

AUTOMATED RE-ALLOCATOR OF REPLICAS OVER MOBILE AD HOC NETWORKS

Inas Abuqaddom¹, Dr. Azzam Sleit¹, Dr. Wesam Almobaideen¹

¹king Abdullah II School for Information Technology, University of Jordan, Amman,
Jordan

ABSTRACT

In a Mobile Ad hoc Network (MANET), due to mobility, limited battery power and poor features of nodes, network partitioning and nodes disconnecting occur frequently. To improve data availability, database systems create multiple copies of each data object and allocate them on different nodes. This paper proposes Automated Re-allocator of Replicas Over MANET (ARROM), that addresses these issues. ARROM reduces the average response time of requests between clients and database servers by reallocating replicas frequently. In addition, ARROM increases the average throughput in the network. Our performance study indicates that ARROM improves average response time and average network throughput in MANET as compared to resent existing scheme.

1 INTRODUCTION

Recently, wireless communications have become an important technology in reality, as well as in science. MANET is a wireless network of laptops and PDAs without static infrastructure, which provides high mobility for MANET's nodes (mobile hosts). Hence the network topology is changing frequently [3]. Moreover, the functionalities of mobile hosts (MHs) are both end-systems and routers, which impose workload on MHs [19]. This workload affects MHs because of their limited battery power, limited bandwidth and poor resources [3]. Thus, many mechanisms are used to face these challenges. One of them is a data replication, which improves data accessibility (number of successful requests/number of successful and failed requests), but with increased cost of storage space and communication overhead. In addition, the wired network protocols must be adapted to be used in MANET due to the critical characteristics of MANET [19, 27 and 28].

Distributed Database Management Systems (DDBMSs) consist of data distribution and application distribution among several nodes on network. Types of application distribution are distribution of either DBMS itself or programs that run on DBMS. On the other hand, data distribution has two basic alternatives; replication and fragmentation. Replicas are replicated data of DBMS on different nodes mainly to reduce response time for read operations. Data is replicated either fully, i.e. the database is stored at each node, or partially, i.e. partition of the database is stored at some nodes but not all. The problem appears with write operations, because updates must be propagated to all replicas to keep them consistent. Fragment is equivalent to replica, but they are different in the number of data copies. Fragment has one data copy for each, whereas replica has two or more data copies for each. Moreover, fragment is always for a partition of the database which is operated as a separate database [2, 16].

Replication issues are classified into three categories. First one is replica relocation, which implies when, where and how replicas are allocated. Optimal replica allocation cannot be

determined in MANET due to nodes mobility. Second category is consistency management to keep replicas as consistent as possible as an update is performed on a replica even if there is a partitioning in the network. Third category is location management, i.e. requesting host (RH) must know the location of requested data object or its replicas. Otherwise, requesting host will broadcast the request over the entire network, which causes a heavy traffic [7]. Every replication system provides method to handle each of these issues.

Several researchers [5, 6, 7, 10, 11, 30 and 31] show that all replication methods in MANET perform most of their works periodically, due to nodes' mobility; this period is called relocation period. Some common tasks are performed at every relocation time in any replication system as the following:

1. Determining groups of nodes according to the mechanism, which the replication system uses.
2. Specifying the number of replicas for each data object depending on Read /Write Ratio (RWR), that is explained in section 2.
3. Allocating all data objects and their replicas.
4. Discarding replica allocation information and logs of past data accesses to rebuild them during and for the next period. This information is used in location management methods, such as Access Log (AL) and Group Management (GM) methods.

Reducing the communication cost of accessing replicas improves the usage of MHs' poor resources. The best way to reduce the communication cost of accessing replicas is by reallocating replicas frequently due to nodes mobility and request pattern. In other words, we have to use shorter relocation period but this means loaded network due to performing the above tasks many times without needing to redo them very often. On the other hand, longer relocation period means that replicas are reallocated few times regardless of the MHs' request pattern. To solve this problem, we propose to use appropriate long relocation period and to reallocate existing replicas separately during relocation period. Thus, we get higher performance due to reallocating replicas frequently with longer relocation period.

This paper introduces ARROM algorithm. It reallocates existed replicas in order to reduce the communication cost of both accessing and reallocating them, which improves the usage of MHs' poor resources. ARROM reallocates replicas frequently during relocation period to get higher performance.

Rest of this paper is organized as follows: section 2 describes related works. The details of our proposed approach (ARROM) are explained in section 3. Section 4 presents our simulation settings, while our simulation results in section 5. Finally we give conclusions and future works in section 6 and section 7 respectively.

2. RELATED WORK

In [5, 7, 9, 24, 25, 26 and 29], some protocols were introduced, such as SAF, DAFN, DCG, E-SAF+, E-DAFN+ and E-DCG+. In Static Access Frequency (SAF) method, each MH allocates replicas according to descending order of its own access frequencies and the limitation of its own memory space. Each MH does not know the list of data objects that are allocated in any other MH even if they are neighbors. Thus, SAF provides low accessibility when many MHs have similar access frequencies, because each replica serves only the MH that holds this replica. However, SAF provides low overhead and traffic, because of no relocation after allocation process. In addition, SAF continues execution until no memory space is available, which produces a large number of replicas.

Dynamic Access Frequency and Neighborhood (DAFN) works as SAF with a mechanism to eliminate duplicate replicas among neighboring MHs. The MH whose duplicated replica has lower access frequency than other duplicated replicas, this replica will be replaced by a replica of other data object. Thus, DAFN runs each relocation period, and provides higher accessibility. As a result for elimination duplicated replicas among neighboring MHs which are connected just by one or two hops, DAFN generates overload and traffic higher than SAF.

Dynamic Connectivity based Grouping (DCG) is the same as DAFN but with larger group size. DCG uses broadcast messages to group MHs into biconnected components i.e., if any MH disappears or any link is broken, the group will not be divided into separated smaller groups, this provides groups with high stability. Data objects are selected to be replicated according to the summation of access frequencies of all MHs in the group for the same data object. The data object with the highest summation will be replicated and be allocated in a MH whose access frequency of this data object is the highest among all MHs in the group. DCG provides larger group with sharing replicas, which means higher accessibility but with more overhead and traffic than DAFN.

SAF, DAFN and DCG methods are missing write operations, which means any update operation over a replica of a data object makes all its replicas invalid. To solve this problem an extended E-SAF+, E-DAFN+ and E-DCG+ methods are introduced. These methods work the same as SAF, DAFN and DCG respectively, but rather than using access frequency they use Read/Write Ratio (RWR). It is the ratio between the probability of read operations that are performed on data object by a MH at a unit of time, and the probability of write operations that are performed on the data object by the MH which has the original data object. Higher RWR indicates more read operations than write ones. Thus, we can replicate the data object. Lower RWR indicates the opposite, so no more replication on this data object.

To keep consistency among replicas, these extended methods use a lazy update protocol, which is not executed very often on MHs. Update protocols call the system of sending the updated data object(s) for all MHs that hold its replicas to keep them consistent as propagation system. [4] shows that propagation system has three alternatives; client-server, peer-to-peer and hybrid systems. In client-server, propagation is the responsibility of the MH holding the original data object. Thus, propagation is slow, because it makes a bottleneck at that MH. Additionally, any failure on that MH affects propagation. In peer-to-peer systems, propagation is faster, but this system costly because it is designed to make all nodes tightly connected. Hybrid systems have the advantages of client-server and peer-to-peer systems. In hybrid system, propagation is done by a MH that holds either the original data object or one of its replicas, i.e. peer-to-peer system, with other nodes serving as client i.e. client-server system.

In [13] DRAM scheme was introduced. DRAM groups MHs according to their mobility behavior. The mobility behavior of mobile users is usually regular and has some mobility patterns. DRAM uses Reference Point Group Mobility (RPGM) model to determine the mobility behavior, and uses a decentralized clustering algorithm to cluster MHs with similar mobility behavior into one group. Then data objects are replicated in each mobility group at the MH that has the highest access frequency and enough memory space. As a result, DRAM provides higher data accessibility than E-DCG+ method. Moreover, the broadcasting in each mobility group is limited to its MHs only. This reduces network traffic.

However, all previous methods continue replicating data objects until the memory space of all MHs become full [7, 9 and 13] this huge number of replicas makes consistency management more difficult, and consumes more resources.

[14] introduced Collaborative Allocation and Deallocation of Replicas with Efficiency (CADRE) method, which works on groups like E-DCG+. CADRE adds two new properties to prevent thrashing condition and to achieve fairness in replication. To explain thrashing assume M, as a MH, has a replica and rarely uses, but M's neighbors use this replica a lot. If M decides to remove this replica, this decision leads M's neighbors to allocate a new copy of this replica. As a result, there is a wasted overload due to deallocation and reallocation for the same replica in the same group. CADRE assigns sigma variable for each data object to indicate its importance. Thus, data objects are replicated when their sigmas or access frequencies are high, and that is the fairness in replication.

In [17 and 18], Data Replication Technique for Real-Time Mobile Ad-hoc Network Databases (DREAM) was introduced. It is differentiated from other techniques by using weight concept of access frequencies. Weight is affected by the type of data and transaction, which are classified in many specific classifications. In general, it classifies data type into read-write data and transactions into read transactions and write transactions. Furthermore, read transactions are classified into Most Resent Value (MRV) which needs the most resent value across all replicas in all partitions, Outdated transaction (OD) which can be executed successfully even with stall data and Most Resent Value in Partition (MRVP) which needs the most resent value across all replicas in a partitions. Additionally, DREAM classifies write transactions are classified into Insert/Delete transactions, Use Current Value (UCV) and Overwrite Current Value (OCV).

3. PROPOSED APPROACH

Most replica relocation methods relocate replicas in order to improve data availability in MANET. Our proposed approach (ARROM) relocates replicas to reduce communication cost of accessing these replicas, which improves the usage of MHs' poor resources, in addition to improve data availability in MANET. ARROM has the assumption that MHs have global clock synchronization.

We divide the execution of ARROM into two stages, whereas most replica relocation methods have one stage. Each stage has certain aims, and they are re-executed periodically. We call the period of the first stage the relocation period and the period of the second stage the reallocation period, which must be smaller than the relocation one. First stage is the common stage among most replica relocation methods. We built our first stage exactly as E-DCG+ method, which includes the following tasks:

1. Grouping MHs into biconnected components.
2. Specifying replicas count for each data object according to its RWR.
3. Allocating all data objects and their replicas according to the summation of access frequencies on the data object from all MHs in the group.
4. Discarding replica allocation information and the logs of past data accesses to rebuild them during and for the next period.

ARROM uses broadcast messages to group MHs into biconnected components. Each MH broadcasts its identifier and its own access frequencies on each data object. After completing broadcasts, every MH knows its connected MHs by forwarding broadcast messages, such that each receiver MH adds its identifier to the forwarded message. Then each set of connected MHs executes an algorithm to find biconnected components on the MH having the lowest identifier. This algorithm finds the largest set of connected MHs that can forward a message among them before it comes back to the initial sender. We deal with each biconnected component as a group, i.e. if any MH disappears or any link is broken, the group will not be divided into separated smaller groups due to looping path(s). This provides groups with high stability. If a MH belongs

to more than one biconnected components, it belongs to only one group in which the corresponding biconnected component is first found while executing the algorithm. After creating groups, the summation of access frequencies of all MHs in the group to each data object is calculated. These calculations could be done by any MH in the group, but for standardization, we assumed these calculations are done by the MH having the lowest identifier in the group. Replicas are allocated in the group according to the descending order of the summation of access frequencies on data objects. Each replica is allocated a MH whose access frequency to the data object is the highest among MHs that have free memory space to create it.

Second stage is just for reallocating existing replicas, which are created and allocated in the first stage. We proposed to reallocate existing replicas in order to reduce the communication cost of accessing them in each group. ARROM executes this stage on the MH which has a replica of data object and which has the original data object. We call the former MH as Replica-Server (RSer) and the later MH as Original-Server (OSer). We introduced direction concept, which is determined by grouping MHs according to the last node in the path between a requesting MH and the server (either RSer or OSer). ARROM finds the ratio between the total access costs of read and write operations on a data object from one direction, and the total access costs of read and write operations on that data object from all directions. See equation (1).

$$\text{Ratio}_D = \frac{\text{Access costs to a data object from one direction}}{\text{Access costs to a data object from all directions}} \cdot (1)$$

Ratio is calculated for each direction. The new location for a data object is the direction that has the highest ratio. For further control we use threshold to keep the ratio high enough. Then ARROM calculates the ratio for each Requesting Host (RH) in the selected direction. In other words, ARROM finds the ratio between the total access costs of read and write operations on a data object from one RH in the selected direction, and the total access costs of read and write operations on that data object from all RHs in the selected direction. See equation (2).

$$\text{Ratio}_{MH} = \frac{\text{Access costs to a data object from one RH}}{\text{Access costs to a data object from all RH in the selected direction}} \cdot (2)$$

The data object is allocated in the RH that has enough memory space and has the highest ratio among all RHs in the selected direction.

Each ARROM server (either OSer or RSer) has a Direction Cost Table (DCT) for each data object. DCT has a record for each direction. This record includes the following:

1. Direction identifier, which equals to the identifier of the last MH in the path between RH and the server.
2. Summation of access costs of the data object from all RHs in the direction.
3. Summation of access frequencies of the data object from all RHs in the direction.
4. List of all RHs in the direction. This list includes RH identifier, RH's own summation of access costs on the data object, RH's own access frequencies on the data object and RH's free memory space.

We use the summation of access frequencies, because if two directions or RHs have the same ratio, ARROM compares their access frequencies and decides reallocation according to their access frequencies instead of ratio.

ARROM is a hybrid system, as such, users can perform read and write operations over either the original data object or its replicas and keep them consistent. However, ARROM uses DC/GG

method for consistency management and GM method for location management. Additionally, each group limits broadcasting to its MHs only.

In reallocation condition, we use cost parameter, which is the hops count between RH and server. Our proposed cost model consists of two cases:

1. Case A for read operations:

$$\begin{cases} 0 & \text{if } RH \in SL, & (3) \\ c_d + c_c & \text{if } RH \notin SL, & (4) \end{cases}$$

2. Case B for write operations:

$$\begin{cases} \left(\sum_{s \in SL} c_d \right) & \text{if } RH \in SL, & (5) \\ \left(\sum_{s \in SL} c_d \right) + (c_d + c_c) & \text{if } RH \notin SL, & (6) \end{cases}$$

We group servers both RSer and OSer in a server list (SL). If Requesting Host (RH) is a member of SL, then the cost of read operation is zero as shown in equation (3), whereas cost of write operation equals to the propagation cost, which means to send the updated data object for each member of SL to keep replicas on all servers consistent as show in equation (5). There is an additional cost for each RH that is not a member of SL. It is the cost of getting the data object from the nearest server. This additional cost is divided into two steps. The first one is the cost of sending a control message (c_c) from RH to the nearest server. The second step is the cost of sending the data object (c_d) from the nearest server to the RH. This additional cost appears in equations (4) and (6).

4 SIMULATION SETTINGS

The experiments were evaluated using HP PC. Table 1 presents the system specifications. We used Global Mobile Information System Simulator (GloMoSim) to implement and evaluate ARROM. GloMoSim is a library-based sequential and parallel simulator for wireless networks. It is designed as a set of library modules; each of them simulates a specific wireless communication protocol in the protocol stack. This simulator was built based on the Parallel Simulation Environment for Complex Systems (PARSEC) simulation language [21]. Additionally, GloMoSim has a layered approach similar to the OSI model. Furthermore, GloMoSim provides different protocols for the Physical layer (PHY), Data link layer (MAC), Network layer (Routing), Transport layer and Application layer [23].

Table 1: System specifications

Item	Value
Processor	Intel Pentium 4 CPU 3.20GHz
System Model	HP Compaq
Memory (RAM)	512MB
OS Name	Microsoft Windows XP Professional Version 2002 service Pack 3

The simulation experiments contained 37 clients, one OSer and two RSers according to the maximum size of traffic file [19], which is selected according to the GloMoSim possibilities. These 40 MHs were distributed randomly for generalization using uniform distribution model over a terrain of 2000 × 2000 meters. This terrain is sufficient to disseminate 40 MHs without changing the default radio power, which generates a coverage area with 391 meters as radius. We used random waypoint model for mobility, because GloMoSim provides only this model for random mobility. In waypoint model a node randomly selects a destination and moves along the direction of that destination in a speed uniformly chosen between the minimum speed that we selected and the maximum speed. These speeds specified as MOBILITY-WP-MIN-SPEED and MOBILITY-WP-MAX-SPEED parameters respectively. When the node reaches its destination, it stays there for pause time, which is specified in MOBILITY-WP-PAUSE parameter [1, 15, 20 and 21]. We selected the speed between 0 meter/second and 10 meters/second with 30 seconds pause period according to our experiments. We chose Ad hoc On-demand Distance Vector (AODV) protocol for routing, because it is one of the most famous protocols for flat ad hoc networks [22]. Moreover, to get more accurate results, each experiment was repeated 30 times with thirty different SEEDs (1 to 30). The SEED represents a random number used to initialize various random numbers in the simulation such as the delay time before broadcasting packets [15]. The simulation parameters are summarized in Table 2. The table presents the general parameters used in all experiments. Most of the general parameters are chosen as their default values in IEEE standards and GloMoSim possibilities, except that the parameters whose effects on ARROM are studied.

Table 2: Simulation Parameters

Parameter	Value
Terrain	2000 x 2000 m
Node placement	UNIFORM
Node mobility	RANDOM-WAYPOINT
MOBILITY-WP-PAUSE	30 seconds
MOBILITY-WP-MIN-SPEED	0 meter/second
MOBILITY-WP-MAX-SPEED	10 meters/second
PROPAGATION-PATHLOSS	TWO-RAY
RADIO-TYPE	RADIO-ACCNOISE

Parameter	Value
RADIO-BANDWIDTH	2000000
MAC-PROTOCOL	IEEE 802.11
NETWORK-PROTOCOL	IP
Traffic Generator	FTP/Generic
SEEDs	1-30
ROUTING-PROTOCOL	AODV

Each traffic file has two important parameters; which nodes represent RHs, and how many requests are sent from each one. These parameters were chosen randomly using uniform distribution model to generate 40 different traffic files for each experiment to get more accurate results. Each traffic file has 30 requests, because GloMoSim can deal with only 120 requests per experiment. If the 30 requests were write requests, they will become 120 requests during execution due to propagation process. However, these requests were for one data object and they were generated using FTP/Generic protocol. These requests are served remotely or locally when RH is also RSer or OSer. To avoid changing the server, when it has some requests in progress, we studied the maximum response time among all requests that were used in our experiments. Then we made our reallocation period greater than this maximum response time, which was approximately 2 seconds. For more accuracy, we doubled the maximum response time that was extracted to become 5 seconds. Furthermore, we generated one request per second and selected 5 seconds as the minimum period between any two requests. Therefore, our reallocation period was 5 seconds to guarantee that the server has no requests in progress.

We used E-DCG+ approach for comparison, because that ARROM works exactly as E-DCG+ with the addition to reallocate replicas during relocation period. Thus, we can get the exact performance of reallocation protocol. We chose 10 minutes as a relocation period, because it is the average minimum relocation period in previous works. Thus, our reallocation protocol runs 120 times during each relocation period.

Our performance metrics are Average Response Time (ART) of a request and Average Network Throughput (ANT). ARROM aims to reduce the communication cost of requests. The most important parameter in communication cost is the response time, because it indicates the consumption of MHs' resources. Each MH acts as both end system and router. Therefore, during response time, the request is either being forwarded among MHs or waiting in some MHs due to congestion. We selected throughput as another performance metric, because it indicates the number of sent bytes per unit of time. Therefore, improving throughput means increasing the usage of transmission media, which is one of the most important poor resources in MHs. Furthermore, we define the Improvement Ratio (IR) as:

$$IR = \frac{MAX\ value - MIN\ value}{MAX\ value}, \quad (7)$$

We used IR to show the improvement of ARROM comparing with E-DCG+. Equation (7) normalizes the differences between the two approaches to make values between zero and one.

5. RESULTS AND ANALYSIS

We studied the effects of reallocation threshold on ARROM. Figure 1 shows the reallocation threshold effects on average response time. For all thresholds ARROM gives better average

response time than E-DCG+. Additionally, average response time of E-DCG+ is constant for all threshold values, because E-DCG+ does not reallocate replicas during relocation period.

When reallocation threshold is large, ARROM becomes close to E-DCG+. This is because large reallocation thresholds reduce reallocation occurrences as Figure 2 shows. On the other hand, small reallocation thresholds reallocate the data object many times. As a result, we get better average response time when the reallocation thresholds decrease. This result is not true all the time, because there are limitations on decreasing average response time, such as the transition media speed and the processing time of each request. Figure 1 shows that the average response time of ARROM with the first several small thresholds are approximately constant, although their corresponding reallocation frequencies are different. Figure 1 and Figure 2 illustrate that with three reallocation thresholds, which are 5, 10 and 20.

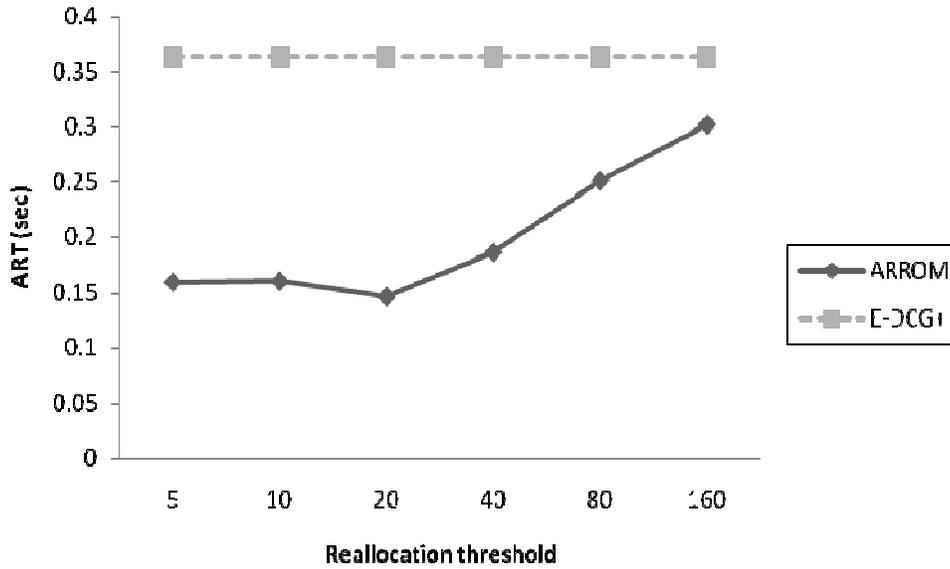


Figure 1: The ART of a request with different reallocation threshold values.

Figure 3 shows the reallocation threshold effects on average network throughput. For all thresholds ARROM outperforms E-DCG+ in the average network throughput. Additionally, E-DCG+ is constant for all thresholds, because E-DCG+ does not reallocate replicas during relocation period. Notice that Figure 1 and Figure 3 have the same behavior, because it depends totally on reallocation occurrences. At the same time, there are limitations on increasing the average network throughput, such as the bandwidth of transition media. Therefore, the average network throughputs of ARROM with the first several small thresholds are approximately constant, although their corresponding average reallocation frequencies are different. Figure 2 and Figure 3 illustrate that with three reallocation thresholds, which are 5, 10 and 20.

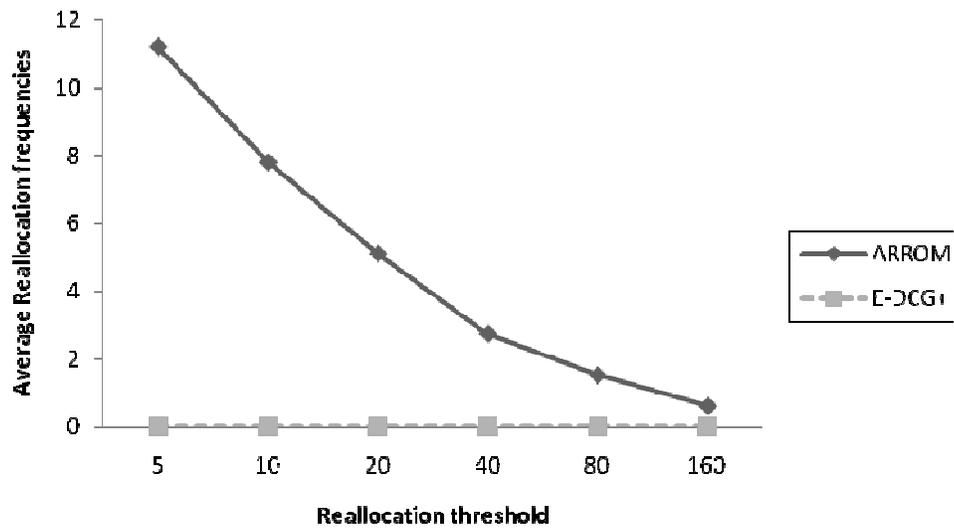


Figure 2: The average reallocation frequencies for different reallocation thresholds.

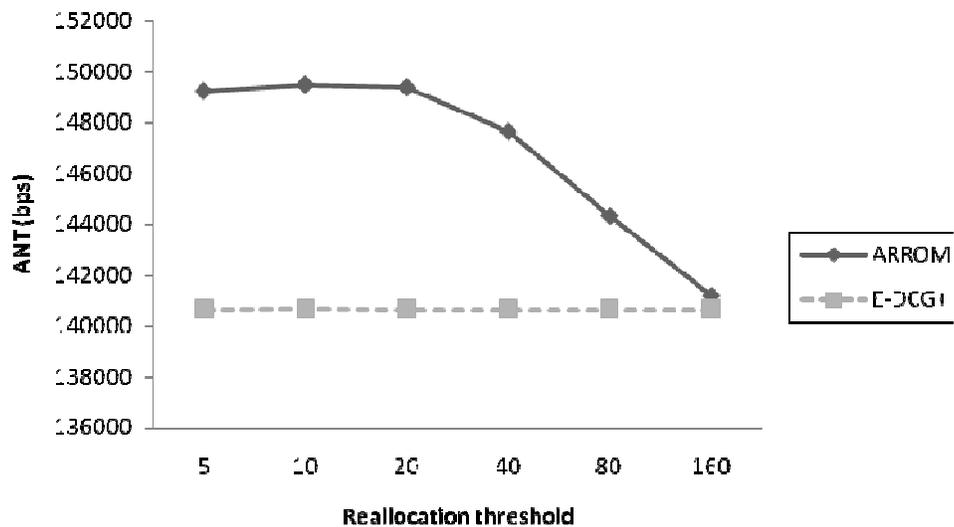


Figure 3: The ANT with different reallocation threshold values.

The best threshold here is 20, because it provides minimum average response time and maximum average network throughput with minimum reallocation frequencies. However, the best threshold reallocates replicas not very often, because each reallocation process includes the following:

1. Moving the data object to the new location.
2. Broadcasting the new location of the data object to all MHs in the group.

To get high performance, we must keep the reallocation occurrences of the data object in moderate frequencies. Figure 2 shows that threshold 20 gives moderate reallocation frequencies between maximum and minimum reallocation frequencies.

We studied the effects of requested data object size on ARROM. Figure 4 shows the average response time of ARROM and E-DCG+ approaches with different requested data object sizes. The difference between the two approaches increases as the data object size increases, because the number of sent bytes increases. In other words, the delay of sending a set of bytes increases as the number of sent bytes increases.

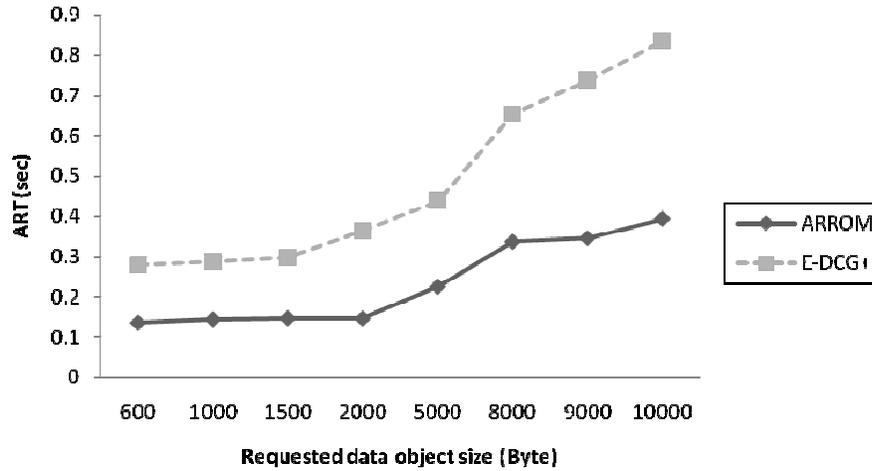


Figure 4: The ART of a request with different sizes of requested data object.

The same behavior is for average network throughput in Figure 5. Additionally, IR of ARROM for the average response time with different requested data object sizes equals to 51.99%. On the other hand, the IR of ARROM for the average network throughput with different requested data object sizes equals to 6.68%. As we mentioned in chapter two, each replica reallocation needs to broadcast the new location for all MHs in the group. This broadcasting reduces IR of average network throughput. In other words, increasing traffic in a network makes the network congested, which reduces the number of sent bytes per a unit of time (throughput).

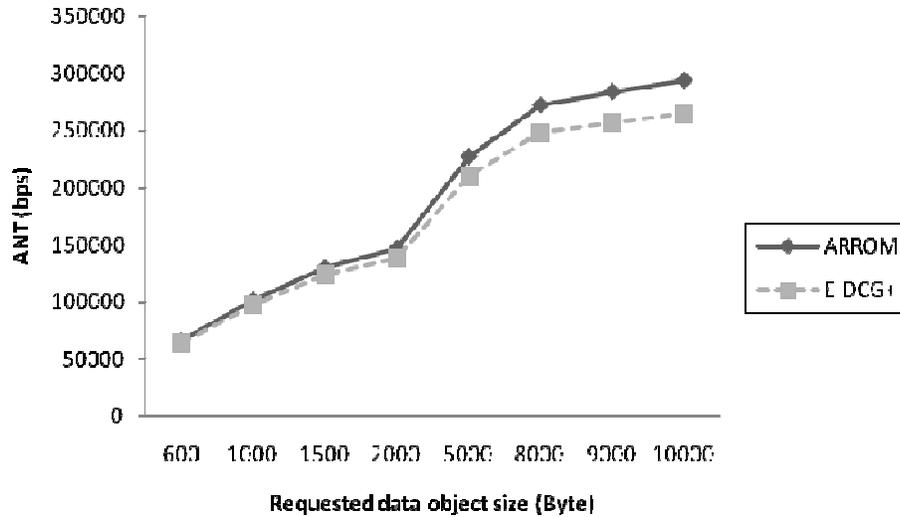


Figure 5: The ANT with different sizes of requested data object.

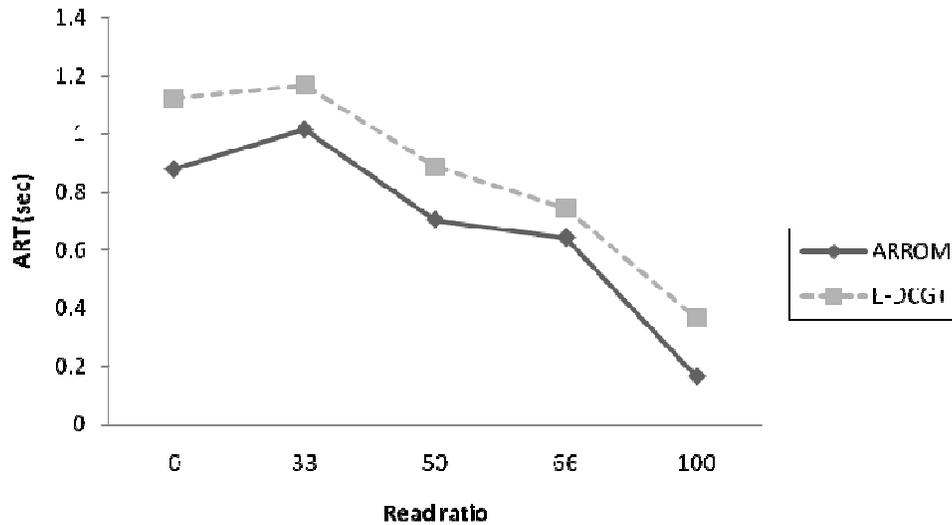


Figure 6: The ART of a request with different Read/Write ratios.

We studied the effects of read and write ratio on ARROM. Figure 6 shows the average response time of a request with different read ratios. The complement of each read ratio is write ratio, for example 50 read ratio has 50 write ratio at the same time. ARROM outperforms E-DCG+ in average response time with all read ratios. When all requests are write ones (*Read ratio = 0*), we get the maximum average response time difference between ARROM and E-DCG+. This difference decreases as the number of read requests increases. At the same time, ART of E-DCG+ and ART of ARROM decrease as the number of read requests increases. The reason behind this scale down is that write request has a number of propagation requests due to consistency management. Thus, when the number of write requests increases, the network traffic increases i.e. the average response time becomes longer due to congestion.

Figure 7 shows the average network throughput for different read ratios. When all requests are read ones (*Read ratio = 100*), we get the best average network throughput and the best difference between ARROM and E-DCG+. At the same time, average network throughput of both E-DCG+ and ARROM decrease as the number of write requests increases due to propagation process. Notice that when we have any number of write requests (*Read ratios = 0, 33, 50 and 66*), E-DCG+ outperforms ARROM due to the reallocation condition, which includes the cost between the new location and each server of the reallocated data object. However, ARROM may decide to reallocate a data object in insufficient location in terms of hops count between the new location and locations of other servers. This case increases the communication cost and leads us to another reallocation, which maybe insufficient too. This lopping affects average network throughput more than average response time, because of broadcasting.

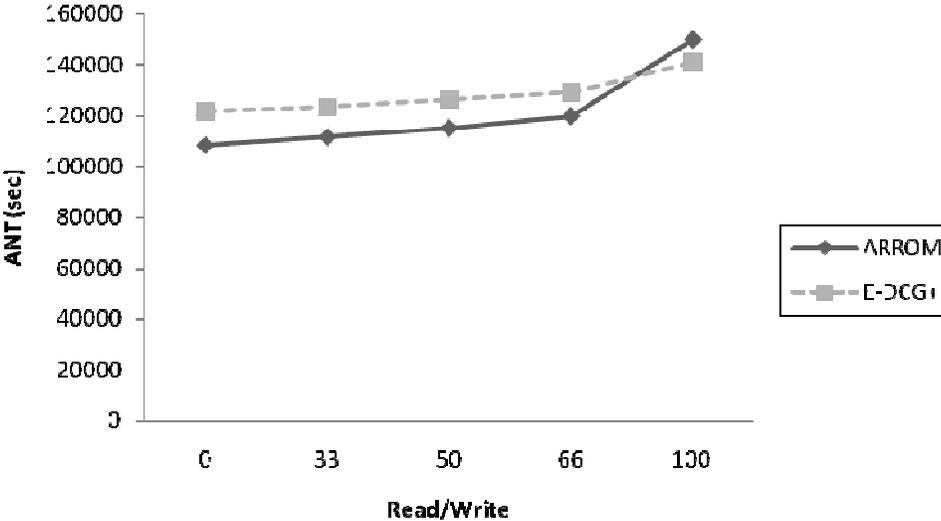


Figure 7: The ANT with different Read/Write ratios.

Finally, we studied the effects of radio transition power on ARROM. We show the radio transition power in dBm unit. 10 dBm, 15 dBm, 20 dBm and 40 dBm are equivalent to 282.6 meters, 376.78 meters, 502.5 meters and 1589 meters respectively.

Figure 8 shows the average response time of a request with different values of radio transition power. Notice that increasing radio transition power decreases the average response time, because of decreasing hops count between client and server. However, decreasing hops count makes ARROM performance very close to E-DCG+ performance, because reallocation frequencies decrease, see Figure 9. As a result, reallocation threshold must be decreased as the radio transition power increases to keep the ARROM performance high.

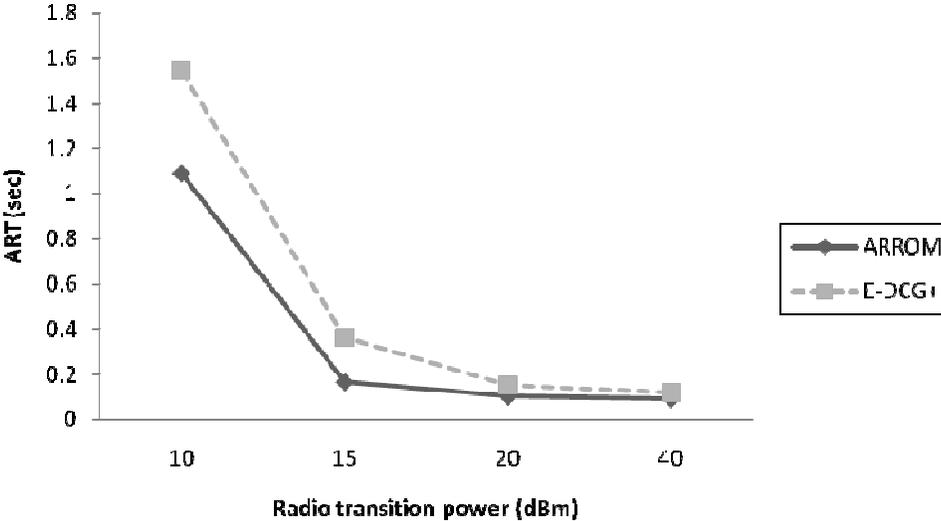


Figure 8: The ART of a request with different values of radio transition power.

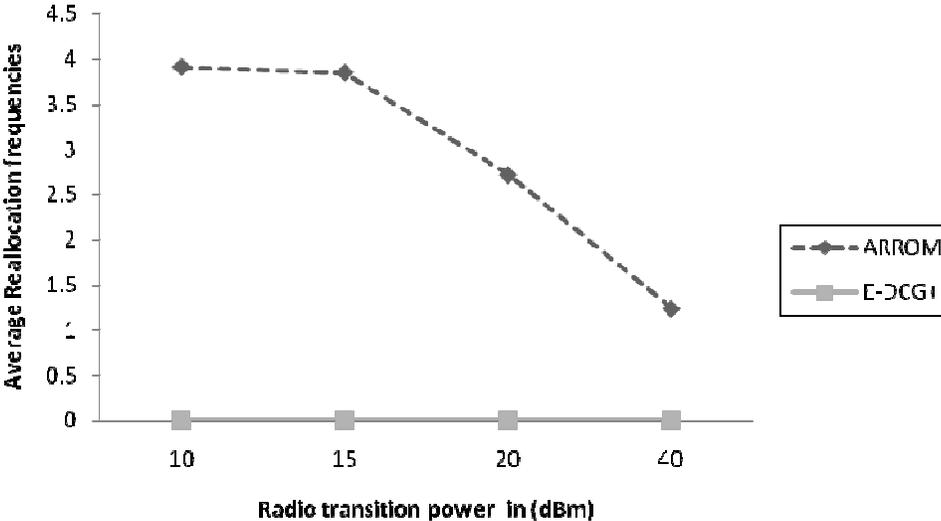


Figure 9: The average reallocation frequencies for different values of radio transition power.

Figure 10 shows that average network throughput increases as the radio transition power of nodes increases. This behavior is due to the reallocation frequencies as shown in Figure 9. As mentioned previously, each reallocation causes broadcast message, which increases the network traffic, therefore, the average network throughput is decreased. Notice that ARROM gives better average network throughput than E-DCG+ approach for all radio transition power values.

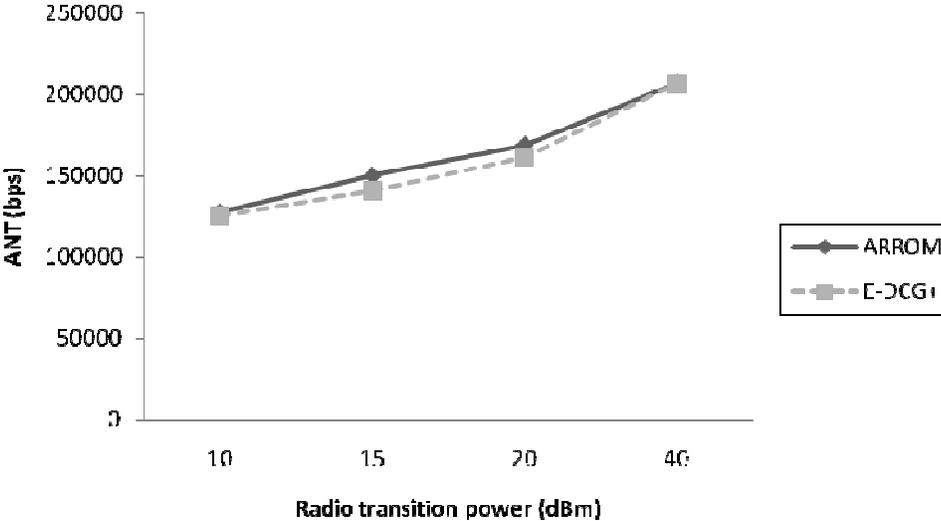


Figure 10: The ANT with different values of radio transition power.

The results show that ARROM approach outperforms E-DCG+ approach in terms of average response time and average network throughput. Table 3 shows the improvement ratio of ARROM using equation (7).

Table 3: The improvement ratios of ARROM with different parameters

Parameter	IR of average response time	IR of average network throughput
reallocation threshold	46.23%	4.41%
requested data object size	51.99%	6.68%
Read/Write ratio	24.72%	-6.76%
radio transition power	36.12%	3.09%
Average IR	39.76%	1.86%

For average response time in IR of ARROM, we assumed the ART of E-DCG+ as the max value and the ART of ARROM as the min value, because we decreased the average response time to improve it. The improvement of the average network throughput is by increasing it, therefore, the max value in IR of ARROM in average network throughput is the ANT of ARROM and the min value is the ANT of E-DCG+. In other words, ARROM generally decreases ART so the ART of ARROM is the min value and ARROM generally increases ANT so the ANT of ARROM is the max value.

As mentioned previously, E-DCG+ outperforms ARROM in the average network throughput with different read/write ratios due to reallocation condition, which includes the propagation cost of each write request. Therefore, ANT of different Read/Write ratios is negative value. In general, ARROM outperforms E-DCG+ in both the average response time by 39.76% and the average network throughput by 1.86%.

6 CONCLUSION

This paper presents a new replication system called Automated Reallocation of Replicas Over MANET (ARROM), which attempts to get the best replica allocation during its lifetime in terms of communication cost. To achieve that, ARROM uses appropriate long relocation period and reallocates existing replicas separately during relocation period in order to reduce the communication cost of both accessing and reallocating replicas, which improves the usage of MHs' poor resources. We evaluated ARROM using GloMoSim simulator with four parameters, which are reallocation threshold, data object size, read/write ratio and radio transition power.

ARROM outperforms E-DCG+ in the average response time and in the average network throughput with different reallocation thresholds. In general, ARROM decreases the average response time by 46.23% and increases the network throughput by 4.41%. In terms of data object size, ARROM gives better results for the average response time by 51.99% and average network throughput by 6.68%. For read/write ratio parameter ARROM outperforms E-DCG+ in the average response time by 24.72%, but E-DCG+ outperforms ARROM in the average network throughput by 6.76%. This weak point is due to reallocation condition. Finally, our proposed ARROM gives better results for the average response time by 36.12% and the average network throughput by 3.09% using different radio transition power.

In general, simulation results show that ARROM decreases the average response time compared with E-DCG+ approach by 39.76%. Furthermore, ARROM increases the average network throughput by 1.86%. Additionally, we do not recommend using ARROM in an environment, whose requests are mostly write ones such as in registration systems. Especially, if increasing the

average network throughput is more important than decreasing the average response time in such environment.

7 FUTURE WORKS

In the near future, we will enhance ARROM to work effectively with write operations; either by ignoring propagation cost in reallocation condition, or by changing the threshold dynamically. We will study the effects of more parameters on ARROM such as the mobility level of MHs. Additionally; we will insert the reallocation algorithm to other methods for example DRAM to get an enhancement method. At the same time, we want to improve the reallocation condition to include more performance metrics somehow, such as energy power of MHs and the bandwidth usage of MHs. In terms of security, we will use encryption and decryption techniques in transferring data objects to protect them from intruders.

We will improve ARROM by adding an algorithm that reallocates replicas even if one of their servers has a request in progress.

REFERENCES

- [1] Bajaj Lokesh, Takai Mineo, Ahuja Rajat, Tang Ken, Bagrodia Rajive and Gerla Mario (1999), GloMoSim: A Scalable Network Simulation Environment, Technical report for Computer Science Department CA 90095, University of California.
- [2] Elmasri Ramez and Navathe Shamkant B. (2007), Fundamentals of Database Systems, (5th), United States of America: Addison Wesley.
- [3] Fife Leslie D. and Gruenwald Le (2003), Research Issues for Data Communication in Mobile Ad-Hoc Network Database Systems, Proceedings of the SIGMOD Record, Vol. 32, No. 2, June 2003.
- [4] Guy Richard, Reither Peter, Ratner David, Gunter Michial, Ma Wilkie and Popek Gerald (1998), Rumor: Mobile Data access Through Optimistic Peer-to-Peer Replication, This work was supported by the United States Defense Advanced Research Projects Agency under contract number DABT63-94-C-0080 (1998).
- [5] Hara Takahiro (2003), Replica Allocation Methods in Ad Hoc Networks with Data Update, Netherlands: Kluwer Academic Publisher, Mobile Networks and Applications 8,343-354, 2003.
- [6] Hara Takahiro, Loh Yin-Huei and Nishio Shojiro (2003), Data Replication Methods Based on the Stability of Radio Links in Ad Hoc Networks, Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA'03) 1529-4188/03, 2003 IEEE.
- [7] Hara Takahiro (2005), Data Replication Issues in Mobile Ad Hoc Networks, Proceedings of the 16th International Workshop on Database and Expert Systems Applications (DEXA'05)1529-4188/05 2005 IEEE.
- [8] Hara Takahiro (2006), Data Replication and Update Management in Mobile Ad Hoc Networks (Invited Paper), Proceedings of the 17th International Conference on Database and Expert Systems Applications (DEXA'06)0-7695-2641-1/06 2006 IEEE.
- [9] Hara Takahiro and Madria Sanjay K. (2006), Data Replication for Improving Data Accessibility in Ad Hoc Networks, Proceedings of the IEEE CS, CASS, ComSoc, IES, and SPS, 1536-1233/06 2006 IEEE.
- [10] Hayashi Hideki, Hara Takahiro and Nishio Shojiro (2005, a), A Replica Allocation Method Adapting to Topology Changes in Ad Hoc Networks, Proceedings of DEXA 2005, and Proceedings of LNCS 3588, pp. 868-878, 2005.
- [11] Hayashi Hideki, Hara Takahiro and Nishio Shojiro (2005, b), Updated data dissemination methods for updating old replicas in ad hoc networks, Proceedings of the Pers Ubiquit Comput 9:273-283, published online: 15 January 2005.

- [12] Hayashi Hideki, Hara Takahiro and Nishio Shojiro (2006), On Updated Data Dissemination Exploiting an Epidemic Model in Ad Hoc Networks, Proceedings of the BioADIT 2006, LNCS 3853, PP. 306-321, 2006.
- [13] Huang Jiun-Long, Chen Ming-Syan and Peng Wen-Chih (2003), Exploring Group Mobility for Replica Data Allocation in a Mobile Environment, Proceedings of the CIKM'03, November 3–8, 2003, Copyright 2003 ACM 1581137230/03/0011.
- [14] Mondal Anirban, Madria Sanjay Kumar and Kitsuregawa Masaru (2006), CADRE: A Collaborative replica allocation and deallocation approach for Mobile-P2P networks, Proceedings of the 10th International Database Engineering and Applications Symposium (IDEAS'06) 0-7695-2577-6/06 2006 IEEE.
- [15] Nuevo Jorge (2004), A Comprehensible GloMoSim Tutorial, March 4, 2004.
- [16] Özsu M. Tamer and Valduriez Patrick (1999), Principles of Distributed Database Systems, (2nd) United States of America: Prentice Hall, Paperback: 666 pages, (January 29, 1999).
- [17] Pabmanabhan Prasanna and Gruenwald Le (2006, a), DREAM: A Data Replication Technique for Real-Time Mobile Ad-hoc Network Databases*, Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)8-7695-2570-9/06 2006 IEEE. And Proceedings of the NFS Grant No. IIS-0312746.
- [18] Pabmanabhan Prasanna and Dr.Gruenwald Le (2006, b), MANAGING DATA REPLICATION IN MOBILE AD_HOC NETWORK DATABASES (Invited Paper)*, Proceedings of the IEEE (1-4244-0429-0/06), And proceedings of the National Science Foundation (NSF), grant No. IIS-0312746.
- [19] Shi Ke and Haas Zygmunt J. (2007), Quantitative Analysis of Partition Statistics and their Impact on Data Replication in MANETs, Proceedings of the 6th International Symposium on Parallel and Distributed Computing (ISPDC'07) [0-7695-2936-4/07 2007 IEEE.
- [20] Takai Mineo, Martin Jay, Meyer Richard, Park Brian and Song Ha Yoon (1999), PARSEC User Manual, PARSEC was developed by members of the UCLA Parallel Computing Laboratory, written by Richard A. Meyer and Rajive Bagrodia, Revised in September 1999.
- [21] Zeng Xiang, Bagrodia Rajive and Gerla Mario (1998), GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks, Proceedings of the IEEE 1087-4097/98.
- [22] Cheng R-H., Wu T-K., Yu C. W., and Kuo C-H. (2006), An Altitude Based Dynamic Routing Scheme for Ad Hoc Networks, Proceedings of the WASA 2006 and of the LNCS 4138, pp. 609 – 619, 2006, Published in Springer-Verlag Berlin Heidelberg 2006.
- [23] Bajaj L., Takai M., Ahuja R., Tang K., Bagrodia R. and Gerla M. (1999), GloMoSim: A Scalable Network Simulation Environment, Technical report for Computer Science Department CA 90095, University of California.
- [24] Ms.RESHMA S NAIR , Mrs.A.Vegi Fernando, ME CSE (2013), Achieving Data Availability and Reducing Query Delay Using Data Reapplication Technique in MANETS, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3, March 2013 ISSN: 2277 128X.
- [25] T.H.Feroz Khan (2013), ON MAXIMIZING DATA ACCESSIBILITY IN MOBILE AD HOC NETWORK , International Journal of Engineering Science and Technology, e-ISSN : 0975-5462 , Vol. 5 No. 04S Apr 2013
- [26] Rashid Azeem and Muhammad Ishfaq Ahmad Khan (2012), Techniques about Data Replication for Mobile Ad-hoc Network Databases, International Journal Of Multidisciplinary Sciences And Engineering, VOL. 3, NO. 5, MAY 2012.
- [27] Natarajan Meghanathan and Chih-Lin Hu (2011), Special Issue on “Networks and Communications”, International Journal of Computer Networks & Communications (IJCNC) Vol.3, No.3, May 2011i IJCNC 2011.

- [28] B R Sujatha and M V Satyanarayana (2009), IMPROVED NETWORK CONNECTIVITY IN MANETS, International Journal of Computer Networks & Communications (IJCNC), Vol.1, No.3, October 2009.
- [29] Emre Atsan and Öznur Özkasap (2013), SCALAR: Scalable data lookup and replication protocol for mobile ad hoc networks, ELSEVEIR, E. Atsan, Ö. Özkasap/Computer Networks 57 (2013) 3654–3672.
- [30] K.P.Shanmuga Priya and V.Seethalakshmi (2013), Replica Allocation In Mobile Adhoc Network For Improving Data Accessibility Using SCF- Tree, International Journal of Modern Engineering Research (IJMER), Vol.3, Issue.2, March- April. 2013 pp-915-919, ISSN: 2249-6645.
- [31] P Mukilan and A Wahi (2012),CDRA: Consistency based Data Replication Algorithm for MANET, International Journal of Computer Applications, 51(14):1-8, August 2012. Published by Foundation of Computer Science, New York, USA.