

REALM: Replication of Data for a Logical Group Based MANET Database

Anita Vallur, Le Gruenwald*, and Nick Hunter

The University of Oklahoma
School of Computer Science
Norman, Oklahoma 73019 USA
ggruenwald@ou.edu

Abstract. Mobile Ad-Hoc Networks, or MANETs, provide communication between free-roaming mobile hosts without a fixed infrastructure. These MANETs operate under conditions of limited battery power and may also experience frequent network partitioning. This paper introduces a data replication technique called REALM (REplication of data for A Logical group based MANET database). REALM takes advantage of application semantics when determining where to place replicas of data items. It also tries to predict network partitioning and disconnection ahead of time and creates copies of data items accordingly. Experiments show that REALM increases the percentage of successful transactions with or without the presence of network disconnection and partitioning.

Keywords: Replication, data replication, algorithm, mobile databases, mobile ad hoc network databases.

1 Introduction

A mobile ad hoc network is a collection of wireless enabled mobile devices called mobile hosts. Every mobile host performs the functions of a router to enable communication between those mobile hosts that are not directly within each other's communication range. The unpredictable movement of mobile hosts and the absence of a fixed network infrastructure make MANET suitable for applications such as battlefield, rescue operations and conferences.

Since power restrictions and network partitioning in MANETs cause data to become unavailable, multiple copies of the same data are often created. This process of creating multiple copies of the same data item is called data replication. Data replication is more complex in MANETs than in conventional distributed database systems. This is because replication for MANETs should address the issues of power restrictions, mobility of clients and servers, network disconnection and partitioning, as well as real-time requirements since many MANET applications are time-critical [12]. In addition to these concerns, in practice, a MANET data replication scheme also needs to deal with the following limitations:

* Contact author.

1. The data replication scheme should be able to work in the absence of a global node positioning/location management system. This is because the use of such devices/systems increases the power consumption of the mobile hosts in the network. In addition, it is unrealistic to impose the use of these systems on applications which may otherwise have no use for them.
2. The absence of motion parameters of mobile hosts like speed and direction should not cause the data replication scheme to fail. It is not realistic to assume that information about the movement of users is always known ahead of time, especially in MANET applications like rescue operations.

MANET replication is a widely researched area. However, to the best of our knowledge, all existing MANET data replication techniques assume the availability of a GPS and/or that of the motion parameters of mobile hosts. This paper attempts to propose a MANET data replication technique, called REALM (Replication of data for A Logical group based MANET database) that does not make such assumptions.

Mobile hosts in typical MANET applications belong in groups based on the functions they perform in the applications. Since all mobile hosts in a group perform the same set of operations, they access the same set of data items. Members of every group are spread across the network, and perform their functions in their locations. Due to this many tend to move within a limited distance of one or two hops. They collaborate with members of other groups in their locations to achieve the goals of the MANET application. REALM capitalizes on this common MANET application semantic by grouping mobile hosts into logical groups. Each logical group is comprised of mobile hosts that will need to access the same data items.

REALM aims at increasing the percentage of successful transactions while reducing the energy consumption of mobile hosts, as well as balancing the energy consumption of servers. By considering such groups of mobile hosts, REALM will be able to replicate data items closer to clients that will need to access the data items more often. This will reduce the execution time of transactions, and hence, will result in an increased percentage of successful transactions.

The rest of this paper is organized as follows: Section 2 presents some representative MANET applications and identifies their semantics; Section 3 reviews some existing works in MANET data replication; Section 4 describes the proposed replication technique (REALM); Section 5 introduces the prototype used to evaluate the performance of REALM; Section 6 presents the results of the experiments performed; and finally, Section 6 provides conclusions and future research directions.

2 MANET Applications

Since MANETs do not require a fixed infrastructure, they find use in a range of applications such as disaster rescue, personal area network [14] and one laptop per child project [9]. In this section, we briefly describe some of these applications and identify the semantics that should be incorporated into the design of a data replication technique.

1. *Disaster Rescue Application*: A disaster rescue application involves functions like identifying injured people, getting medical aid to the injured, and identifying the dead. Since each of these functions is performed by a group of personnel, they will access the same set of data items. Members of different groups collaborate with each other to achieve the overall goals of the application. For example, a member of the group that rescues people from the disaster will need to collaborate with members of the first aid team. Members of the same group are spread out, and they provide their services in the rescue site at their allotted location. They may also move around to provide their services at other locations.
2. *One Laptop per Child Project Application*: This non-profit project aims to sell cheap and durable laptops to governments such as Africa. The laptops are a special prototype designed by the project and serve as learning tools for their recipients. One particular aspect of the design is that the computers work in an ad-hoc network mode and can keep an Internet connection concurrently. The computers are created with a long battery life, but are limited by their low processing capability and storage space. In a school environment, groups of students require data for their different classes. Students will also move from class to class where different data may be needed.
3. *Personal Area Network (PAN)*: PAN is a very popular application of MANET. It allows users to set up their own network to be able to access their information on different systems like laptops, desktops, pocket PCs and cell phones. In a household or small office environment, different groups of people can use the PAN to access information pertaining to their interest. For example, in an office environment, the accounts team may need to access the revenue and expenditure of the business, while the investment team may need access to sales statistics and fund details. Since the network is set up for a small office, the users move around the boundaries of the office.

The example MANET applications described above have the following common characteristics:

1. Users belonging to the same group need to perform the same set of tasks. Hence they access the same data items. For the purposes of data replication, such users who need access to the same data item can be considered to be in the same logical group. Knowledge of the existence of logical groups will allow every server in the network to identify data items that will be highly accessed by clients that may submit their transactions to it.
2. Users move randomly; however, their movement is mostly within a finite distance. On average, the range of an 802.11b wireless card is about 100 meters. Hence, from the above description, it is realistic to say that the movement of the users is within one or two hop communication distance of an 802.11b wireless card.
3. Members of one group tend to communicate and collaborate with members of other groups in order to achieve the objectives of the application.

The above identified common characteristics of MANET applications can be used to improve data replication, as we will show in Section 4 where our proposed algorithm, REALM, is presented.

3 Literature Review

While MANET replication is a widely researched area, there exists a limited number of data replication techniques designed for MANET database systems. [4] provides three models for replicating data in a distributed database. Static Access Frequency allocates data on each server until storage is exhausted. Dynamic Access Frequency and Neighborhood, introduces additional functionality to remove duplication between neighboring hosts. The Dynamic Connectivity Based Grouping method improves replication by removing redundancy between groups of mobile hosts. [5] improves on these techniques by considering the occurrence of data update.

[6] uses the link strength between hosts when deciding which data item replicas to eliminate. However, [6] relies on the availability of the movement information of mobile hosts to calculate the link strength between them. These works also assume that the access frequencies of data items are known ahead of time and do not change. [7] improvises on these techniques by considering correlation between data items in the calculation of data access frequencies.

[12] weighs data access frequencies based on the remaining energy level of the host and the link strength of connections to other hosts. It considers server workload and energy consumption when accessing replicas. It relies on the availability of GPS as well as the movement information of the mobile hosts to calculate link strength as well as to predict network partitioning.

As discussed in Section 1, the availability of a GPS or movement information of mobile hosts is not always realistic in MANET applications. In addition, data access frequency calculated by a server may change as move near and away from the server. Hence, it is not realistic to assume that access frequencies are known ahead of time and are constant. Apart from these limitations, no existing technique has explored the effect of application semantics on data access in MANET applications. From our study of MANET applications, we believe that knowledge of application semantics can help servers store data items closer to clients that access them more frequently. The next section describes REALM (REplication of data for A Logical group based MANET database). REALM tries to incorporate application semantics in data replication to overcome the shortcomings of existing MANET replication works.

4 Proposed Replication Technique (REALM)

The main goal of REALM is to increase the percentage of successful transactions. It also attempts to reduce the power consumed by the mobile hosts, and balance the power consumed by servers in the network.

Unlike existing MANET works, REALM takes advantage of the characteristics of typical MANET applications (presented in Section 2). REALM groups mobile hosts based on the data items they will need to access. Mobile hosts that access the same set of data items belong in the same logical group. Group membership of mobile hosts helps in identifying the data items that any mobile host in the network will need to access, as well as to identify the most frequently accessed data item on every server.

REALM does not require the use of GPS or the knowledge of movement information of the mobile hosts in the network.

REALM consists of four algorithmic components, namely *MANET Partition Prediction Algorithm*, *Replica Allocation Algorithm*, *Replica Access Algorithm* and *Replica Synchronization Algorithm*. Before describing these components, Sections 4.1 and 4.2 describe the transaction management architecture and data and transaction types considered by REALM, respectively.

4.1 Transaction Management Architecture

Clients in a MANET application submit their transaction to any server in the network. The coordinator is the server that receives the transaction from the client. The coordinator selects servers to access the data items required to execute the transaction. After the execution of the transaction, the coordinator collects the results and forwards them to the client. As described in Section 2, group members in MANET applications are spread out in the area of the application. Due to this, selecting participant servers based on group membership will incur increased transaction execution times. Hence, REALM relies on a decentralized MANET architecture in which data is accessed from servers regardless of their group membership.

4.2 Data and Transaction Types

REALM considers both read only and read write data. Apart from real-time transaction types, firm and soft, REALM also considers the following transaction types:

1. *Accurate Value Transactions*: These transactions need the most recent value of the data to succeed. A transaction initiated to increase the death toll by 1 is an example of an Accurate Value transaction.
2. *Approximate Value Transactions*: These transactions can tolerate outdated values of data items. An example of this type of transaction is one that tries to access weather information.

4.3 MANET Partition Prediction Algorithm

The MANET partition prediction algorithm is based on the Optimized Link State Routing Protocol (OLSR) [11]. It is used by all servers in the network to predict the occurrence of network disconnection/partitioning.

The MANET partition prediction algorithm capitalizes on the hello packets that are used by OLSR to implement neighbor discovery. OLSR transmits one hello packet every x seconds. In addition, OLSR calculates the strength of every link once every y seconds (x and y are configurable). The value of y should be chosen after considering the tradeoff between keeping the servers updated with changes in link quality and the energy consumption overhead that it may pose. This ensures that the routing neighbor tables are kept updated with minimal overhead of messages. Using the number of hello packets that were expected and the number of hello packets that were actually received, mobile hosts S_1 and S_2 at either side of a link calculate the link quality (LQ) using the following formula [11]:

$$LQ = \frac{\text{No. of hello packets received in the last } y \text{ seconds}}{\text{No. of hello packets expected in the last } y \text{ seconds}} \quad (1)$$

Then, S_1 and S_2 exchange their calculated values of LQ through a ‘Topology Control’ message. At either host, the value of LQ received from the other end is called Neighbor Link Quality (NLQ). Using LQ and NLQ, S_1 and S_2 calculate the bidirectional link quality as follows [11]:

$$\text{Bidirectional link quality of the link connecting } S_1 \text{ and } S_2 = LQ * NLQ \quad (2)$$

When the bidirectional link quality decreases beyond a preset threshold accompanied by an increase in the transmission time, S_1 and S_2 predict their disconnection. Then they identify other the mobile hosts that will become unreachable due to their disconnection. When the total number of servers that will be disconnected from S_1 (S_2) increases beyond a preset threshold, S_1 (S_2) triggers replication.

4.4 Replica Allocation Algorithm

The replica allocation algorithm is executed by every server in the network to determine the data items that must be stored on it. It assumes that the group definitions and group memberships of every mobile host are known by all servers. In addition, the location of the original copy of every data item is stored in the replica table of every server. Every server S triggers the execution of this algorithm when any one or more of the following conditions is true:

1. The percentage of successfully executed transactions at S falls below a preset percentage. This ensures improved data availability in the network.
2. The occurrence of network partitioning is predicted by S .
3. The percentage of transactions executed on S that access data on other servers is greater than those that access data on S . This is done to minimize the number of remote transactions since such transactions consume higher server energy.
4. The energy level of S falls below the average energy level of the servers that are one or two hops away from S . This is done for the server S to reduce the energy consumption of S .

To illustrate the working of the replica allocation algorithm, consider the example MANET topology in Fig. 1 where C_i and S_j denote a client and a server, respectively. The ovals around C_i and S_j denote the communication range of C_i and S_j . This allows the figure to represent the number of communication hops between C_i and S_j . Mobile hosts with overlapping communication ranges can communicate directly. Mobile hosts also act as routers and relay communication between unconnected mobile hosts. For example, S_1 and C_1 communicate directly, while C_1 and S_2 communicate through S_1 . Let S_1 , S_2 and S_3 store the original copies of data items D_1 , D_2 and D_3 . Let the original copies of D_4 , D_5 and D_6 be disconnected from the network. This might be due to loss of energy on these servers or their movement into a different partition. Tables 1 and 2 represent the group membership of the mobile hosts in the network and the access log maintained at server S_3 respectively. At the time when replication is triggered by S_3 , let C_6 be disconnected from S_3 .

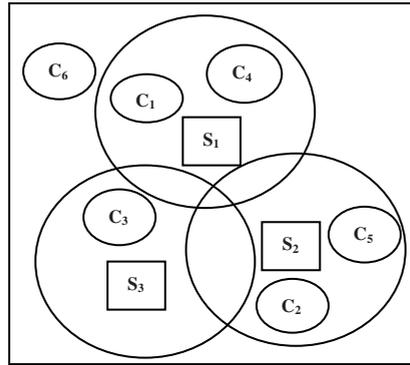


Fig. 1. Example MANET Topology

Table 1. Group Membership of Mobile Hosts

Group Name	Group Members	Data Items of Interest to Group Members
G ₁	C ₁ , C ₄	D ₁ , D ₃
G ₂	C ₃ , C ₅	D ₂ , D ₅
G ₃	C ₂ , C ₆ , S ₂	D ₂ , D ₄ , D ₆

Table 2. Access Log at Server S₃

Data Items	Mobile Hosts	Number of Accesses of the Data Item
D ₄	C ₃	10
D ₅	C ₃	12
	S ₂	20
D ₂	C ₃	25
	C ₆	8

When replication is triggered on S₃, the following steps are performed:

1. Calculation of access frequencies: The access frequency of a data item D on server S (AF^D_S) is calculated as the number of transactions which were submitted to S by clients that are currently within two hops from it.

At S₃, the access frequencies of the various data items are calculated as follows:

$$AF^{D_4}_{S_3} = \text{Number of accesses made by } C_3 = 10$$

$$AF^{D_5}_{S_3} = \text{Number of accesses made by } C_3 + \text{Number of accesses made by } S_2 = 12 + 20 = 32$$

$$AF^{D_2}_{S_3} = \text{Number of accesses made by } C_3 = 25$$

The number of accesses for D_6 made by C_6 is not considered in the calculation of the access frequency of D_2 since C_6 is not a one or two hop neighbor of S_3 .

2. Arrange the data items in the descending order of their calculated access frequency in a list L . At S₃, L consists of data items D_5 , D_2 and D_4 , after arranging the data items in the descending order of their access frequencies.

3. For each mobile host within two hops from S , determine the data items of interest to the mobile host based on its group membership. If any of these data items are not already present in L , add it to L . At S_3 , D_6 is added to L since it is one of the data items of interest to group G_3 , to which S_2 belongs. Hence, L at S_3 consists of D_5 , D_2 and D_4 and D_6 .
4. For each data item D in L , determine whether the original copy server of D or a replica of D is connected to S . If so, eliminate D from L . In the example, since S_2 stores the original copy of D_2 , S_3 eliminates D_2 from L . Hence, L at S_3 consists of D_5 , D_4 and D_6 .
5. For every remaining data item D in L , if the mobile host that has accessed D maximum number of times is a server, request that server to store a replica of D . If the request is accepted, eliminate D from L . At S_3 , S_2 is the mobile host which accesses D_5 . Hence S_3 requests S_2 to store a copy of D_5 . Once S_2 accepts this request, L at S_3 then consists of D_4 and D_6 .
6. Store the remaining data items in L on S until S runs out of free space. S_3 stores D_4 and D_6 .

4.5 Replica Access Algorithm

The aim of the replica access algorithm is to help increase the percentage of successful transactions while also conserving the power consumption of mobile hosts. The data access is based on the data and transaction type.

Since firm transactions have only one deadline, they are executed at the nearest server that stores the required data. Soft transactions, on the other hand, have a higher probability of execution compared to firm transactions. Hence, they are executed at the nearest server with highest energy that stores the required data.

Accurate value transactions are executed at the server that has the most recent copy of the data items. This is determined by the coordinator of the transaction by requesting the most recent timestamp of the required data item from servers that store copies of it. Approximate value transactions are executed at the most updated server within two communication hops of the coordinator.

Read only data items can be accessed from any server on which they are stored. Hence, they are executed based on the transaction type. Read write data items are accessed from any server they are stored in, if the energy level of the server is greater than or equal to the average energy level of the servers in the network. This is possible because the replica synchronization algorithm of REALM ensures that servers with energy level at least equal to the average server energy level in the network are updated at all times.

4.6 Replica Synchronization Algorithm

The replica synchronization algorithm ensures that replicas of every data item in the network are in sync. It does this by propagating updates to servers that store the updated data item. However, since such an approach may lead to high server power consumption, the update operation and the most recent update timestamp of the data item is propagated only to servers with energy level at least equal to the average power level of the servers in the network.

Once a server S receives a propagated update for a data item D it stores, it compares the received timestamp with the last updated timestamp of its copy of D . If the timestamps match, the two copies of D are in sync with each other. So S executes the received update operation on D . If the timestamps differ, S requests a copy of the most recent value of D .

5 Prototype Model

REALM has been implemented on an existing MANET prototype ([3]). This prototype consists of servers (laptops) on Fedora Core Linux operating system running the MySQL database [10], and clients (PDAs) running the Familiar Linux operating system. Both servers and clients in the network run the Optimized Link State Routing Protocol (OLSR) with its link quality extension enabled. Clients in the network generate real-time firm and soft transactions.

Apart from REALM, two other replication models, namely, the No Replication model and the Hara model ([5], [6]) have been implemented on the prototype for purposes of performance comparison. The No Replication model allows us to compare the performance of REALM with that of a system that does not store multiple copies of data items. REALM and the Hara model are similar in that both assume a decentralized MANET architecture and consider update transactions. REALM replicates data items differently from the Hara model by allowing application semantics to influence data replication. In addition, REALM also tries to predict the occurrence of network partitioning and replicates data items as necessary.

We created a sample database based on the requirements of a fire rescue scenario. The experimental prototype consisted of five servers and eight clients. Four logical groups of mobile hosts were defined. At the start of every experiment, only the original copy of every data item was created. Each client initiated a total of 1000 transactions for every run of the experiment. A comparison between the performances of REALM and the Hara model, as presented in Section 5, allows us to gauge the effectiveness and overhead incurred by REALM.

6 Experimental Results

Experiments were performed using the prototype described in Section 5, varying the dynamic parameters namely, Percentage of Firm transactions, Percentage of Accurate transactions, Access Frequency, Frequency of Network Partitioning, Transaction Inter Arrival Time and Read/Write Ratio. Due to space constraints, we present only some of the graphs obtained from the experiments performed.

6.1 Effects of Firm Transactions

Fig. 2 shows the effect of the percentage of firm transactions on the percentage of successful transactions and the server energy consumption. As the percentage of firm transactions increases, the percentage of successful transactions decreases. REALM executes a higher percentage of transactions successfully as it continually monitors

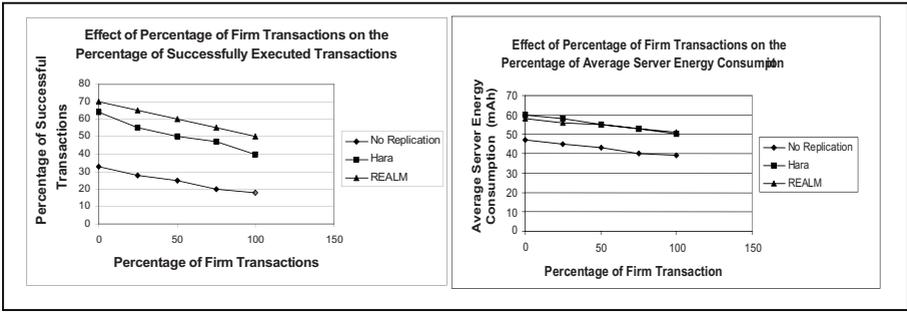


Fig. 2. Effect of Percentage of Firm Transactions

the percentage of successful transactions and replicates data items to improve this metric. As a consequence of this, however, REALM incurs high server energy consumption. The Hara model also incurs comparable server energy consumption as it broadcasts data items in order to synchronize its replicas.

6.2 Effects of Accurate Value Transactions

Fig. 3 shows the effect of percentage of accurate value transactions on the percentage of transactions executed and the average difference in energy consumption of two servers.

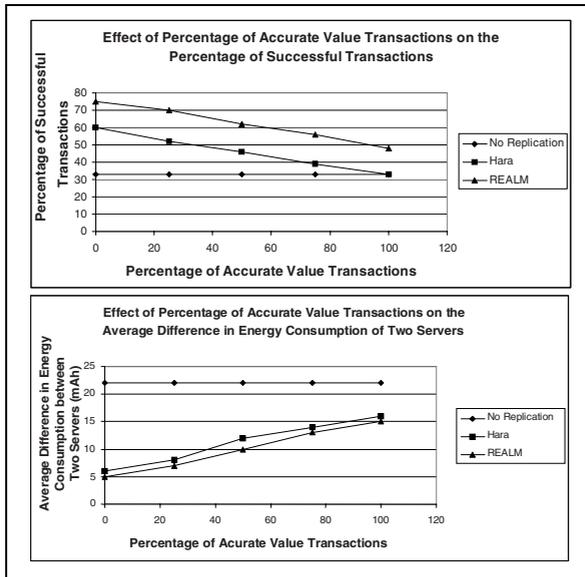


Fig. 3. Effect of Percentage of Accurate Value Transactions

The percentage of transactions successfully executed by the No Replication model is independent of the percentage of accurate value transactions since it does not maintain copies of data items. REALM propagates updates as soon as they are processed; increasing the number of updates replicas. Hence, it executes the highest percentage of successful transactions among the models being studied. With increase in the number of accurate value transactions, the workload of those servers with the most updated copy of data items increases. This causes the increase in imbalance of server energy consumption of both the Hara model and REALM. REALM incurs the least imbalance among the three models. This is because it replicates and accesses data items considering the energy levels of servers.

6.3 Effect of Data Access Frequency

The effect of data access frequencies on the percentage of successful is shown in Fig. 4. Data access frequency is represented using the zipf parameter θ . As θ increases, the number of data items being accessed frequently decreases. Due to this, Hara model as well as REALM increases the number of replicas in the network. Hence, the percentage of successful transactions yielded by both models increases with increase of θ . Unlike the Hara model, REALM considers the requirements of the clients in addition to the access frequencies. Due to this, it is able to cater to the needs of the clients better than the Hara model. This allows REALM to yield higher percentage of successful transactions even as the value of θ increases.

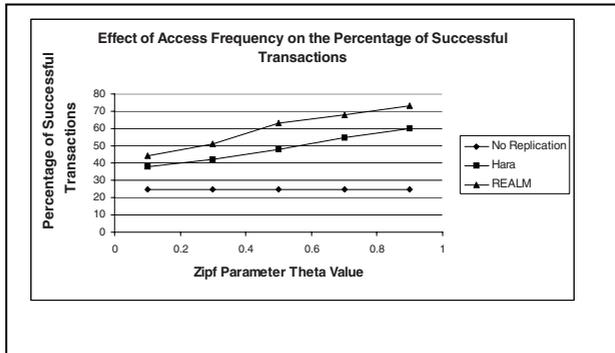


Fig. 4. Effect of Data Access Frequency

6.4 Effects of Network Partitioning Frequency

Fig. 5 shows the effect of frequency of network partitioning on the percentage of successful transactions and the average server energy consumption. With increase in the frequency of network partitioning, servers move into different partitions, increasing the unavailability of data. REALM yields the highest percentage of successful transactions among the models studied. This is because REALM tries to predict network partitioning and triggers replication ahead of its occurrence. However, this also

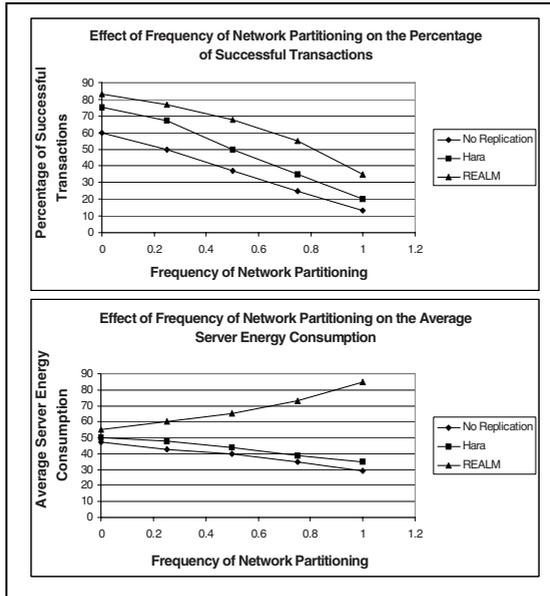


Fig. 5. Effect of Frequency of Network Partitioning

causes its server energy consumption to rise. This is because, with every partition prediction, REALM replicates data items. As the frequency of partitioning increases, the number of times that replication is triggered increases.

6.5 Effects of Transaction Inter-arrival Time

Fig. 6 shows the effect of transaction inter arrival time (IAT) on the percentage of successful transactions. The percentage of successful transactions increases with increase in IAT. This is due to decrease in the time rate of server workload.

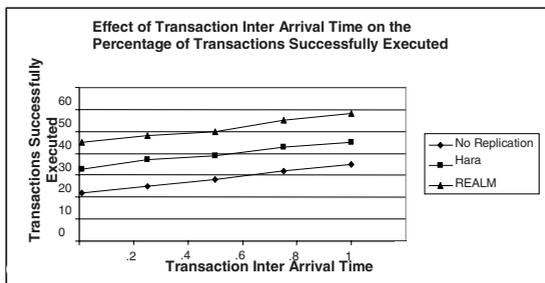


Fig. 6. Effect of Transaction Inter Arrival Time

The No Replication model queues up transactions that access the same data item on a single server because each data item is only available in a single location. Unlike the Hara model, REALM reduces the total execution time of every transaction considering communication distance as a factor for choosing participants. Hence, it yields the highest percentage of successful transactions among the models studied.

7 Conclusions and Future Work

This paper presented a data replication technique called REALM for logically group based MANET real-time databases. By considering groups of mobile hosts which access the same set of data items, REALM is able to take advantage of the needs of the clients in the network at the time of replication. REALM yields the best performance in terms of percentage of successful transactions and balance in server energy consumption when compared to the existing techniques.

REALM can be further strengthened by developing a rigorous, power-efficient partition prediction scheme. Clients can maintain replica tables to be able to make a more informed choice of the transaction coordinator. The conditions for triggering replication can be checked after a preset number of transactions have been executed. Guidelines to choose this value can be established. In addition, REALM can also be extended to tolerate overlapping and variable group definitions. This will involve a scheme to update the group membership of mobile hosts.

References

1. Bellavista, P., Corradi, A., Magistretti, E.: REDMAN: A Decentralized Middleware Solution for Cooperative Replication in Dense MANETs. In: International Conference on Pervasive Computing and Communications Workshops, pp. 158–162 (2005)
2. Chen, K., Nahrstedt, K.: An integrated data lookup and replication scheme in mobile ad hoc networks. In: SPIE International Symposium on the Convergence of Information Technologies and Communications, pp. 1–8 (2001)
3. Gruenwald, L., Bernedo, P., Padbmanabhan, P.: PETRANET: A Power Transaction Management Technique for Real Time Mobile Ad-Hoc Network Databases. In: IEEE International Conference on Data Engineering (ICDE), p. 172 (2006)
4. Hara, T.: Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility. In: IEEE INFOCOM, pp. 1568–1576 (2001)
5. Hara, T.: Replica Allocation Methods in Ad Hoc Networks with Data Update. ACM-Kluwer Journal on Mobile Networks and Applications, 343–354 (2003)
6. Hara, T., Loh, Y.H., Nishio, S.: Data Replication Methods Based on the Stability of Radio Links in Ad Hoc Networks. In: The 14th International Workshop on Database and Expert Systems Applications, pp. 969–973 (2003)
7. Hara, T., Murakami, N., Nishio, S.: Replica Allocation for Correlated Data Items in Ad Hoc Sensor Networks. ACM SIGMOD RECORD, 38–43 (2004)
8. Hauspie, M., Simplot, D., Carle, J.: Replication decision algorithm based on link evaluation services in MANET. CNRS UPRESA 8022 – LIFL Univ. Lille (2002)
9. One Laptop Per Child (last accessed, April 2007), <http://www.laptop.org>
10. MySQL (last accessed, February 2007), <http://www.mysql.com>

11. OLSR (last accessed, February 2007), <http://olsr.org>
12. Padmanabhan, P., Gruenwald, L.: DREAM: Data Replication in Ad Hoc Mobile Network Databases. In: IEEE International Conference on Data Engineering (April 2006)
13. Ratner, D., Reiher, P., Popek, G.J.: Roam: A Scalable Replication System for Mobility. *Mobile Networks and Applications* 9(5), 537–544 (2004)
14. Sun, J.: Mobile Ad Hoc Networking: An Essential Technology for Pervasive Computing. In: Proc. of ICII, Beijing, China, pp. 316–321 (2001)