Cluster Based Data Replication Technique Based on Mobility Prediction in Mobile Ad Hoc Networks

Mohammed Qayyum[1], Khaleel Ur Rahman Khan[2], and Mohammed Nazeer[3]

[1] Dept. of Computer Engg, King Khalid University, Abha, Saudi Arabia
mdqayyum.se@gmail.com
[2] Dept. of CSE, ACE Engineering College, Hyderabad, India
khaleelrkhan@aceec.ac.in
[3] Dept. of CSE, Muffakham Jha College of Engineering and Technologhy, Hyderabad, India
nazeer857@yahoo.co.in

Abstract. The mobile database system in a MANET is a dynamic distributed database system, which is composed of some mobile MHs. The key issues in MANETs for mobile database are: How to optimize mobile queries, cache and replicate data, manage transactions and routing. In this proposal, we wish to take the problems of data replication in solving the mobile database issues. Replication of data in a MANET environment focuses on to improve reliability and availability of data to the mobile clients (node). There are many issues revolving around replication of data in such a scenario like power, server and node mobility, networking partition and frequent disconnection. We propose a cluster based data replication technique for replication of data and to overcome the issues related to node mobility or disconnection problem in MANET environment. Our approach has two phases; initial phase consists of formation of cluster and cluster head and in the second phase, the distributions of data (replicated data) to the respective cluster head. By NS2 simulation, we will show that our proposed technique attains better data consistency and accuracy with reduced delay and overhead.

1 Introduction

1.1 Mobile Ad-Hoc Network (MANET)

A mobile ad hoc network (MANET) is a compilation of autonomous, mobile, wireless devices which forms a communications network even in the absence of fixed infrastructure. The primary goal of MANET network designers is to provide a self-protecting, "dynamic, self-forming, and self-healing network" for nodes on the move. [1] Each MANET node may move arbitrary and dynamically there by connected to form network depending on their positions and transmission range and as well as can act as a self router. The topology of the ad hoc network depends on the transmission power of the nodes and the location of the Mobile Nodes, which may change with time [2].

S.C. Satapathy et al. (eds.), *Emerging ICT for Bridging the Future – Volume 2,*
Advances in Intelligent Systems and Computing 338, DOI: 10.1007/978-3-319-13731-5_35

The feature of Ad hoc networks has an added advantage with respect to quick deployment and easy reconfiguration, which makes these system an ideal in situations where installing an infrastructure is too expensive or too susceptible.[5] MANETs have applicability in several areas like [4];

- Soldiers relaying information for situational awareness on the battlefield.
- Emergency disaster relief personnel coordinating efforts after a fire, hurricane, or earthquake.
- Personal area and home networking
- Business associates sharing information during a meeting.
- Attendees using laptop computers to participate in an interactive conference.
- Location-based services and sensor networks.

The three major drawback related to the quality of service in MANET are bandwidth constraints, dynamic topology of MANET and the limited processing and storing capacity of mobile nodes.[3]

1.2 Mobile Databases in MANET

The mobile database system in a MANET is a dynamic distributed database system, which is composed of some Mobile Heads (MHs). Each MH comprises of local database system. [8]. In storing capacity of mobile nodes, Transaction Manager (TM) of a mobile multi database management system is accountable for providing dependable and steady units of computing to all its users. [6]

Nodes in a MANET can be classified by there capabilities that is as a client or a server.

1. A Client or *Small Mobile Host (SMH)* is a node with reduced processing, storage, communication and power resources.
2. A Server or *Large Mobile Host (LMH)* is a node having a larger share of resources.

Servers, due to their larger capacity contain the complete DBMS and abide primary responsibility for data transmission and satisfying client queries. Clients typically have enough resources to cache portions of the database as well as some DBMS query and processing modules. [7]

There are three layers in a mobile distributed database system; [8]

- **The application layer** – In this layer the user queries are accepted.
- **The middleware layer** – In this layer, Queries are processed and transmitted to the middleware of other MHs in the network. The middleware of a MH sends queries to the local database system. After the database finishes executing a query, the results are transmitted from the middleware layer to the application layer and then the results are returned to the user.
- **The database layer** – In this layer all the information and data are stored.

The middleware layer is the core of the mobile distributed database system and it is divided into three sub-layers [8];

- **The network layer** – It manages location information of nodes, divides nodes into groups, routs data packets between the query layer, and the cache layer.
- **The cache layer** – It stores the data which are accessed frequently by the query nodes or their neighbors.
- **The query layer** – It parses the syntax of user queries and determines the query types

1.3 Issues of Mobile Databases in MANET

The key issues in MANETs for mobile database are: How to optimize mobile queries, cache and replicate data, manage transactions and routing. [8] Issues related to mobile database in MANET are as; [7,9,10,11]

- **Power** - All mobile devices in the MANET are battery powered. In traditional mobile networks, only the power needs of the clients are considered. But in the present scenario's, the power of the server, which provides DBMS data services, is perhaps more important as it provides DBMS services to potentially many clients.
- **Mobility of the nodes** - Due to the dynamic nature of a MANET, it exhibits frequent and unpredictable topology changes. The MANET not only operates within the ad-hoc network, but may also require access to a public fixed network. MANETs therefore should be able to adapt the traffic and propagation conditions to the mobility patterns of the nodes.
- **Resource availability** – A node should supply mechanisms for proficient use of processing, memory and communication resources, while maintaining low power consumption. A node should bring about its basic operations without resources exhaustion.
- **Response Time** - Regardless of the method of communication used, access time and tuning time must be considered. *Tuning time* is the measure of the amount of time each node spends in Active Mode. This is the time of maximum power consumption for a client. *Access time* measures the sensitivity of the algorithm. It refers to the amount of time a client must wait to receive an answer to a database query.
- **Quality of Service** – Nodes become disengaged for a variety of reasons. This may be due to location or lack of power, dynamic nature and redundancy. The accuracy of information stored at each node: server and client are alike. When portions of the network become separated for a time, data accuracy may become impossible.
- **Data Broadcast** - The size and contents of a broadcast have an effect on power consumption and the frequency of data queries. If the broadcast is too outsized, unnecessary information may be broadcast. If too little information, then wrong information is broadcasted. Thus increasing the on-demand requests. Also if several servers attempt to broadcast simultaneously, there will be a collision and the broadcast of all will be jumbled.

The issues mentioned are some of the major issues which we come across in the mobile database in MANET. Based on the above issues, it is necessary to provide a solution for the mobile database management. Below are some recent literature works which throw a light on the problems and its major solutions.

2 Related Work

Jin-Woo Song et.al. [12] have proposed a lucrative replica server allocation algorithms which induces a present algorithm with more careful analysis of the moving patterns of mobile device users. Here existing four algorithms are studied and customized. First a modified "vertex occurrence counts (VOC) -neighbor reduction" algorithm is introduced for lowering the vertex occurrence counts of the neighbors of the cells selected in VOC. A similar algorithm, called the greedy set-cover algorithm, which designate, replicated servers so that they are not allocated to the cells adjacent to each other. The replicated server clustering algorithm exploits the k-means clustering. The algorithm returns a set of clusters; the center of each cluster becomes the location for a replicated server.

Anita Vallur et.al [13] have proposed a data replication technique called REALM (REplication of data for A Logical group based MANET database) for logically group based MANET real-time databases. REALM, groups mobile owner based on the data items which they need to access. Mobile hosts that access the same set of data items are grouped into the same logical group. Group membership of mobile owner helps in identifying the data items that any mobile owner in the network will need to access, as well as to identify the most frequently accessed data item on every server. REALM tries to increase the percentage of successful transactions in real-time database of MANET.

Deniz Altınbuken and Oznur Ozkasap [14] have come upon with an comprehensive SCALAR (Scalable Data Lookup and Reactive Replication) framework for updated data, named as SCALARUP. It presents a replication service for updated data items in a fully distributed approach. Here the data are updated randomly by the owner of a data item and a new write frequency value is transmitted to the system. When a node acquires a new write frequency value for a data item, it restores its write frequencies table and invalidates old replicas of the updated item. During this restoration process the message overhead is altered only in terms of the write frequency broadcast messages. This dynamic backbone construction algorithm used in this paper, minimizes the time required to search and retrieve data for replication.

Thomas Plagemann et.al. [15] have proposed an approach for reliability management in shared data spaces for emergency and rescue operations. The work shows that the application specific requirement tries to solve the problem of consistency management for optimistic replication in Sparse MANETs. The problem of consistency management during replica synchronization is labeled by showing the need of data deleted at the crisis and rescue environment.

Prasanna Padmanabhan and Le Gruenwald [16] have proposed a data replication technique called DREAM for real time mobile ad-hoc network database systems. It focuses on data convenience while addressing the issue of power restriction by replicating hot data items before cold data items at the servers that have high remaining power. It handles the real-time transaction issue by giving a superior priority for replicating data items that are accessed frequently by firm transactions than those accessed frequently by soft 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.

Matiasko K and Zabovskı M [17] have proposed an algorithm for dynamic re-allocation of data with a mobile computers incorporated in replication schema. The replication schema used here consists of two steps; Step 1: Test of expansion, implementation is done after the particular number of transactions to develop replication schema expansion when changes improve solution. Step 1: Test of contraction solves the problem with wired nodes included in the replication schema. The motivation is given by assumption that is easier to prevent site failure due to the communication network problem than to solve its failure.

Hao Yu et.al [18] have proposed a cluster-based optimistic replication management system for large-scale MANETs named as Distributed Hash Table Replication (DHTR) It uses a dispersed hash table technique and a dispersed replica information directory to enhance the efficiency of update propagation. Distributed clustering offers a method of maintaining hierarchical structures in ad-hoc networks to simplify the system control and decrease overhead messages and Distributed hash table technology is normally applied in peer-to-peer networking environments to help the user locate the resources quickly.

3 Proposed Scheme

Replication of data in a MANET environment focuses on to enhance dependability and availability of data to the mobile clients (node). There are many issues revolve around replication of data in such a scenario like power, server and node mobility, networking partition and frequent disconnection [19].We propose an approach for replication of data and to overcome the issues related to node mobility or disconnection problem in MANET environment.

Our approach has two phases; initial phase consists of formation of cluster and cluster head and in the second phase, the distributions of data (replicated data) to the respective cluster head.

3.1 Initial phase – Creating "Basket Node"

In the initial phase, nodes are clustered based on two factors; received signal strength of nodes and distance from past movements. The node which is most stable is elected as the cluster head. These cluster head act ultimately as "basket nodes" (those cluster heads which posses the replicated data) and share with the member nodes (according to the queries). We will discuss about the calculation of received signal strength of nodes and distance from past movements. We compare the two factors and select the best stable node as cluster head.

3.1.1 Received Signal Strength of Nodes

In the Received signal strength [20], we calculate the most stable node with respect to the signal strength received. In the signal strength scheme, each node communicate with there one hop neighbour with sending/receiving an "alive" message. Each node calculates the pairwise relative mobility metrics (RM) by reception of two successive "alive" message. The pairwise relative mobility metrics is calculated by;

$$\text{RM (a,b)} = 10\log_{10}\frac{\overset{\text{new}}{Rg(b)\times P}}{\overset{\text{old}}{Rg(b)\times P}} \tag{1}$$

Where, RM(a,b) denotes the relative mobility metric of node "a" with respect to node "b". $\frac{Rg(b)\times P}{Rg(b)\times P}$ is the ratio of the new and old power level product of received signal strength detected at the receiving node "a". Before sending the next broadcast packet to its neighbors, a node computes the aggregate relative mobility metric (RMAGG), which is calculated by the expected value of the squares of the relative mobility samples from neighbors given by;

$$\text{RMAGG (a)} = \text{E}\,[\,\text{RM(a, i)}^2\,] \tag{2}$$

The mobility of the nodes can be calculated by ratio of the new message and old message. If the power of new message is less than old message, then RM value will be negative, which indicates that the two nodes are moving away with respect to each other. On the other hand, if the power of new message is more than old message, then RM value will be positive, which indicates that the two nodes are moving away with respect to each other.

3.1.2 Distance from Past Movements

The distance from past movements is calculated by "Average Movement (AM) and Range time (RT)" [21] equation. Average Movement represents the average mobility of a resource and/or user based on user and resource mobility. It is calculated based on two recent communications between user/initiator and resource with respect to the user/initiator. AM is calculated by;

$$\text{AM} = \left|\,\text{old location - new location - user average movement}\,\right| \tag{3}$$

The location provides the distance between user and resource, where user average movement is the location of users past location history. This provide us with the Range time (RT), which is given by

$$RT = \frac{(User\ Range - Dis\tan ce)}{Average\ Movement} \quad (4)$$

3.1.3 Combined Scheme

The initial period of all nodes is in the un-clustered random form. During a specific time period every node broadcasts two successive "alive" messages. During its reception, each node calculates the RMAGG and RT. These factors are stored in the neighbor table of each neighbor along with a time-out period (TO) set. A node receives the aggregate mobility values from its neighboring nodes, and then compares its own mobility value with those of its neighbors. If a node has the lowest value of RMAGG as well as, high RT amongst all its neighbors it assumes the status of a Cluster Head. The range leads to the formation of cluster boundaries. If a node is a neighbor of two cluster heads, then it becomes a "gateway" node.

If two neighboring nodes in an un-clustered state have the same value of RMAGG and RT, we resort to comparison of IDs and follow the Lowest-ID algorithm. That is the mobility metric of two cluster-head nodes is the same, and they are in contention for retaining the Cluster Head status, then the selection of the cluster-head is based on the Lowest-ID algorithm wherein the node with the lowest ID gets the status of the Cluster Head. In a mobile scenario, if a node with Cluster Member status with a low mobility moves into the range of another Cluster Head node with higher mobility, re-clustering is not triggered, but a Cluster conflict (CC) period is allowed for incidental contacts between passing nodes. If the nodes are in transmission range of each other even after the CC period has expired, re-clustering is triggered, and the node with the lower mobility metric assumes the status of Cluster Head.

These cluster head act ultimately as "basket nodes" which posses the replicated data and share with the member nodes according to the queries. We assume an offline server to collect queries from the nodes, after the cluster formation. The server maintains two tabular columns regarding node location and cluster details. During a query request, the node sends request message to the server. The message comprises of the query id (QID), the query (QRY), the node id (NID) and the cluster head (CH). After analyzing the query, the server cross checks both the table and locate the node and its cluster head position. After the node and cluster head positioning the server distributes the respective data according to the queries.

QID	QRY	NID	CH

Fig. 1. Query Request

3.2 Second Phase – Distribution of Data to Basket Nodes

In the second phase, the distribution of data to the "basket node" from the main server is done. This distribution (replication data's) depends on the queries from the nodes. According to the queries, the replicated data's are sent to the nearest cluster heads from which uninterrupted data services can be maintained.

Replication of data into a single cluster head increases the overhead of the cluster head as well as the power consumptions. In our approach, the offline server acts as a centralized distributor. The server intakes the queries and based on the clusters formed, the server can distribute the replicated data to nearest cluster heads according to the queries.

We also propose a distributed data replication, where the nearer cluster heads can distribute the data if the queries are similar. For example, when members of two nearer clusters (A and B) access data of similar queries, the server distributes the replicated data proportionally for both the cluster heads A and B. When cluster member of A seeks data which is stored in B, the cluster head of A seeks for the information through cluster head of B. In such a scenario, the overhead on cluster heads decreases due to the distribution of replicated data.

Also in our approach, we allocate a time period to each cluster head to analyze the consistency of data on cluster heads. If a cluster head is idle for allocated time period, then the corresponding replicated data is deleted. Otherwise it is updated according to the queries.

Algorithm
We describe our algorithm in the following step;

1. Creating "Basket Node"
1.1 Node sends successive "alive" message to form cluster.
1.2 Nodes are clustered based on two factors; received signal strength of nodes and distance from past movements.
1.3 With successive messages, nodes calculates RMAGG and RT
1.4 With the values of RMAGG and RT, Cluster head is elected. (if RMAGG is low and RT is high)
1.5 The elected cluster head maintains its members detail and transmit it to the server.
1.6 The server maintains table for node location and cluster details.

2. Distribution of data to basket nodes (refer figure 2)
2.1 The nodes send the query request to the server.
2.2 From the table in server, location of the nearest cluster head is determined. The data's related to the query details are sent to the cluster head.
2.3 If similar data are sent to nearer cluster heads, data can be distributed proportionally between the two cluster heads.
2.4 The query data are sent to the respective cluster member through cluster heads.
2.5 Queried data remain in the cluster head for the certain data time and are deleted if idle for allocated time period

Thus our approach tries to predict the node mobility to replicate the data, by determining the received signal strength of nodes and distance from past movements of nodes. We also reduce the cluster head overhead by timely refreshing or deleting the nodes and by distributing the replicated data proportionally to nearer cluster heads. Distribution of replicated data among the nearer cluster heads reduce the cluster head consumption of memory and power.

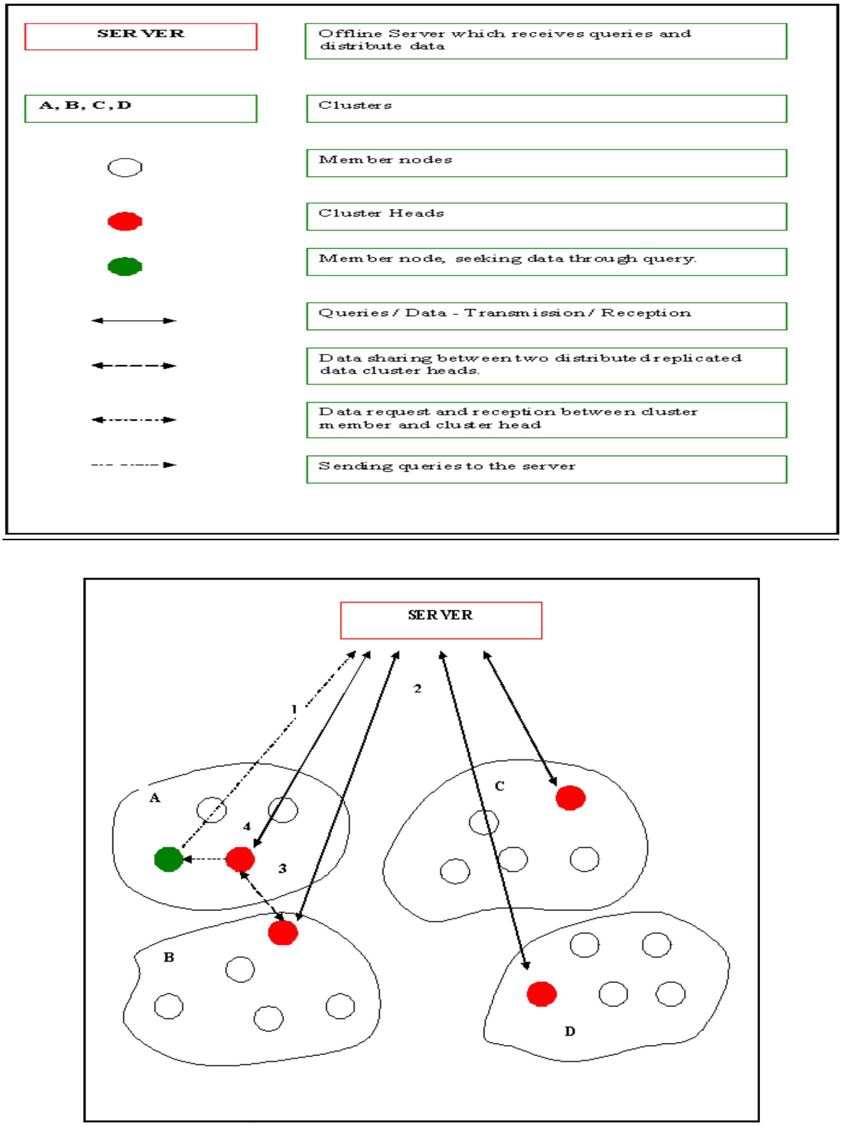

Fig. 2. Data Distribution

4 Simulation Results

4.1 Simulation Setup

This section deals with the experimental performance evaluation of our algorithms through simulations. In order to test our protocol, The NS2 simulation [22] is used. In our simulation, the channel capacity of mobile hosts is set as 2 Mbps. In the simulation, 50 mobile nodes move in a 600 meter x 600 meter region for 50 seconds simulation time. Initial locations and movements of the nodes are obtained using the random waypoint (RWP) model of NS2. All nodes have the same transmission range of 250 meters. The routing protocol used is AODV. The average speed of the mobile is varied from 5m/s to 20m/s.

In all the experiments, the following evaluation criteria have been employed. A comparison between the proposed CBDR technique and the REALM [13] scheme is performed.

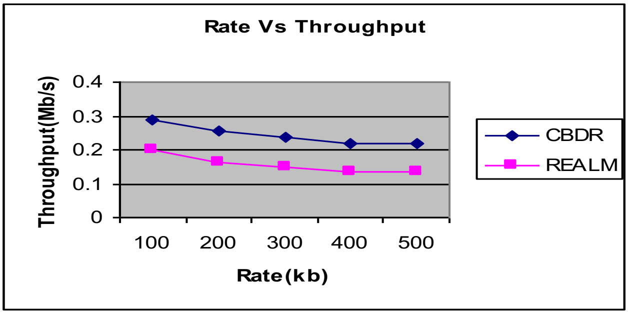

Fig. 3. Rate Vs Throughput

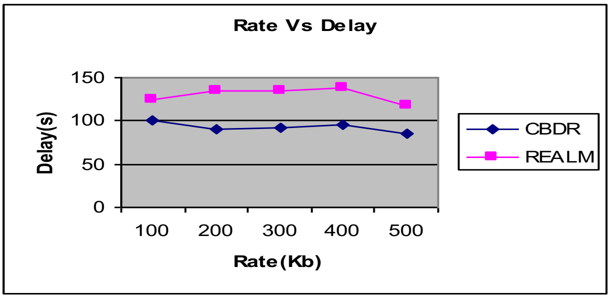

Fig. 4. Rate Vs End-to-End Delay

Fig 3 shows the average throughput for different traffic rates. Since CBDR uses cluster based data replication technique, throughput will be high. From the figure it can be observed that CBDR has higher throughput, when compared with REALM. Fig 4 shows average end-to-end delay for different traffic rates. Since the queries are processed by the resource efficient cluster heads, the end-to-end delay will be significantly less. From figure it can be observed that CBDR has less delay, when compared with REALM.

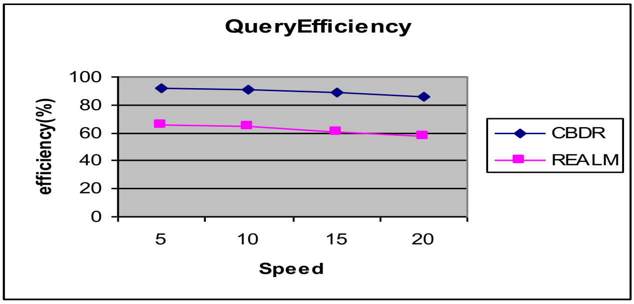

Fig. 5. Mobile Speed Vs Query Efficiency

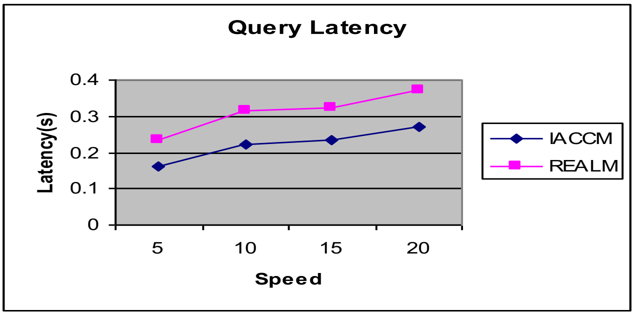

Fig. 6. Mobile Speed Vs End-to-End Delay

Fig 5 and 6 show the average query efficiency and latency for varying the mobile speed. As the mobile speed increases, it results in network disconnection or portioning, thereby reducing the query efficiency and increasing the latency. Since CBDR uses the cluster based data replication technique, the efficiency is high with low latency when compared with REALM.

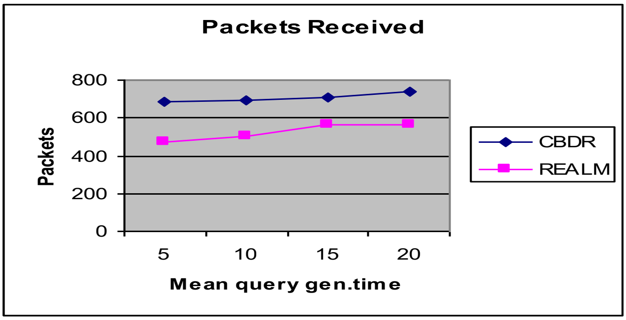

Fig. 7. Query generation timeVs Packets Received

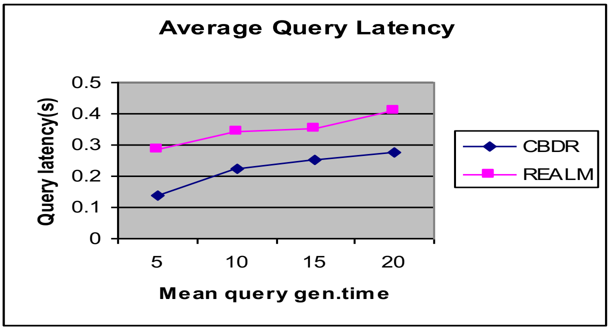

Fig. 8. Query generation timeVs Query Latency

Fig. 7 and Fig 8 show the results for the number of packets received and the average query latency, respectively, when query generation time *T*q is varied. Note that if several mobiles request for the same data item during the same interval, the data are replicated in nearby clusters. Hence number of packets received is more and query latency is less for CBDR when compared with REALM.

5 Conclusion

Replication of data in a MANET environment focuses on to improve reliability and availability of data to the mobile clients (node). There are many issues revolving around replication of data in such a scenario like power, server and node mobility, networking partition and frequent disconnection. We propose a dual phase scheme for replication of data and to overcome the issues related to node mobility or disconnection problem in MANET environment. The initial phase is used to form a cluster and a stable cluster head known as "basket nodes" using the two factors; received signal strength of nodes and distance from past movements. The offline server in our scheme acts as a centralized distributor which intakes the queries and based on the clusters formed will distribute the replicated data to nearest cluster heads according to the queries. Our scheme also proposes a distributed replicated method in which the nearer cluster heads can distribute the data if the queries are similar. We provide a periodic refresher to reduce overhead in basket nodes. Thus our scheme solves the mobility prediction as well as overhead problem in mobile database system in a MANET under dynamic distributed database system.

References

1. Orwat, M.E., Levin, T.E., Irvine, C.E.: An Ontological Approach to Secure MANET Management. In: Proceedings of the 2008 Third International Conference on Availability, Reliability and Security, pp. 787–794 (2008)

2. Saad, M.I.M., Ahmadnn, Z.: Performance Analysis of Random-Based Mobility Models in MANET Routing Protocol. European Journal of Scientific Research 32(4), 444–454 (2009)
3. Uma, M., Padmavathi, G.: A comparative study and performance evaluation of reactive quality of service routing protocols in Mobile Adhoc networks. Journal of Theoretical and Applied Information Technology 6(2) (2009)
4. Wu, B., Chen, J., Wu, J., Cardei, M.: A Survey on Attacks and Countermeasures in Mobile Ad Hoc Networks. In: Xiao, Y., Shen, X., Du, D.-Z. (eds.) Wireless/Mobile Network Security. Springer (2006)
5. Huang, Y., Jin, B., Cao, J., Sun, G., Feng, Y.: A Selective Push Algorithm for Cooperative Cache Consistency Maintenance over MANETs. In: Kuo, T.-W., Sha, E., Guo, M., Yang, L.T., Shao, Z. (eds.) EUC 2007. LNCS, vol. 4808, pp. 650–660. Springer, Heidelberg (2007)
6. Gruenwald, L., Banik, S.M.: A Power-Aware Technique to Manage Real-Time Database Transactions in Mobile Ad-Hoc Networks. In: Proceedings of the 12th International Workshop on Database and Expert Systems Applications, pp. 570–574 (2001)
7. Fife, L.D., Gruenwald, L.: Research Issues for Data Communication in Mobile Ad-Hoc Network Database Systems. ACM SIGMOD Record 32(2), 42–47 (2003)
8. Li, J., Li, Y., Thai, M.T., Li, J.: Data Caching and Query Processing in MANETs. JPCC 1(3), 169–178 (2005)
9. Hadim, S., Al-Jaroodi, J., Mohamed, N.: Middleware Issues and Approaches for Mobile Ad hoc Networks. In: Proceeding of IEEE Consumer Communications and Networking Conference (CCNC 2006), Las Vegas, Nevada (January 2006)
10. Artail, H., Safa, H., Pierre, S.: Database Caching in MANETs Based on Separation of Queries and Responses. In: Proceeding of WiMob 2005, Montreal, Canada (August 2005)
11. Pushpalatha, M., Venkatraman, R., Ramarao, T.: An Approach to Design an Efficient Data Replication Algorithm in Mobile Ad hoc Networks. IJRTE 1(2) (May 2009)
12. Song, J.-W., Lee, K.-J., Kim, T.-H., Yang, S.-B.: Effective Replicated Server Allocation Algorithms in Mobile computing Systems. IJWMN 1(2) (November 2009)
13. Vallur, A., Gruenwald, L., Hunter, N.: REALM: Replication of Data for a Logical Group Based MANET Database. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2008. LNCS, vol. 5181, pp. 172–185. Springer, Heidelberg (2008)
14. Altınbuken, D., Ozkasap, O.: ScalarUp: Scalable Data Lookup and Replication Framework for Updated Data. In: Proceeding of 5th International Advanced Technologies Symposium (IATS 2009), Karabuk, Turkey, May 13-15 (2009)
15. Plagemann, T., Munthe-Kaas, E., Goebel, V.: Reconsidering Consistency Management in Shared Data Spaces for Emergency and Rescue Applications. In: Model Management und Metadaten-Verwaltung (BTW-MDM 2007), Aachen (March 2007)
16. Padmanabhan, P., Gruenwald, L.: Managing Data Replication in Mobile Ad-Hoc Network Databases. In: International Conference on Collaborative Computing: Networking, Applications and Worksharing (November 2006)
17. Matiasko, K., Zabovskı, M.: Adaptive replication algorithm for mobile computers. Acta Electrotechnica et Informatica 8(2), 18–21 (2008)
18. Yu, H., Martin, P., Hassanein, H.: Cluster-based Replication for Large-scale Mobile Ad-hoc Networks. In: Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access International Workshop on Data Engineering for Wireless and Mobile Access, Vancouver, Canada, pp. 39–46 (2008)

19. Padmanabhan, P., Gruenwald, L., Vallur, A., Atiquzzaman, M.: A survey of data replication techniques for mobile ad hoc network databases. The International Journal on Very Large Data Bases 17(5), 1143–1164 (2008)
20. Basu, P., Khan, N., Little, T.D.C.: A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks. In: Proceedings of the 21st ICDCS, April 16-19, p. 413 (2001)
21. Farooq, U., Mahfooz, S., Khalil, W.: An Efficient Resource Prediction Model for Mobile Grid Environments. In: Proceeding of 1st Annual Post Graduate Symposium on the Convergence of Telecommunication Networking and Broadcasting, IEE/EPSRC. Liverpool John Moores University, UK, June 26-27 (2006)
22. Network Simulator, http://www.isi.edu/nsnam/ns