Volume: 02, Issue: 06, Pages: 927-932 (2011)

Disconnected Transaction Management Scheme for Agent Based Mobile Networks

J.L. Walter Jeyakumar
Department of Computer Science and Engineering, Manonmaniam Sundaranar University,
Tirunelveli, Tamilnadu, INDIA
Email: walterjeya@hotmail.com
R.S. Rajesh
Department of Computer Science and Engineering, Manonmaniam Sundaranar University,
Tirunelveli, Tamilnadu, INDIA
Email: rs_rajesh1@yahoo.co.in

----------**ABSTRACT**----------

In this paper, we define a disconnected transaction management scheme for agent based mobile networks. This scheme proposes an environment in which mobile users can have simultaneous access to data by using the cache stored in the Fixed Agent. Data Access Manager module at the Fixed Agent controls the concurrency using invalidation technique. This approach supports disconnected transaction execution by allowing a disconnected mobile host to transfer transaction execution to the Data Access Manager in the fixed agent. The proposed transaction management scheme has been simulated in Java and performance of the scheme is presented.

Keywords – Transaction, Concurrency Control, Mobile Host, Fixed Agent, Cache invalidation

Date of Submission: 26 August 2010 Revised: 01 December 2010 Accepted: 24 April 2011

1. Introduction

Mobile computing environment consists of Fixed Hosts, Mobile Hosts and Base stations or Mobile Support Stations (MSS). MH is connected to the Fixed network through MSS via wireless channels. The Geographical area covered by a MSS is called a cell. Mobile Hosts are portable computers which move around in a cell. When a MH enters into a new cell hand-off or hand-over takes place. MH communicates only with the MSS responsible for its cell. Transactions and data management functions are done using the data base servers installed at MSS. Frequent disconnections, mobility, limited battery power and resources pose new challenges to mobile computing environment. Frequent aborts due to disconnection should be minimized in mobile transactions. Correctness of transactions executed on both fixed and mobile hosts must be ensured by the operations on shared data. Blocking of mobile transactions due to long disconnection periods should be minimized to reduce communication cost and to increase concurrency. After disconnection , mobile host should be able to process transactions and commit locally. Mobile computing provides the possibility of concurrent access of data by mobile hosts which may result in data inconsistency. Concurrency control methods have been used to control concurrency. Due to limitations and restrictions of wireless communication channels, it is difficult to ensure consistency of data.

In this paper, we present an agent based scheme for mobile transaction. Frequently accessed data are cached in the Fixed agent situated in the fixed wired network. Whenever an MH enters into a Fixed agent area it can connect and access the data in the cache. But upon update request by a MH, updation is done at the local cache and invalidation report is sent to all the mobile hosts which have already accessed the same data. This will force the mobile hosts to refresh their data values. Data Access Manager at the fixed agent is responsible for concurrency control and data invalidation. This framework also takes into account transaction update during disconnection.

The remaining part of this paper is organized as follows. Section 2 summarizes the related research. Section 3 focuses on the agent based architecture. Section 4 specifies the proposed framework for disconnected mobile computing. Section 5 gives the performance analysis and in Section 6 and Section 7 discussion and conclusion are presented.

2. Related work

When simultaneous access to data is made at the server, concurrency control techniques are employed to avoid data inconsistency. Conventional locking based concurrency control methods like centralized Two Phase locking and distributed Two Phase locking are not suitable for mobile environment. The system overhead that arises due to concurrency control mechanism can create a serious performance problem because of low capacity and limited resources in mobile environment [1]. More over, it makes mobile hosts to communicate with the server continuously to obtain and manage locks [2].

In Timestamp approach, the execution order of concurrent transactions is defined before they begin their execution. The execution order is established by associating a unique timestamp to every transaction. When two transactions conflict over a data item, their timestamps are used to enforce serialization by rolling back one of the conflicting transactions [3]. In optimistic concurrency control with dynamic time stamp adjustment protocol, client side write operations are required. But it may never be executed due to delay in execution of a transaction[4]. In multi version transaction model [5], data is made available as soon as a transaction commits at a mobile host and another transaction can share this data. But data may be locked for a longer time at a mobile host before the lock is released at the database server.

In [6], a transaction model for supporting mobile collaborative works was proposed. This model makes use of Export-Import repository which is a mobile sharing work space for sharing data states and data status. But in the Export-Import repository based model, locking is the main technique which has the following disadvantages. (i) More bandwidth is needed for request and reply since the locking and unlocking requests have to be sent to the server. (ii) Disconnection of mobile host or a transaction failure will result in blocking of other transactions for a long period.

In [7] , AVI (Absolute Validity Interval) was introduced for enforcing concurrency control without locking. AVI is the valid life span of a data item. But it calculates AVI only based on previous update interval. In [8], a method based on PLP(Predicted Life Period), which takes care of the dynamicity of the life time of data was proposed. Here, life span of data is predicted based on the probability of updation of data item. This method makes PLP of data item very close to the actual valid life span of a data item. But this approach did not take into account the disconnection issue. Hence in this paper, we have taken the issue of disconnection of mobile nodes and modified the algorithm discussed in [8] and proposed a new model. The proposed approach is better than this framework since it transfers transaction execution to the agent in the Fixed network when a Mobile Host is disconnected.

3. Agent Based Architecture

The proposed Agent based architecture model is illustrated in Fig 1. This model is a modified version of the model discussed in [8]. In this model we introduce Fixed Agents in which frequently accessed data are stored in a cache. This proposed model gives provision for disconnected transaction execution which was not supported by the existing model. The proposed model consists of Server, Fixed Agents and Mobile Hosts. The server can be directly connected to a mobile host. Fixed Agents are connected to the server through wired network. Fixed Agent has a communication range and any mobile host that enters into the agent area can connect to the server through the agent.

In Fixed Agents, cache is used to store the data. Mobile hosts are allowed to access data from the cache. When data request is made for the first time, data is retrieved from the server and stored in the cache. Subsequent requests are handled by the Data Access Manager module itself. When a mobile host requests for data update, after local updation of the data item, invalidation report is sent to all the mobile hosts that have already accessed the same data. This makes all the mobile hosts to refresh their data values. When a mobile host is disconnected from the Fixed agent after updation request, the updation task is transferred to the Data Access Manager in the Fixed Agent. Data Access Manager module is used to coordinate the operations in the cache.

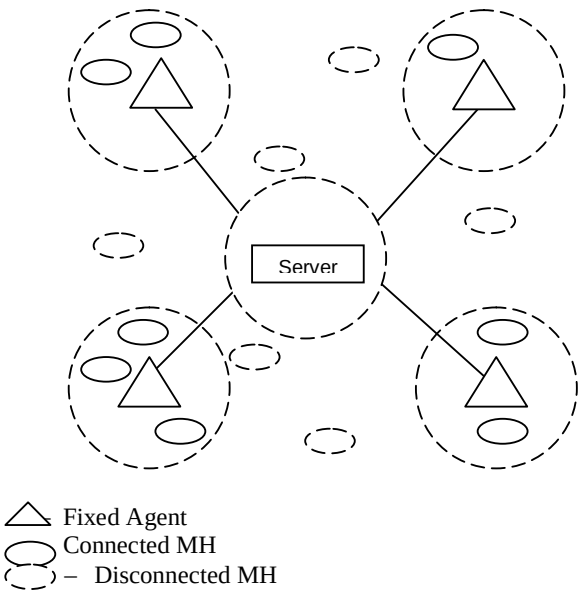

Fig.1 Agent Based Architecture model

4. Mobile Transaction Framework

Mobile hosts can directly connect to the server. But simultaneous access to data at the server will increase the system overhead. To overcome this disadvantage, data are cached in the Fixed Agent. The Data Access Manager at the Fixed Agent is responsible for enforcing concurrency and cache invalidation. Fig 2 and 3 illustrate the steps involved in MH, server and DAM algorithms.

4.1 Concurrency Control Mechanism

When more number of mobile hosts are accessing data simultaneously the problem of data inconsistency arises. This problem can be solved if we use an efficient concurrency control mechanism. When data request is made for the first time, data is retrieved from the server and stored in the cache. Future requests for data are managed directly by the Data Access Manager.

Data Access Manager uses a suitable data item format [8] to store data in the cache. It has (id, TLU, PLP, dataval, NT) where id denotes unique Id of the data item, TLU indicates time of Last Update, PLP is Predicted Life Period, dataval is current value of the data item and NT denotes number of transactions that concurrently access the data item.

When Data Access manager fetches data for the first time from the server, it sets TLU to current time, PLP to optimal time based on the nature of data item and NT to 1. NT is incremented whenever a new data access request is made. Data in the cache becomes invalid, once it is updated in the server. Life span of a data item is predicted using PLP. It makes use of the probability of updation as a basis for setting valid life span of a data item. In PLP interval, data item is valid and all the mobile hosts can access same data item concurrently. When a MH makes update request or PLP expires, the data item is invalidated. Now PLP is modified and invalidation report is sent. The predicted life period of data item is computed using the formula given in [8]. PLP=PPLP ± (p*PPLP) Where PPLP is Previous Predicted Life Period and p is predicted probability of updation of data item. p = Total_updates / NT. It is the ratio of data item update to data item access. Since predicted probability of updation is based on recent past history of updation rate, it is highly probable that PLP is very close to the actual validity interval of the data item.

4.2 Transaction Execution in the MH

After connecting to the server, MH gets a copy of a data item from the Data Access Manager using read request. If the MH wants to update data, it checks, if the MH is about to be disconnected. If so, the updation task is assigned to the DAM before disconnection. Otherwise update request is sent to DAM.

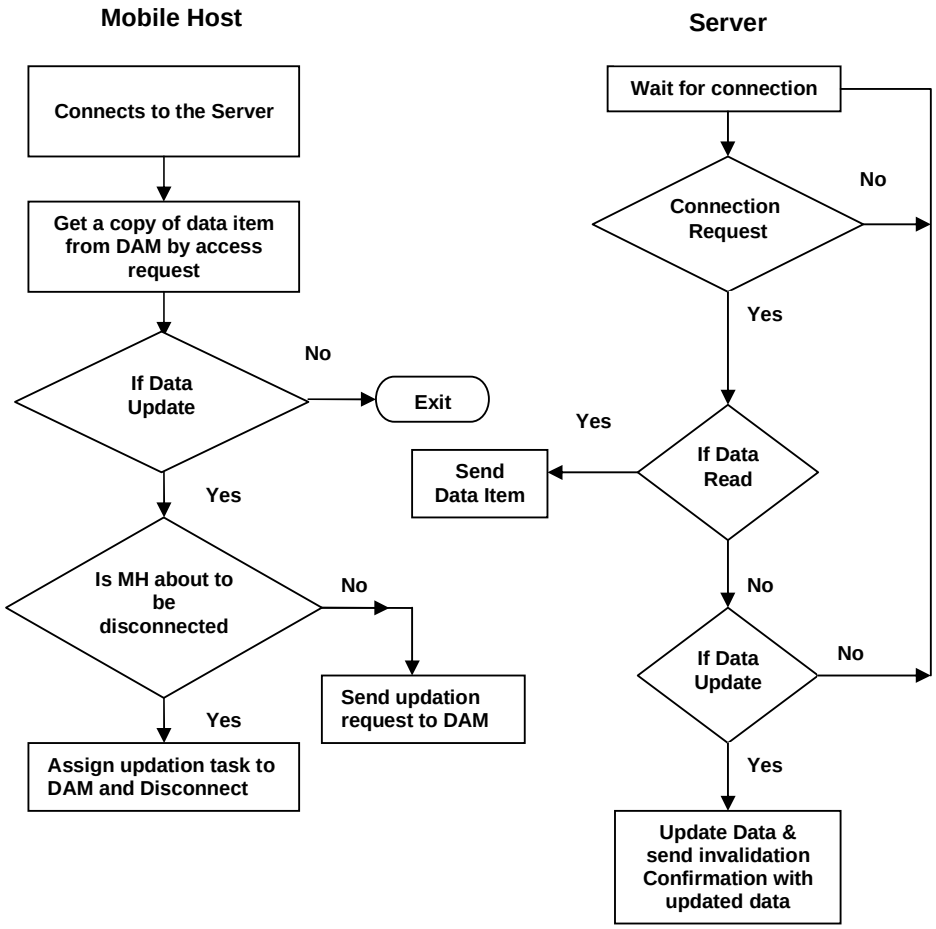

Fig. 2 Transaction Flow Diagram for Mobile Host and Server

4.3 Function of Data Access Manager and Server

When MH makes a read request, if it is in the cache of the Fixed Agent, NT is incremented by one. Otherwise, data is fetched from the server and data item format is initialized. Then data is sent to MH.

When MH makes an update request, DAM updates data locally and invalidation report is sent to all the mobile hosts that have already accessed the same data item. This forces all the transactions to refresh their data values. This update request is now forwarded to the server. The server updates the data and sends invalidation confirmation along with the updated value. Once Data Access Manager receives the confirmation, it updates the data in the cache. The data in the cache is invalidated if updation is made in the server or PLP expires.

If transaction update is made by the Data Access Manager for the disconnected MH, the above procedure is followed except that at the end, DAM generates updation report and forwards it to the MH when it gets reconnected.

5. Performance Analysis

The model discussed in section 3 has been simulated using Java (JDK 1.6) in a Pentium Dual Core processor based System @ 2.4 GHz with 3 GB RAM. A mobile network is simulated with 25 nodes and 4 Fixed Agents at an area of 1 km X 1 km radius. Mobile nodes move using random walk mobility model [9]. The response time for different number of transactions are evaluated for the existing and proposed schemes for non disconnected transaction case. As the existing model does not support disconnected operations, only the proposed model is taken for disconnected case. The behaviors of the models for the two cases are presented in Figures 4 and 5. Response time is calculated as the time taken to service the request host.

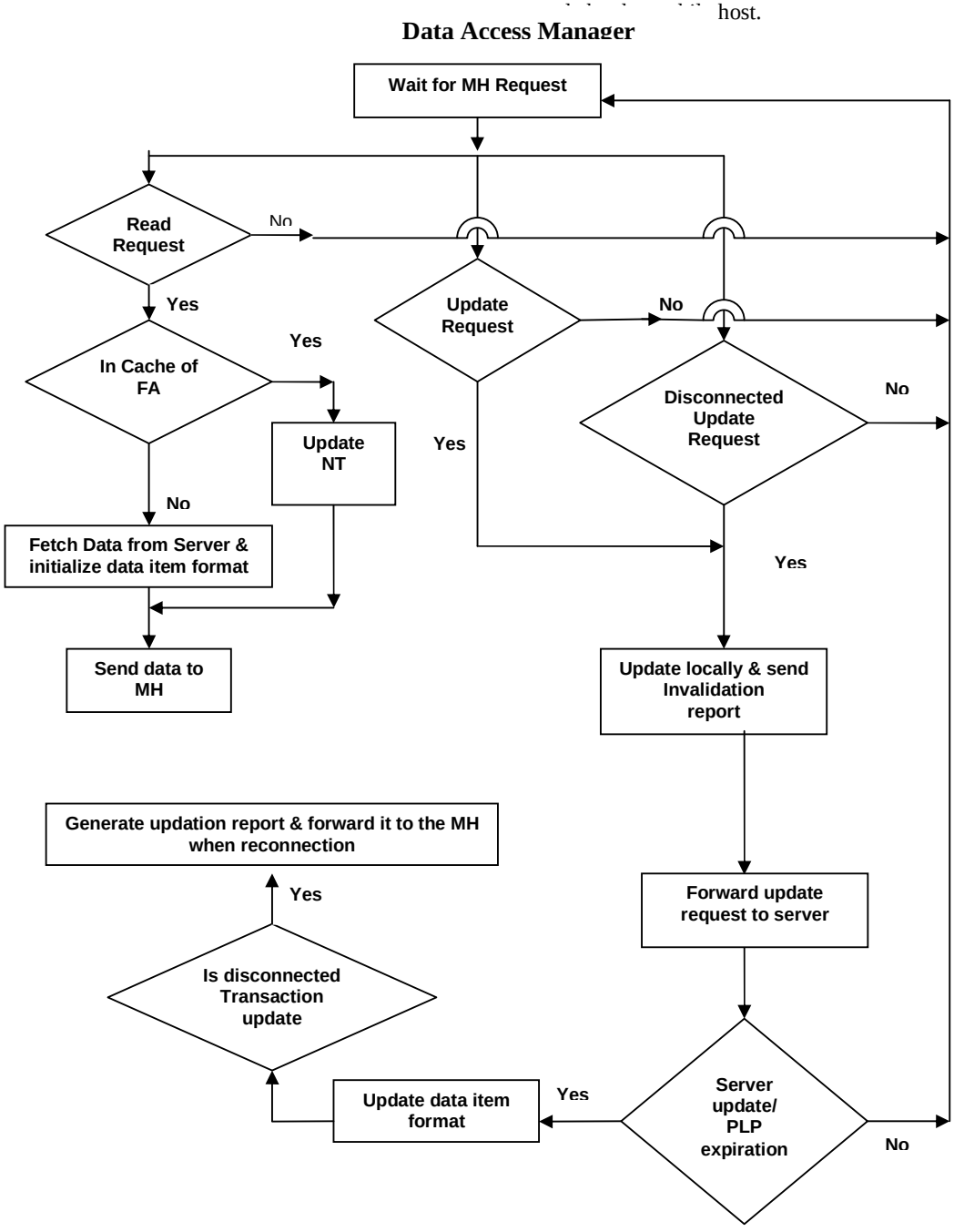

Fig. 3 Transaction Flow diagram for Data Access Manager

6. Discussion

Fig 4 presents the response time for different number of transactions for non-disconnected case. The performance of the proposed scheme matches more or less with the existing scheme. But the proposed scheme suffers from slight increase in response time (5%) than the existing scheme for number of transactions more than 20. This is due to the presence of agent delay.

However the disconnected case in Fig 5 shows that disconnected support is provided by our model up to 15 transactions. But response time increases steeply when number of transactions exceeds 15. This is due to the overhead associated with more number of updation tasks that are transferred to Data Access Manager in the Fixed Agent, when the number of transactions increases. Also from Fig 4 and 5, we understand that transactions without disconnections take less response time compared to transactions with disconnections. This is due to the extra time involved in handling disconnected operations.

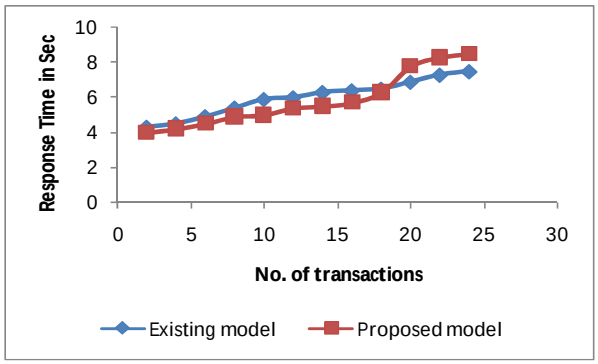

Fig. 4. Analysis of Response time for Transactions without disconnection

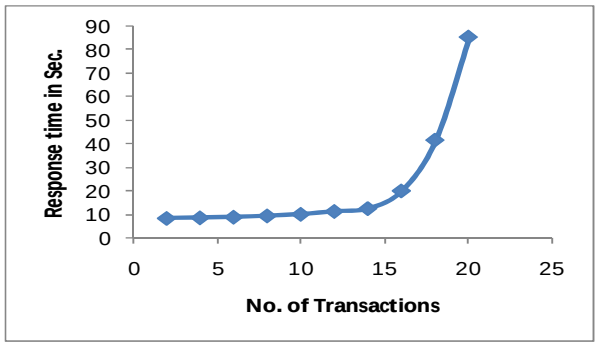

Fig. 5. Analysis of Response time for Transactions with Disconnection

7. Conclusion

In this paper, we have proposed a transaction framework for disconnected mobile computing environment. We make use of Fixed Agents in the wired network to store the cache. This cached data can be accessed by the mobile hosts when they get connected. By using a Fixed Agent and concurrency control with out locking for accessing data, we claim that message communication costs and database update costs are minimized to a larger extent. When mobile hosts are disconnected, transaction update task can be transferred to the Fixed Agent.

REFERENCES

[1] Vijay Kumar, *Mobile Database Systems* (Wiley Interscience, 2006).

[2] Victor C.S., Kwok wa Lam and Son, S.H., Concurrency Control Using Timestamp Ordering in Broadcast Environments, *The Computer Journal, Vol.45* No.4, 2002,
410-422.

[3] P.A Bernstein, V. Hadzilacos and N. Goodman, *Concurrency control and Recovery in Database Systems* (Addison Wesley, 1987).

[4] Ho-Jin Choi, Byeong-Soo Jeong, A Timestamp Based Optimistic Concurrency Control for Handling Mobile Transactions, *ICCSA 2006, LNCS 3981*, 2006, 796-805.

[5] Madria, S. K., M. Baseer, and S. S. Bhowmick, A Multiversion Transaction Model to Improve Data Availability in Mobile Computing, *CoopIS/DOA/ODBASE*, 2002, 322-338.

[6] Le, H. N., and M. Nygård, A transaction model for Supporting mobile Collaborative Works, *IEEE*, 2007, 347-355.

[7] Salman Abdul Moiz, Mohammed Khaja Nizamuddin, Concurrency Control without Locking in Mobile Environments, *IEEE, 2008*, 1336-1339.

[8] Miraclin Joyce Pamila J.C and Thanuskodi K, Framework for transaction management in mobile computing environment, *ICGST-CNIR Journal*, 2009, 19-24.

[9] T. Camp, J. Boleng, and V. Davies, A survey of mobility models for ad hoc network research, *Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, vol. 2, no. 5, 2002, 483-502.

Authors Biography

Mr J.L. Walter Jeyakumar has obtained his M.C.A from Bharathidasan University in 1988 and completed M.Phil from Manonmaniam Sundaranar University in 2007. He is currently working as Associate Professor at St. Xavier's college, Palayamkottai, Tirunelveli. He has 22 years of teaching experience. His areas of interests are Networks and Mobile Computing.

Dr. R. S. Rajesh received his B.E and M.E degrees in Electronics and Communication Engineering from Madurai Kamaraj University, Madurai, India in the year 1988 and 1989 respectively, and completed his Ph.D in Computer Science and Engineering from Manonmaniam Sundaranar University in the year 2004. In September 1992 he joined in Manonmaniam Sundaranar University where he is currently working as Associate Professor in the Computer Science and Engineering Department. He got more than 19 years of PG Teaching and Research experience. His current research interests include Digital Image Processing, Wireless Networks, Pervasive Computing and Parallel Computing.