International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

Commit Protocols improvement in Mobile Environments based on Concurrency Control

Arti Nag, Sambhav Agrawal
Computer Science and Engineering,
Bansal Institute of Engineering and Technology, Lucknow, India

Abstract: In modern time the mobile computing based applications needs advanced data synchronization. In this paper we have developed and investigated the complete database distributed among wireless components as in mobile switching stations. In this approach the entire database is being distributed in wireless components of the computer systems. Some of the parameters that influence and complicate database management are design of database and replication of database. We have developed a mobile environment protocol that can handle the distributed database of several clients using priority based concurrency control mechanism with considerations of Hand Off situation. For this case we have developed our algorithm using MATLAB 2010.An advanced priority queue mechanism is applied to reduce the transaction aborting issues.

Keywords: Mobile Computing, MDRTDBS, Optimistic Concurrency Control , WSN.

1. Introduction:

With the rapid advances in mobile computing technology, there is an increasing demand for processing real-time transactions in a mobile environment. Based on the High Priority Two Phase Locking (HP-2PL) scheme, we propose a distributed real-time locking protocol, called Distributed High Priority Two Phase Locking (DHP-2PL), for MDRTDBS. In the protocol, the characteristics of a mobile computing system are considered in resolving lock conflicts. Two strategies are proposed to further improve the system performance and to reduce the impact of mobile network on the performance of the DHP-2PL: (1) A transaction shipping approach is proposed to process transactions in a mobile environment by exploring the well-defined behavior of real-time transactions. (2) We explore the application semantics of real-time database applications by adopting the notion of similarity in concurrency control to further reduce the number of transaction restarts due to priority inversion, which could be very costly in a mobile network. A detailed simulation model of a MDRTDBS has been developed, and a series of simulation experiments have been conducted to evaluate the performance of the proposed approaches and the effectiveness of using similarity for concurrency control in MDRTDBS. Recent advances in wireless communication technology have made mobile information services a reality. A number of novel mobile computing systems, such as tele-medicine systems, realtime traffic information and navigation systems, and mobile Internet stock trading systems, are emerging as mobile users require instant access to information using their palmtops, personal digital assistant (PDA) and notebook computers. Mobile computing technology not just only improves the distribution and flow of information, but at the same time, it also greatly increases the functionality of real-time database applications. The realization of "instant" information access over a mobile network relies on real-time processing of transactions and it makes the timeliness of data accesses an important issue. As a result, research on processing soft real-time transactions in mobile distributed real-time database systems (MDRTDBS) is receiving growing attention in recent years. Universal access and management of information has been one of the driving forces in the evolution of computer technology. Central computing gave the ability to perform large and complex computations and advanced information manipulation. Advances in networking connected computers together and led to distributed computing. Web technology and the Internet went even further to provide hyper-linked information access and global computing. However, restricting access stations to physical locations limits the boundary of the vision. The real global network can be achieved only via the ability to compute and access information from anywhere and anytime. This is the fundamental wish that motivates mobile computing. This evolution is the cumulative result of both hardware and software advances at various levels motivated by tangible application needs. Infrastructure research on communications and networking is essential for realizing wireless systems. Equally important is the design and implementation of data management applications for these systems, a task directly affected by the characteristics of the wireless medium and the resulting mobility of data resources and computation. Although a relatively new area, mobile data management has provoked a proliferation of research efforts motivated both by a great market potential and by many challenging research problems. The focus of Data Management for Mobile Computing is on the impact of mobile computing on data management beyond the networking level. The purpose is to provide a thorough and cohesive overview of recent advances in wireless and mobile data management. Data Management for Mobile Computing provides a single source for researchers and practitioners who want to keep abreast of the latest innovations in the field.

Further evolution of Internet technologies will yield a wide-area network based on component-oriented, dynamic applications, which will support efficient, scalable resource sharing for a large number of mobile and nomadic users. As users gradually grow to rely on the Internet as an indispensable tool, most users will become mobile or nomadic users, or both. While mobile users access the Internet from a portable computer, nomadic users may move from terminal to terminal. In either case, a user would ideally be able to accomplish the same tasks with equal ease from any location either on his portable computer or at any Internet-connected terminal. Many other issues also have in the field of distributed systems, database management, transaction management, operating or file systems, information retrieval or dissemination, and web computing. Mobile computing is a revolutionary technology, born as a result of remarkable advance in the development of computer hardware and wireless communication. It enables us to access information anytime and anywhere even in the absence of physical network connection. More recently, there has been increasing interest in introducing ad hoc network into mobile computing, resulting in a new distributed computing style known as peer-to-peer (P2P) computing. In this paper, we discuss the data management issues in mobile and P2P environments. The use of wireless communication makes the data availability the most important problem here, so we focus on the problem of data availability and provide detailed discussion about replicating mobile databases. Not only that, we extend our discussion to mobile-P2P environment. At the end, we discuss the general data management issues in P2P environment. To design efficient data management policies to support the dissemination of large amount of information to different mobile users are the big issues.

2. Related Work:

Kam-Yiu Lam et. al. (2000) [16] they proposed a distributed real-time locking protocol, called Distributed High Priority Two Phase Locking (DHP-2PL), for MDRTDBS. With the rapid advances in mobile computing technology, there is an increasing demand for processing real-time transactions in a mobile environment. Based on the High Priority Two Phase Locking (HP-2PL) scheme. In the protocol, the characteristics of a mobile computing system are considered in resolving lock conflicts. Two strategies are proposed to further improve the system performance and to reduce the impact of mobile network on the performance of the DHP-2PL: (1) A transaction shipping approach is proposed to process transactions in a mobile environment by exploring the well-defined behavior of real-time transactions. (2) We explore the application semantics of real-time database applications by adopting the notion of similarity in concurrency control to further reduce the number of transaction restarts due to priority inversion, which could be very costly in a mobile network. A detailed simulation model of a MDRTDBS has been developed, and a series of simulation experiments have been conducted to evaluate the performance of the proposed approaches and the effectiveness of using similarity for concurrency control in MDRTDBS.

The distributed transaction commit problem requires reaching agreement on whether a transaction is committed or aborted. The classic Two-Phase Commit protocol blocks if the coordinator fails. Fault-tolerant consensus algorithms also reach agreement, but do not block whenever any majority of the processes are working. The Paxos Commit algorithm runs a Paxos consensus algorithm on the commit/abort decision of each participant to obtain a transaction commit protocol that uses 2F + 1 coordinators and makes progress if at least F+1 of them are working properly. Paxos Commit has the same stable-storage write delay, and can be implemented to have the same message delay in the fault-free case, as Two-Phase Commit, but it uses more messages. The classic Two-Phase Commit algorithm is obtained as the special F = 0 case of the Paxos Commit algorithm proposed by Jim Gray et al. (2004) [17].
Two-Phase Commit is the classical transaction commit protocol. Indeed, it is sometimes thought to be synonymous with transaction commit [17]. Two-Phase Commit is not fault tolerant because it uses a single coordinator whose failure can cause the protocol to block. We have introduced Paxos Commit, a new transaction commit protocol that uses multiple coordinators and makes progress if a majority of them are working. Hence, 2F + 1 coordinators can make progress even if F of them are faulty. Two-Phase Commit is isomorphic to Paxos Commit with a single coordinator. In the normal, failure-free case, Paxos Commit requires one more message delay than Two-Phase Commit. This extra message delay is eliminated by Faster Paxos Commit, which has the theoretically minimal message delay for a non-blocking protocol.

Non-blocking transaction commit protocols were first proposed in the early 1980s [3, 4, 19]. The initial algorithms had two message delays more than Two-Phase Commit in the failure-free case; later algorithms reduced this to one extra message delay [3]. All of these algorithms used a coordinator process and assumed that two different processes could never both believe they were the coordinator an assumption that cannot be implemented in a purely asynchronous system. Transient network failures could cause them to violate the consistency requirement of transaction commit. It is easy to implement non-blocking commit using a consensus algorithm an observation also made in the 1980s [16]. However, the obvious way of doing this leads to one message delay more than that of Paxos Commit. The only algorithm that achieved the low message delay of Faster Paxos Commit is that of Guerraoui, Larrea, and Schiper [11]. It is essentially the same as Faster Paxos Commit in the absence of failures. (It can be modified with an optimization analogous to the sending of phase 2a messages only to a majority of acceptors to give it the same message complexity as Faster Paxos Commit.) This similarity to Paxos Commit is not surprising, since most asynchronous consensus algorithms (and most incomplete attempts at algorithms) are the same as Paxos in the failure-free case. However, their algorithm is more complicated than Paxos Commit. It uses a special procedure for the failure-free case and calls upon a modified version of an ordinary consensus algorithm, which adds an extra message delay in the event of failure. With 2F + 1 coordinators and N resource managers, Paxos Commit requires about 2FN more messages than Two-Phase Commit in the normal case. Both algorithms incur the same delay for writing to stable storage. In modern local area networks, messages are cheap, and the cost of writing to stable storage can be much larger than the cost of sending messages. So in many systems, the benefit of a non-blocking protocol should outweigh the additional cost of Paxos Commit. Paxos Commit implements transaction commit with the Paxos consensus algorithm. Some readers may find this paradoxical, since there are results in the distributed systems theory literature showing that transaction commit is a strictly harder problem than consensus [10]. However, those results are based on a stronger definition of transaction commit in which the transaction is required to commit if all RMs are nonfaulty and choose to prepare even in the face of unpredictable communication delays. In contrast, our Non-Triviality condition requires the transaction to commit only under the additional assumption that the entire network is non faulty meaning that all messages sent between the nodes are delivered within some known time limit. (Guerraoui, Larrea, and Schiper stated this condition more abstractly in terms of failure detectors.) The stronger definition of transaction commit is not implementable in typical transaction systems, where occasional long communication delays must be tolerated.

Salman Abdul Moiz et al. (2010), [18] they worked for any database environment either wired or wireless, if multiple host access similar data items it may lead to concurrent access anomalies. As disconnections and mobility are the common characteristics in mobile environment, preserving consistency in presence of concurrent access is a challenging issue. Most of the approaches use locking mechanisms to achieve concurrency control. This leads to increase in blocking and abort rate in mobile environments. However the dynamic timer adjustment strategies may use locking mechanism to efficiently implement concurrency control. To reduce deadlocks and blocking of resources an enhanced optimistic approach for concurrency control is proposed by Salman Abdul Moiz et al. (2010) [18]. To show the effectiveness of the commit protocols in mobile environments, a simulator is designed and implemented to demonstrate how the transactions are committed and how the data consistency is maintained when the transactions are executed concurrently. The simulator was tested for both pessimistic and optimistic approaches.

3. Methodology:

We have developed a mobile environment protocol that can handle the distributed database of several clients using priority based concurrency control mechanism with considerations of Hand Off situation. For this case we have developed our algorithm using MATLAB 2010.The main step of our algorithm is given as below:-

1. Request for data transaction.
2. Checking for nodes Cell Address for allotting Base Station.
3. Checking for the validity of the transaction.
4. Rejection of the node request which have not permission for data transaction.
5. Request acceptance and acknowledgement to the active valid nodes by the base station.
6. Priority listing of the nodes for data transaction.
7. Data transmission of higher priority nodes through the channel.
8. Queue allotment of the remaining nodes in priority listing for data transmission in next round.

Hence in summarize way our propose algorithm follows following rules for accepting the transaction request of the clients:

International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

If Pr < Ph and Pr abort in last session
Add Pr to the highest priority list
Else
Pr < Ph and Pr did not abort or request in last session
Abort Pr request
End

The mobile database network that we have considered here has following specifications:-

Table 1: Model Parameters and Their Baseline Values

Parameters	**Baseline Values**
System Level	
Number of MTSO	**1**
Number Of Cell Sites	**9**
Location Update Interval	**1sec**
Number Of Channels For Each Cell Site	**4/5**
Mobile Network	
Number Of Mobile Clients	**63**
Database	
Number Of Local Databases	**9**
Database Size	**200**
Concurrency Control	**yes**

We have considered 'C' clients in a network of area N*N Km Square. The nodes are scattered in 9 different cells randomly. Each cell has a one base station hence we have 9 base station located at the centre of each cells.
The nodes are dynamic and charging their position by dx and dy displacement where dx and dy varies in the range of -0.5 to + o.5 kms. in each round maximum.

The coordinates of C nodes are defined as Cx and Cy hence in every round the position is updated as Cx=Cx + dx and Cy=Cy + dy in each round. The distance between each node from every base station is updated and minimum distance is determined for all the nodes with respect to base station. The nearest station .The nearest station the table below shows the node id of all the nodes and respective x and y coordinates cx and cy and their distance from all he base stations. Using this table for every round the algorithm generates the cell address of all the nodes.

4. Result and Discussion:

We have taken two kinds of algorithm one having no priority queuing and for this case for 4 no. of channels we have shown. he results in terms of no. of dropouts after 50 rounds for every nodes. Our algorithm displace the status of all the nodes with respect to its communication with BTS and finally it also shows the nodes that are under goes through the transaction from each channel from ch1 to ch4.Some of the snapshot are shown below in figure 1.

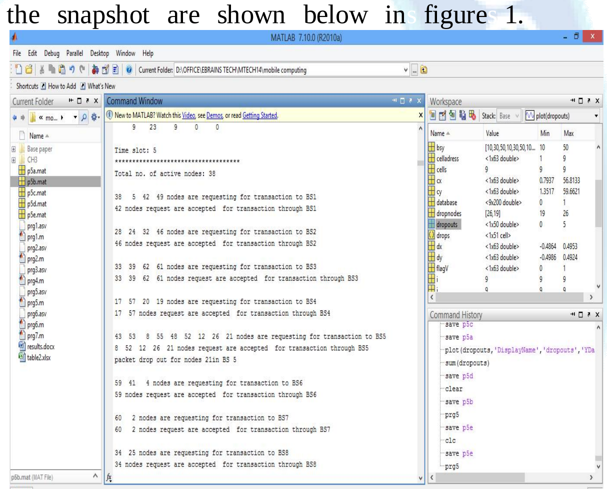

Fig 1 : Algorithm execution snapshot

Table 2. Data Transaction Report Without Priority Queue

Time slot: 1 ************************************ Total no. of mobile clients: 38 42 49 50 38 27 5 13 7 nodes are requesting for transaction to BS1 49 38 27 5 13 nodes request are accepted for transaction through BS1 packet drop out for nodes 13in BS 1	nodes are requesting for transaction to BS7 nodes request are accepted for transaction through BS7 25 34 nodes are requesting for transaction to BS8 25 nodes request are accepted for transaction through BS8 23 37 40 nodes are requesting for transaction to

International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

46 28 14 24 nodes are requesting for transaction to BS2
28 14 24 nodes request are accepted for transaction through BS2

39 61 35 62 nodes are requesting for transaction to BS3
39 61 62 nodes request are accepted for transaction through BS3

44 57 11 56 30 17 nodes are requesting for transaction to BS4
57 30 17 nodes request are accepted for transaction through BS4

52 53 26 21 36 43 48 8 nodes are requesting for transaction to BS5
53 36 8 nodes request are accepted for transaction through BS5

15 59 45 nodes are requesting for transaction to BS6
59 45 nodes request are accepted for transaction through BS6

BS9
37 nodes request are accepted for transaction through BS9

total drop outs in round 1 = 1

Nodes Transaction priorty queue:
B.S. No. Ch1 Ch2 Ch3 Ch4
1 49
38 27 5
2 28
14 24 0
3 39
61 62 0
4 57
30 17 0
5 53
36 8 0
6 59
45 0 0
7 0
0 0 0
8 25
0 0 0
9 37
0 0 0

Time slot: 2
****
Total no. of mobile clients: 48

7 50 5 38 27 49 nodes are requesting for transaction to BS1
7 38 27 49 nodes request are accepted for transaction through BS1

3 14 24 32 28 46 nodes are requesting for transaction to BS2
32 28 46 nodes request are accepted for transaction through BS2

33 35 39 61 62 nodes are requesting for transaction to BS3
33 35 39 61 62 nodes request are accepted for transaction through BS3
packet drop out for nodes 62in BS 3

60 2 51 31 58 nodes are requesting for transaction to BS7
2 51 58 nodes request are accepted for transaction through BS7
16 34 nodes are requesting for transaction to BS8
nodes request are accepted for transaction through BS8
9 10 37 nodes are requesting for transaction to BS9
9 37 nodes request are accepted for transaction through BS9
total drop outs in round 2 = 2
Nodes Transaction priorty queue:

International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

44 57 19 1 47 11 nodes are requesting for transaction to BS4
44 19 1 47 nodes request are accepted for transaction through BS4

52 55 8 26 53 12 21 36 48 18 43 nodes are requesting for transaction to BS5
8 53 12 21 48 nodes request are accepted for transaction through BS5
packet drop out for nodes 48in BS 5
45 15 41 59 nodes are requesting for transaction to BS6
45 41 nodes request are accepted for transaction through BS6

B.S. No. Ch1 Ch2 Ch3 Ch4
1 7
38 27 49
2 32
28 46 0
3 33
35 39 61
4 44
19 1 47
5 8
53 12 21
6 45
41 0 0
7 2
51 58 0
8 0
0 0 0
9 9
37 0 0

Time slot: 3
****
Total no. of active mobile clients : 27

27 29 50 nodes are requesting for transaction to BS1
27 50 nodes request are accepted for transaction through BS1

22 46 28 24 nodes are requesting for transaction to BS2
24 nodes request are accepted for transaction through BS2

33 nodes are requesting for transaction to BS3
nodes request are accepted for transaction through BS3

47 30 57 20 19 44 11 nodes are requesting for transaction to BS4
30 57 20 19 44 11 nodes request are accepted for transaction through BS4
packet drop out for nodes 44 11in BS 4

60 31 51 2 nodes are requesting for transaction to BS7
60 51 nodes request are accepted for transaction through BS7

25 nodes are requesting for transaction to BS8
25 nodes request are accepted for transaction through BS8

63 nodes are requesting for transaction to BS9
63 nodes request are accepted for transaction through BS9

total drop outs in round 3 = 2

Nodes Transaction priorty queue:
B.S. No. Ch1 Ch2 Ch3 Ch4
1 27
50 0 0
2 24
0 0 0
3 0
0 0 0

International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

12 48 52 26 55 nodes are requesting for transaction to BS5 26 55 nodes request are accepted for transaction through BS5 45 nodes are requesting for transaction to BS6 nodes request are accepted for transaction through BS6	4 30 57 20 19 5 26 55 0 0 6 0 0 0 0 7 60 51 0 0 8 25 0 0 0 9 63 0 0 0
Time slot: 4 ************************************ Total no. of active mobile clients : 34 13 42 50 49 7 5 nodes are requesting for transaction to BS1 42 50 49 7 nodes request are accepted for transaction through BS1 46 3 22 14 nodes are requesting for transaction to BS2 46 22 nodes request are accepted for transaction through BS2 62 61 nodes are requesting for transaction to BS3 nodes request are accepted for transaction through BS3 17 56 44 47 nodes are requesting for transaction to BS4 44 nodes request are accepted for transaction through BS4 8 55 52 26 36 nodes are requesting for transaction to BS5 8 52 26 36 nodes request are accepted for transaction through BS5 41 15 4 45 59 nodes are requesting for transaction to BS6 nodes request are accepted for transaction through BS6	nodes are requesting for transaction to BS7 nodes request are accepted for transaction through BS7 25 16 nodes are requesting for transaction to BS8 16 nodes request are accepted for transaction through BS8 37 23 40 9 63 54 nodes are requesting for transaction to BS9 23 9 nodes request are accepted for transaction through BS9 total drop outs in round 4 = 0 Nodes Transaction priorty queue: B.S. No. Ch1 Ch2 Ch3 Ch4 1 42 50 49 7 2 46 22 0 0 3 0 0 0 0 4 44 0 0 0 5 8 52 26 36 6 0 0 0

International Journal of Research and Development in Applied Science and Engineering (IJRDASE)
ISSN: 2454-6844

	0 7 0 0 0 0 8 16 0 0 0 9 23 9 0 0
Time slot: 5 ************************************ Total no. of active mobile clients : 38 38 5 42 49 nodes are requesting for transaction to BS1 42 nodes request are accepted for transaction through BS1 28 24 32 46 nodes are requesting for transaction to BS2 46 nodes request are accepted for transaction through BS2 33 39 62 61 nodes are requesting for transaction to BS3 33 39 62 61 nodes request are accepted for transaction through BS3 17 57 20 19 nodes are requesting for transaction to BS4 17 57 nodes request are accepted for transaction through BS4 43 53 8 55 48 52 12 26 21 nodes are requesting for transaction to BS5 8 52 12 26 21 nodes request are accepted for transaction through BS5 packet drop out for nodes 21in BS 5 59 41 4 nodes are requesting for transaction to BS6 59 nodes request are accepted for transaction through BS6	60 2 nodes are requesting for transaction to BS7 60 2 nodes request are accepted for transaction through BS7 34 25 nodes are requesting for transaction to BS8 34 nodes request are accepted for transaction through BS8 54 63 40 9 6 37 nodes are requesting for transaction to BS9 54 63 9 nodes request are accepted for transaction through BS9 total drop outs in round 5 = 1 Nodes Transaction priorty queue: B.S. No. Ch1 Ch2 Ch3 Ch4 1 42 0 0 0 2 46 0 0 0 3 33 39 62 61 4 17 57 0 0 5 8 52 12 26 6 59 0 0 0 7 60 2 0 0 8 34 0 0 0

	9		54
	63	9	0

In table 2 we have shown the report for data transaction for round 1 to 5. For example in table 1 first of all number of active nodes are shown as 38 out of 63 nodes. These nodes are sending request to their respective base station. For example nodes 42, 49, 50, 38, 27, 5, 13 and 7 belongs to BS1 and sending request to it for getting permission for data transaction. Thereafter BS1 checks the validity of each node and in the second line we can see that only request for 49, 38, 27, 5 and 13 are accepted. Whereas transaction Request of node 42, 50 and 7 are rejected. Hence there are only 5 nodes which are in the request queue. Since there are only 4 channels hence the transaction of node 13 has gone through are dropout and only transaction has occurred initial 4 nodes i.e.49, 38, 27 and 5. Similarly we can see that there is no dropout in BS2 to BS9. At the end of round we can see that the node transaction priority queue in a tabular form having first column indicates the base station no. and column 2 to 4 shows the node id which has gone through transaction from ch1 to ch4.

5. Conclusion:

In this thesis we have developed three different algorithms for providing an environment of mobile distributed network system. For fulfilling the objectives of reducing packet drop out during the data transaction process. In our distributed system a transaction is performed by using collection of several step which are collectively called as Resource Management executed by Base Station to establish a connection and data transmission. In between different mobile node and the MTSO unit the transmission is assumed to be completed when any node request is either ends up with transaction commit or transaction dropout.

References:

[1] Jensen, C. S. Research Challenges in Location-enabled M-services. In Proceedings of the 3rd International Conference on Mobile Data Management (MDM'02), 2002.

[2] Pitoura, Evaggelia, Samaras, George, "Data Management for Mobile Computing, Advances in Database Systems, Vol 10, 1998, pp-172.

[3] T. Imielinski and B. R. Badrinath, Wireless Mobile Computing: Challenges in Data Management, Communications of the ACM, 37(10), 1994, pp18-28.

[4] Smith Wayne Plourede: "Handling Concurrency Issues in .NET", http://www.15seconds.com/issue/030604.htm, Web retreieve on February 19, 2005).

[5] Salman Abdul Moiz, Dr. Lakshmi Rajamani, "Single Lock Manager Approach for achieving Concurrency in Mobile Environments", 14th IEEE International Conference on High Performance Computing (HiPC), 2007. Springer LNCS 4873, ISBN 978-3-540-77219-4, pp. 650-660, 2007.

[6] Salman Abdul Moiz, Dr. Lakshmi Rajamani, "Concurrency Control Strategy to Reduce frequent rollbacks in Mobile Environments", 2009 IEEE/IFIP International Symposium on Trusted Computing (TrustCom 2009), ISBN# 978-0-7695-3823-5, Vol 2. Pp. 709-714, 2009.

[7] Salman Abdul Moiz, Dr. Lakshmi Rajamani, "An Efficient Strategy for achieving Concurrency Control in Mobile Environments", 12th IEEE Asia Pacific Network Operations & Management (APNOMS) symposium, 2009. Springer LNCS 5787, ISBN# 978-3-642-04491-5, pp. 519-5222, 2009.

[8] Salman Abdul Moiz, Dr. Lakshmi Rajamani, "An Algorithmic approach for achieving concurrency in Mobile Environments", 1st National Conference on Computing for Nation Development, INDIACom 2007, ISBN #978-81-094526-0-1, ISSN # 0973-7529.

[9] Salman Abdul Moiz, Dr. Lakshmi Rajamani, "A Real Time Optimistic Strategy to achieve Concurrency control in Mobile Environments using on-demand multicasting", International Journal of Wireless & Mobile Networks (IJWMN), Vol.2, No.2, May 2010, pp. 172-185, ISSN #0975-3834 (online), ISSN #0975-4679 (print).

[10] Salman Abdul Moiz, Dr. Lakshmi Rajamani, Supriya N. Pal, " Design and Implementation of Pessimistic Commit Protocols in Mobile Environments", The First International Workshop on Database Systems (DMS-2010), Springer Verlag Berlin Hiedelberg, CNSA 2010, CCIS 89, pp. 603-612, 2010.

[11] J. Gray and A. Reuter. Transaction Processing: Concept and Techniques. Morgan Kaufmann (1993).

[12] L. Golab and M. T. Ozsu. Issues in data stream stead of the validating transaction, the conflicting transaction management. SIGMOD Rec., 32(2), 2003, pp 5-14.

[13] Jayant H., Ramesh G. Kriti.R, S. Seshadri, "Commit processing in Distributed Real-Time Database Systems", Tech. Report-TR-96-01, Pro. Pro. Of 17th IEEE Real-Time Systems Symposium, USA,1996.

[14] Jan Lindstrom, "Relaxed Correctness for Firm Real-Time Databases," rtcsa,pp.82-86, 12th IEEE International Conf erence on Embedded and Real-Time Comp uting Systems and Applications (RTCSA '06)., 2006

[15] L. Sha, R. Rajkumar, S.H. Son and C.H. Chang. A Real-time Locking Protocol. IEEE Transactions on Computers, 40(7):793-800 (1991).

[16] kam-yiu lam et. al., "Concurrency Control In Mobile Distributed Real-Time Database Systems", Information Systems Vol. 25 No. 4, pp. 261–286, 2000.

[17] Jim Gray and Leslie Lamport, "Consensus on Transaction Commit" 1 January 2004 revised 19 April 2004, 8 September 2005.