only once by all the servers when running the replication algorithm for the first time. These access frequencies are then further calculated using a weight factor based on the data and transaction types presented in Sections 4.1 and 4.2. The resulting access frequencies are thus the weighted ones to reflect replication prioritization as shown in the following sections.

a) Firm and Soft Transactions: if a data item is accessed more often by firm transactions than another data item is, then the former data item is given a higher priority to be replicated than the latter. This is to reduce the number of transaction aborts since firm transactions must be aborted if they missed their deadlines. Firm transactions’ execution time can decrease considerably if their required data items reside in the servers to which the requests were sent. The replication priority is set by assigning weighted access frequencies to data items that are accessed by firm transactions using the below formula where \( \text{Access frequency} \) Firm is the number of times data item \( d \) is accessed by firm transactions at server \( s \).

\[
\text{Access Frequency}_{d} = \text{Access Frequency}_{d} + \text{Access Frequency}_{d, \text{Firm}} * \left( \frac{\text{Access Frequency}_{d, \text{Firm}}}{\text{Access Frequency}_{d}} \right)
\]
2. Read Transactions

If the requested data item is a read-only data item, it can be accessed from any server that holds it. Similarly, if the initiated transaction is an OD transaction, it can be accessed from any server that holds it. For both of these two cases, if the requested data item is available at more than one server, the decision to choose an appropriate server is based on the real time transaction type. A firm transaction is sent to the nearest server with the least workload, while a soft transaction is sent to the highest energy server with the least workload, for transaction processing. The objective is to reduce the number of transaction aborts and, at the same time, balance the energy consumption distribution among servers.

If the requested transaction is a MRV transaction, the requested data item should be accessed from the server that has its most recent value among all the servers that hold it. If the requested transaction is a MRVP transaction, the requested data item should be accessed from the server that has its most recent value among all the servers in its network Partition that holds it. Based on the Local_Update_Timestamps of all servers s, the most recent value of data item d can be determined. A periodic update data item is updated once every certain period of time called its update frequency. Hence, a MRV or a MRVP transaction accessing a periodic update data item can access it from any server that has updated it during its last known update time interval. For example, consider a periodic data item d that is updated every one hour. A server s has the most recent value of d if the difference between the current time (Tnow) and the last update timestamp of d in s (Local_Update_Timestamps) is less than its update frequency (Frequency_Updateds), which is one hour in this example.

3. Write Transactions

If the transaction is an update transaction and if the coordinating server is connected to the primary copy server of the requested data item, the update transaction is forwarded to the primary copy server for transaction processing. If the coordinating server holds a replica of the requested data item, and if the transaction is not an UCV transaction, the local replica is also updated as further read requests for that data item can be accessed directly from the local replica. However, if the coordinating server is not connected to the primary copy server, and if the transaction is not an UCV transaction, the update transaction is forwarded to the server that holds the requested data item. If there is more than one server that holds the requested data item, a firm transaction is sent to the nearest server that has the least workload and a soft transaction is sent to the highest energy server that has the least workload for transaction processing.

However, if the coordinating server is not connected to the primary copy server as in this case, the transaction cannot be executed successfully. Hence, the coordinating server will try to connect to the primary copy server unless the deadline of this transaction has expired, in which case the transaction is aborted. The deadline here means the first deadline if the transaction is firm and the second deadline if the transaction is soft.
How to Synchronize Replicas?

Every time during the relocation period, the primary copy server of a data item tries to synchronize its data item with other connected servers that hold its replica. The primary copy server requests for the last updated timestamps from all replicas. Based on the update timestamp of the primary copy and the update timestamps of the replicas, the primary copy server determines if there is any other server that has a more recent value of its data item. If such a server exists, the new value of the data item is synchronized with all other servers. However, a server that is disconnected from the network during the relocation period cannot synchronize its data item. Even if the disconnected server has the most recent value of the data item, the primary copy server cannot determine it since the former is disconnected. Hence, it will only try to synchronize the data item during the next relocation period.

5. REPLICA RELOCATION

As mentioned above, optimal replica allocation cannot be determined in MANETs due to the mobility of mobile hosts. Therefore, we embrace the following heuristic approach:

- Replicas are relocated in a specific period (relocation period).
- At every relocation period, replica allocation is determined based on access frequency from each mobile host to each data item and the network topology at the moment. Based on this approach, in [5] we proposed three replica allocation methods that differ in the emphasis placed on access frequency and network topology.

A. Static Access Frequency (SAF)

Each mobile host allocates replicas of data items in descending order of the access frequencies within the limit of its own memory space. 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 each 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 access frequencies to data items, mobile hosts with the same access characteristics allocate the same replicas. However, 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, SAF provides low data accessibility when many mobile hosts have the same or similar access characteristics.

B. Dynamic Access Frequency and Neighborhood (DAFN) to solve the problem with the SAF method, first the DAFN method determines replica allocation in the same way as SAF. If there is replica duplication of a data item between two neighboring mobile hosts, the mobile host with the lower access frequency relative to the data item replaces the replica with another replica. This procedure is performed at every relocation period. Since the DAFN method eliminates replica duplication and more kinds of replicas can be shared among neighboring mobile hosts, data accessibility is expected to be higher than in the SAF method. However, DAFN cannot eliminate replica duplication among mobile hosts connected by two or more hop links. Moreover, both the overhead and the traffic for replica relocation are higher than the SAF method.

C. Dynamic Connectivity based Grouping (DCG): The DCG method shares replicas in larger groups of mobile hosts than DAFN. At every relocation period, each mobile host broadcasts its host identifier. After all mobile hosts complete the broadcasts, every host knows the connected mobile hosts and the network topology from the
received host identifiers. In each set of mobile hosts connected to each other, the mobile host with the lowest host identifier suffix (I) executes an algorithm to find disconnected components with the network topology known by received messages. Even if a mobile host belongs to more than one disconnected component, it can only belong to one group in which the corresponding disconnected component was found first. By grouping mobile hosts as disconnected components, the group is not divided even if one mobile host disappears from the network or one link is disconnected in the groups. Thus, it is assumed that the group has high stability. Next, replicas of data items are allocated on mobile hosts in each group in descending order of the access frequencies in the group. More specifically, in each group, the mobile host with the lowest host identifier becomes the coordinator of the group, who calculates the group’s access frequency to each data item as a summation of access frequencies of mobile hosts in the group to the item. Then the coordinator determines replica allocation in the group according to the calculated access frequencies. Compared with the DAFN method that shares replicas among neighboring hosts, the DCG method shares replicas in larger groups of mobile hosts that have high stability. Thus, data accessibility is expected to be higher. However, both the overhead and the traffic are higher than the other two methods because at each relocation period, mobile hosts exchange information and widely relocate replicas.

6. CONSISTENCY MANAGEMENT

In [9], we assumed an environment where each data item is updated by the mobile host that holds the original at inconstant intervals and proposed two cache invalidation methods that reduce the number of accesses to old replicas whose original has been updated. We call them the update broadcast and connection rebroadcast methods, respectively. In the following, we describe the details of the two methods A. Update broadcast method [10].

In the update broadcast method, a mobile host holding an original data item broadcasts an invalidation report to connected mobile hosts every time it updates the data item. The invalidation report includes a data identifier and update time (time stamp). When a mobile host receives an invalidation report, it refers to its own time stamp table and checks whether the replica held by the host is valid. More specifically, first the mobile host compares the time stamp in the received invalidation report with the corresponding data item in its own time stamp table. If the former is larger, the host updates the time stamp in its own time stamp table to the received invalidation report. At the same time, the host transmits the received invalidation report to its neighboring mobile hosts. Furthermore, if the host holds an invalid replica of the corresponding data item, the host discards the replica from its own cache. On the other hand, if the time stamp in the received invalidation report is identical to the time stamp table, i.e., the host receives the same invalidation report again, it does not transmit the received invalidation report to its neighboring mobile hosts and discards the report. In this method, the traffic caused by broadcasting invalidation reports is light since mobile hosts broadcast only when updating the original data items. Mobile hosts connected to the host holding the original data item can keep the time stamp of the most recent corresponding data item.

B. Connection rebroadcast method: Similar to the update broadcast method, in the connection rebroadcast method, a mobile host holding an original data item broadcasts an invalidation report every time it updates the data item. In
addition, every time two mobile hosts newly connect with each other, they rebroadcast invalidation reports that they have already received.

1) When two mobile hosts, MI and Maj (I < j), newly connect with each other, the mobile host with larger suffix (j) of host identifier (Maj) transmits its own time stamp table to the other one.

2) Mobile host MI compares the entry for each data item in its own time stamp table with the time stamp table received from mobile host Maj and updates its own time stamp table. Then, the following processes are executed:

- MI broadcasts invalidation reports for data items whose time stamps held by MI are smaller than the ones held by Maj to its connected mobile hosts, except to Maj.
- MI sends the information on the updated time stamps for data items whose time stamps held by Maj are smaller than the ones held by MI to Maj. Then Mj broadcasts invalidation reports for these items to its connected mobile hosts, except to Mi.

3) Mobile hosts that receive invalidation reports invalidate their replicas in the same way as the update broadcast method. In this method, invalidation reports spread among a large number of mobile hosts, and cache 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 update broadcast method. However, when network topology frequently changes, the traffic caused by broadcasting invalidation reports is much higher than the update broadcast method due to the frequent rebroadcast of reports.

C. Extensions

The methods proposed in [10] can reduce the number of accesses to old replicas whose originals have been updated, although they cannot improve data accessibility. In [12], we proposed two updated data dissemination methods to efficiently update old replicas and reduce the number of accesses to old replicas as well as to improve data accessibility. These methods disseminate data items of the most recent version instead of invalidation reports.

7. LOCATION MANAGEMENT

In [9], we discussed the techniques to manage the locations of data items and replicas and efficiently forward Access requests to the locations. In the following, we call these techniques location management methods. To efficiently manage the locations of data items, a key issue is predicting the locations of data items that dynamically change. Here, location changes are caused by two factors: replica relocations and network topology changes due to host mobility. To predict the locations of data items, we use replica allocation information at every relocation time and the logs of past data accesses.

In this section, among the location management methods proposed in [9], we introduce the Access Log (AL) method, which can be adopted for any replica allocation methods, and the Group Management (GM) method,
which is specialized for the DCG method in [5]. Noted that the GM method can be applied to most other replication schemes based on clusters that hierarchically manage data locations.

A. AL method
In this method, each mobile host holds an access log (AL) table that consists of pairs of a data identifier and a list of host identifiers that represents holders of the data item corresponding to the data identifier.

1) Recording location information: Suppose that a mobile host issues an access request to data item Do and the request succeeds by accessing the original or its replica held by mobile host Mi. In this case, the request issued host inserts data identifier MI at the top of the list of host identifiers corresponding to do. If MI already exists in the list, the old one is deleted from it. Here, the size of each list is limited to L and the last one is deleted if the size exceeds L. Every time a relocation period comes, each mobile hosts Discards all the information in its own AL table.

8. FUTURE DIRECTIONS
As mentioned above, replica relocation, consistency management, and location management are significant issues for data replication in MANETs. However, several other issues still remain to be addressed to put the data replication techniques into practice. In this section, we describe these remaining issues as future directions.

A. Power Consumption
Since we consider the reduction of the power consumption of mobile hosts a significant issue in MANETs, network traffic has been chosen as a performance metric. However, although the reduction of traffic in the entire network can reduce the total power consumption among all mobile hosts, not every mobile host is guaranteed to survive for a long time. In other words, the power consumption of mobile hosts may become unbalanced, causing some to deplete their batteries and disappear from the network. Usually, such mobile hosts are at the central part of networks that have a large number of links with other hosts and relay many communication packets. Thus, the disappearance of these mobile hosts heavily damages the network.

In [11], [12], [13], the authors proposed routing protocols and caching strategies that consider the power consumption of mobile hosts in MANETs. The ideas in these conventional works can be applied to balance power consumption among mobile hosts for data replication in MANETs. We are currently considering new approaches that relocate replicas and route data requests by taking remaining batteries into account.

B. Consistency Levels and Their Management Protocols
In our work, we assume that replicas of a data item become invalid after the host, who is holding the original, updates it and consistency among replicas is maintained in the entire network. However, since mobile hosts frequently disappear from the network and partitions frequently occur in a MANET, such strict consistency management heavily deteriorates the data ( replica) availability. Moreover, many MANET applications do not
require such strict consistency. For instance, consider a situation where members of a rescue operation construct an ad hoc network in a disaster area. They are divided into several groups each of which is responsible for a particular region and the members in each group share information on their progress, i.e., the information is replicated at mobile hosts to deal with possible network divisions. In this situation, consistency among replicas must be strictly maintained in the same group, which is not required among replicas in different groups since the information being shared in a different group is only for reference.

Moreover, in our work consistency among replicas is protected by an optimistic approach in which each mobile host tentatively accesses a replica when it is not connected to the mobile host holding the original and validates the tentative access afterward. This approach may cause many transactions to remain tentatively committed for a long time as well as cascade aborts. In [14], [15], the authors proposed consistency management protocols based on a quorum system proposed for distributed systems. Although these protocols cannot strictly maintain consistency among replicas in the entire network, mobile hosts can immediately commit their transactions (database operations). We are currently addressing classification of consistency levels for database operations in MANETs according to application requirements and determining detailed protocols. Some of them employ a quorum system similar to the protocols in [14], [15].

C. Security

In our work we assume that the system environment is a mesoscale MANET in which mobile hosts (users) collaboratively work for the same purpose, such as rescues and military affairs. Thus, we do not consider security issues for data sharing in MANETs. However, if our proposed data replication technologies are used in a large and general public MANET, security issues must be addressed. In some conventional work [4], [13], security issues for routing in MANETs have been discussed. Some ideas in these Conventional work can be applied for data Replication in MANETs.

9. CONCLUSION

In this paper, we briefly explained our recent work addressing data replication issues in MANETs: replica relocation, consistency management, and location management. We also described a few prospects for future directions: power consumption, consistency levels and their management protocols, and security. We believe that data replication in MANETs will become a key technology in future mobile and ubiquitous computing environments.

REFERENCES


[9] Data Replication Issues in Mobile Ad Hoc Networks Takahiro Hara Department of Multimedia Engineering, Osaka University 1-5 Yamadaoka, Suita, Osaka 565-0871.


