Performance Analysis of a Connection Fault-Tolerant Model for Distributed Transaction Processing in Mobile Computing Environment

Tome Dimovski and Pece Mitrevski

St. Clement Ohridski University, Faculty of Technical Sciences, Ivo Lola Ribar bb
7000 Bitola, Republic of Macedonia
{tome.dimovski,pece.mitrevski}@uklo.edu.mk

Abstract. Mobile embedded systems increasingly use transactions for applications like mobile inventory, mobile commerce or commercial applications. Yet, many issues are challenging and need to be resolved before enabling mobile devices to take part in distributed computing. Mobile environment limitations make it harder to design appropriate and efficient commit protocols. There are a handful of protocols for transaction execution in distributed mobile environment, but almost all consider a limited number of communication models. In this paper, we evaluate the performance of a Connection Fault-Tolerant Model, comparing the results in several deferent scenarios, as well as its contribution to the overall mobile transaction commit rate. Our simulation-based performance analysis determines the impact of (i) ad-hoc communication between mobile hosts and (ii) the implementation of a decision algorithm, to the mobile transaction commit rate. We also determine the connection timeout values that contribute the most for a high ad-hoc communication impact on mobile transaction commit rate.

Keywords: Distributed mobile transactions, ad-hoc communication, Decision Algorithm.

1 Introduction

The increasing emergence of mobile devices contributes to rapid progress in wireless technologies. Mobile devices interacting with fixed devices can support applications such as e-mail, mobile commerce (m-commerce), mobile inventory, etc. But there are many issues that are challenging and need to be resolved before enabling mobile devices to take part in distributed computing. For distributed systems, a transaction is a set of operations that fulfill the following condition: either all operations are permanently performed, or none of them are visible to other operations (known as the *atomicity* property). In the execution of transactions the key issue is the protocol that ensures atomicity.

The mobile environment is comprised of mobile devices with limited resources like processing, storage, energy capacity and continuously varying properties of the

S. Markovski, M. Gusev (Editors): ICT Innovations 2012, Web Proceedings, ISSN 1857-7288

wireless channel. Wireless communication induces much lower bandwidth, higher latency, error rates and much higher costs. This increases the time needed for mobile hosts to execute transactions and can even lead to execution failure. *Mobile hosts (MHs)* are highly vulnerable devices because they are easily damaged or lost. MHs naturally show frequent and random network disconnections. These limitations and characteristics of the mobile environment make it harder to design appropriate and efficient commit protocols. A protocol that aborts the transaction, each time the MH disconnects from the network, is not suitable for mobile environments because it is part of the normal mode of operation. In other words, disconnections need to be tolerated by the protocol.

The *two-phase commit (2PC)* protocol [1] that allows the involved parties to agree on a common decision to commit or abort the transaction even in the presence of failures is the most commonly used protocol for fixed networks but is unsuitable for mobile environments [2]. There are several other protocols for transaction execution in distributed mobile environment [3-13], but almost all consider limited number of communication models. A Connection Fault-Tolerant Model (CFT) that shows resilience to connection failures of the mobile devices has been presented in [14]. It differs from other infrastructure based protocols [3-5], [7], in the fact that beside the standard communication between mobile hosts and the fixed network, it supports (i) ad-hoc communication between mobile hosts and (ii) introduces a decision algorithm that is responsible for decision making on behalf of a mobile host in a special case, when neither standard, nor ad-hoc communication is possible.

The main contribution of this paper is the simulation-based performance analysis of the CFT model for mobile distributed transaction processing. We evaluate the performance of the CFT model, comparing the results in several deferent scenarios, as well as its contribution to the overall mobile transaction commit rate. We also determine the optimum connection timeout values that contribute the most for the high ad-hoc communication impact on transaction commit rate.

The paper is organized as follows. Section 2 gives a survey of related work. In Section 3 we present the model of the mobile environment, and in Section 4 the transaction model. Description of the Connection Fault-Tolerant Model is given in Section 5, whereas in Section 6 we present simulation results and their analysis. Section 7 discusses conclusions.

2 Related work

Transaction Commit on Timeout (TCOT) protocol [3] is based on a timeout approach for Mobile Database Systems, which can be universally used to reach a final transaction termination decision in any message oriented system. This protocol limits the amount of communication between the participants in the execution of the protocol. It decreases the number of wireless messages during execution, and does not consider mobile hosts as active participants in the execution of transactions.

The basic idea of the *Two-Phase Commit Protocol for Mobile Wireless Environment (M-2PC)* [4] is to adapt the 2PC protocol for mobile systems with distributed transactions. Mobile hosts are active participants in the execution of a transaction and they send confirmation (that the work is done) to the agent or to the

fixed device, in order to save energy. This model requires simultaneous connection of all mobile participants at the beginning of a transaction. This protocol does not provide adequate management of mobility and failures caused by network disconnection, nor does it provide a mechanism to control the competitiveness of distributed transactions.

Fault-Tolerant Pre-Phase Transaction Commit (FT-PPTC) protocol [5] provides mechanisms for dealing with disturbances in the system in mobile environment. The protocol supports heterogeneous mobile databases. FT-PPTC implements distributed transaction in two phases: pre-phase, one which is covering the mobile hosts and the main phase which refers to the fixed part of the network. Mobile hosts are active participants in the execution of a transaction. No mechanisms are developed for competition in mobile distributed transactions. FT-PPTC does not provide adequate management of mobility, as well, because when mobile hosts are disconnected from the fixed network for a long time, they can block resources on the fixed participants. This, in turn, leads to an increased number of mobile transaction aborts.

In the *concurrency control without locking* approach [6], a concurrency control mechanism in mobile environment is proposed, by introducing the concept of *Absolute Validity Interval (AVI)* which is a time period in which the data item is said to be valid. It does not use the traditional locking mechanism. The new mechanism provides reading same available data item by multiple mobile hosts. If the data item is updated by any mobile host, the mobile hosts which have already read the same value must be invalidated. It reduces the waiting time for execution of a transaction and resources are not unnecessary locked.

3 Model of the mobile environment

In this paper we consider a system model for the mobile distributed environment consisting of a set of *mobile hosts (MHs)* and a set of *fixed hosts (FHs),* presented in Fig. 1. The model has two main parts: the fixed part of the network and a mobile part of the network. Communication between the two is conveyed via *Mobile Support Stations (MSS)*, which are connected to the fixed part of the network via wired links. MHs can cross the border between two different geographical areas covered by different MSSs.

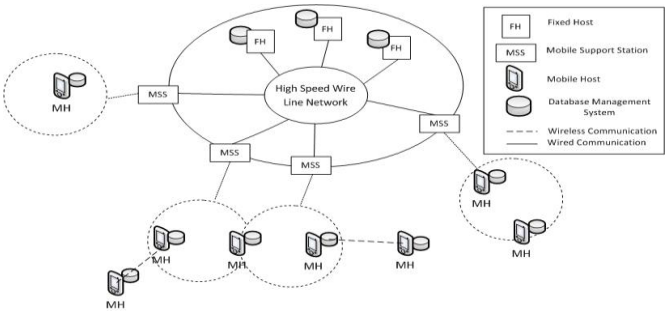

Fig. 1. Communication model in the mobile environment

In the considered system model, first, MHs can communicate with the FHs through a MSS via wireless channels only when they are located within the MSS coverage area. Second, the MHs can ad-hoc communicate with neighboring MHs via wireless channels. When MHs enter a geographical area that is out of coverage of any MSS, in order to access database servers in the fixed network, they may connect through a neighboring MH which is in the covering area of any MSS.

In brief, we consider a mobile distributed environment where (i) MHs can communicate with each other (and/or with FHs) through MSS and, in addition, (ii) MHs can ad-hoc communicate with neighboring MHs in order to reach the fixed part of the network. We assume that database servers are installed on each FH, and each MH has a mobile database server installed.

4 Transaction model

A distributed transaction where at least one MH participates is called a *Mobile Transaction (MT).* We identify a MH where a transaction is issued as a *Home-MH (H-MH).* Participating MHs and FHs in the execution of a mobile transaction are called *participant MHs (Part-MH)* and *participant FHs (Part-FH).*

We assume the existence of a *Coordinator (CO)* which is responsible for coordinating the execution of the corresponding transaction. The CO is responsible for storing information concerning the state of the transaction execution. Based on the information collected from the participants of the transaction, the CO takes the decision to commit or abort the transaction and informs all the participants about its decision. The CO should be executed on a fixed host or hosts. This means that logs will be kept more safely.

5 Connection Fault-Tolerant Model

5.1 Overview

Some of the most frequent failures in mobile environments are communication failures. When MHs are in motion, they may exit the geographical area that is covered by some MSS and the resources of the fixed participants may potentially be blocked for an undefined period of time. If MHs do not reestablish connection with any MSS the transaction is aborted.

To minimize the number of mobile transaction aborts by tolerating failures caused by network disconnections and reduce the resource blocking times of fixed participants, we propose a CFT model for distributed transaction processing in mobile computing environment. The CFT model ensures the atomicity property.

The CFT model supports two communication protocols and a decision algorithm, as well:

1. The first protocol is a *Standard communication protocol* when MHs can directly connect to the fixed part of the network through MSSs.

2. The second protocol is an *Ad-hoc communication protocol* when MHs cannot directly connect to the fixed part of the network through any MSS. With this protocol, MHs can ad-hoc communicate with neighboring MHs in order to reach the fixed part of the network.

In the Standard communication protocol, similar as in [15], to minimize the use of the wireless communication and conservation of the resources of MHs, to each MH we assign a *Mobile Host Agent (MH-Ag)* that we add to the fixed network. We assume that in the execution of a transaction MH-Ag is representing the MH in the fixed network and it acts as an intermediator between the MH and the transaction CO. All communications between MH and CO go through the MH-Ag. The MH-Ag is responsible for storing all the information related to the states of all MTs involving the MH. In the fixed network, a server or servers can be designated, where MH-Ag is created for each participating MH.

The second (Ad-hoc) communication protocol is when MHs cannot directly connect to the fixed network, or MH-Ag cannot directly communicate with its MH through any MSS. In that case, they try to connect via ad-hoc communication with any neighboring MH which is in the covering area of any MSS. To allow this, we assign a MH-*Relay Agent (MH-RAg)* to each a MH. The *MH-RAg* is responsible for ensuring relay wireless link between neighboring MHs. This means that MH which is out of the coverage area can connect to its MH-Ag in the fixed network via *MH-RAg* of the neighboring MH which is in coverage area of any MSS.

In the CFT model we define an additional function of a MH-Ag that we call a *Decision Algorithm (DAlg)*. DAlg is used during the execution of a transaction when MH-Ag cannot directly or ad-hoc communicates with its MH for a defined period of time. DAlg's task is to check if *Transaction Processing Fragment (TPF)* function is **WRITE** (insert/update/delete) or **READ**. If TPF function is **WRITE,** DAlg saves the TPF in a FIFO (First-In First-Out) queue list and makes a decision for the MH to send "Yes" vote to the transaction CO. When the connection between MH and the corresponding MH-Ag is reestablished, MH-Ag's first task is to send all saved TPFs to the corresponding MH. If TPF function is **READ**, DAlg will wait for connection reestablishment between MH and the corresponding MH-Ag, for a defined period of time. If the connection is not reestablished in the specified time period, DAlg makes a decision for the MH to send "No" vote to the transaction coordinator.

5.2 Connection Fault-Tolerant Model operation

In this section we make a short review of the operation of the CFT model. Fig. 2 illustrates the execution of a mobile transaction for the proposed model.

If H-MH is connected to the fixed network through some MSS, it initiates a mobile transaction by sending TPF to the transaction CO through its corresponding MH-Ag which acts as an intermediator between CO and MH.

Transaction CO computes the *Execution timeout (Et)*, which is a time limit for all participants to complete the execution of the TPFs and send a vote to CO. After that, CO sends Et and TPFs to all Part-FHs and MH-Ags that represent the Part-MHs in the fixed network, asks them to PREPARE to commit the transaction, and enters the wait state. Subsequently, every MH-Ag initiates *Connection timeout (Ct)*, which is a time

limit for MH-Ag to establish connection with its MH, and try to send Et and TPFs to its MH through standard or ad-hoc communication protocol. If MH-Ag cannot establish connection with its MH before Ct expires, it activates the DAlg.

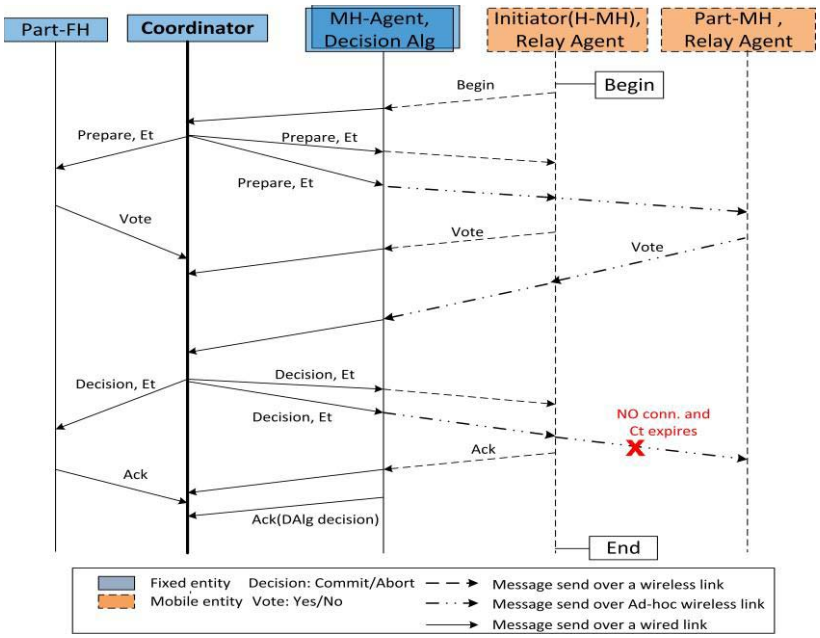

Fig. 2. Execution of a mobile transaction in mobile environment

DAlg checks if the *TPF* function is **WRITE** or **READ**. In the case of **WRITE,** DAlg saves the TPF in a FIFO queue list and makes a decision for the MH to send "Yes" VOTE to the transaction CO. As already mentioned before, if the TPF function is **READ**, DAlg makes a decision for the MH to send "No" VOTE to the transaction CO. If MH-Ag establishes connection with its MH before Ct expires, it sends TPF to its MH and resets Ct.

When the participants receive the PREPARE message, they check if they could commit the transaction. If so, and if MH establishes connection with MH-Ag before Ct expires, MH sends "Yes" VOTE to CO through its corresponding MH-Ag via standard or ad-hoc communication protocol. If Ct expires, MH-Ag activates DAlg.

After the CO has received VOTE from every participant, it decides whether to COMMIT or ABORT the transaction. If, for any reason, even one of the participants votes "No" or Et expires, the CO decides to ABORT the transaction and sends "Abort" message to all participants. Otherwise, if all the received votes are "Yes" and Et is not expired, the CO decides to COMMIT the transaction and sends "Commit" message, with reset Et to all participants. The participants need to ACKNOWLEDGE the CO's decision before the reset Et expires.

6 Simulation results and analysis

In our simulation-based performance analysis of the Connection Fault-Tolerant Model, we focus on mobile transaction commit rate performance metric. For the simulation analysis, we used SimPy [16], a process-based discrete-event simulation package based on standard Python programming language [17]. A simulation run is set to simulate 10 hours. Transactions are generated with exponentially distributed interarrival times, with an average of 30 seconds. We assume that all transactions are of similar length, but experience different connection conditions. The number and the nature of Part-MHs and Part-FHs are randomly selected in order to model arbitrary heterogeneity. Table 1 summarizes our simulation parameters.

Table 1. Simulation settings

Parameter	Value
Number of Part-MHs	3-5
Fragment execution time (Part-MH)	[0.3-0.7]s
Fragment execution time (Part-FH)	[0.1-0.3]s
Transmission delay (wireless link)	[0.2-1.0]s
Transmission delay (wireless ad-hoc link)	[0.4-2.0]s
Transmission delay (wired link)	[0.01-0.03]s
Disconnection Rate	[0 – 95]%
Ad-hoc support	[10 – 90]%
Distributed transaction WRITE function	[10 – 90]%

Hence, disconnection rate is defined as the ratio of the time when the participating MH is disconnected from the fixed network, against the total simulation time. Ad-hoc support is the ratio of the time when ad-hoc communication is available between MHs, against the total simulation time. It is hard to quantify the level of ad-hoc support between MHs in mobile distributed environment. In some parts of the wireless network ad-hoc support can be lower compared to other. For that reason, in our simulation we define three groups that represent different parts of the wireless network with different levels of ad-hoc support (there is a fundamental relationship between node density and delay in wireless ad-hoc networks with unreliable links). Every MH in the wireless network is a member of one of the defined groups.

As we mentioned before in Section 5, the CFT model supports two communication protocols – *Standard* and *Ad-hoc*, and introduces *Decision Algorithm* in order to minimize the number of mobile transaction aborts by tolerating failures caused by network disconnections. Considering several different scenarios with- or without support of the DAlg, our simulation-based performance analysis determines the CFT model contribution to the overall mobile transaction commit rate.

For the maximum ad-hoc communication contribution to the mobile transaction commit rate in the CFT model, we need to determine the values of the Connection timeout, which will allow ad-hoc communication before activating the DAlg. Fig. 3 shows the mobile transaction commit rate against different connection timeouts and different ad-hoc support values. Execution timeout is set to UNLIMITED. We assume that the functions of all mobile transactions are READ, which means that if DALg activates, the transaction will be aborted. From the chart in Fig. 3 can be concluded

that for any level of ad-hoc support, commit rate increment is more evident for increments of connection timeout up to 2.4*s*. After that point, the commit rate only slightly increases when the connection timeout time rises. It means that connection timeout value of 2.4*s* is the *optimum value* for maximum ad-hoc communication contribution on mobile transaction commit rate in the CFT model.

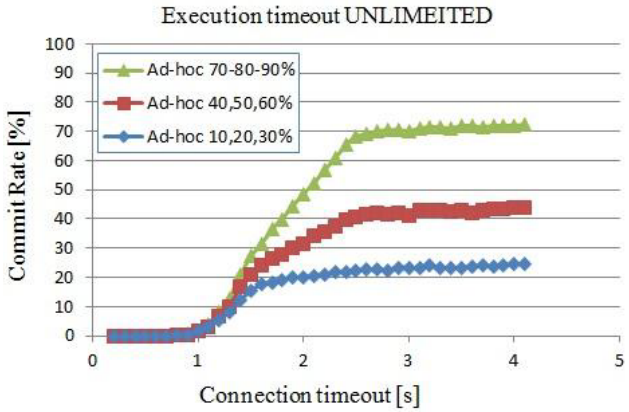

Fig. 3. Impact of Connection timeout on commit rate

To determine the ad-hoc communication contribution and DAlg contribution over transaction commit rate in the CFT model, we performed numerous simulations whose results are shown in Figs. 4-7. Figs. 4 and 6 show the mobile transaction commit rate against different disconnection rates and different percentages of mobile transactions whose function is WRITE. It is evident that higher level of ad-hoc support leads to higher commit rate. It is also evident that the mobile transaction commit rate is higher if the number of mobile transactions, whose function is WRITE, is higher.

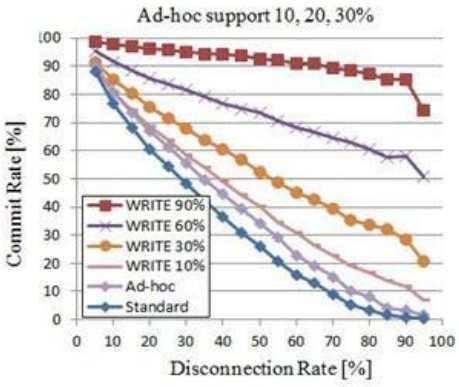

Fig. 4. CFT Model contribution over commit rate

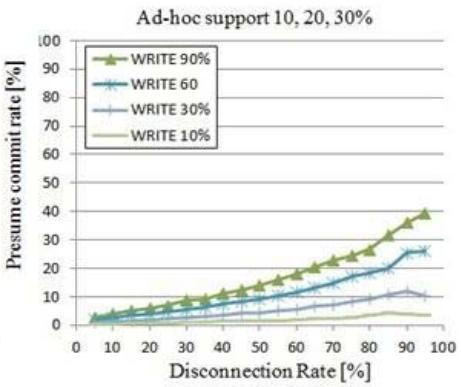

Fig. 5. DAlg contribution over commit rate

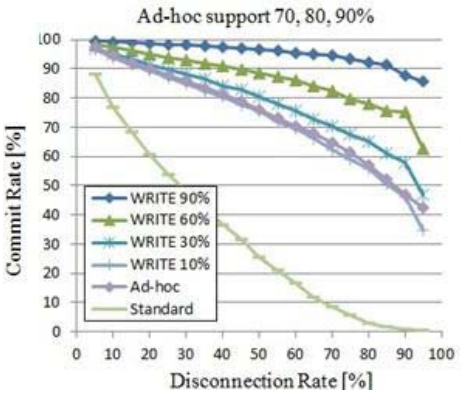

Fig. 6. CFT Model contribution over commit rate

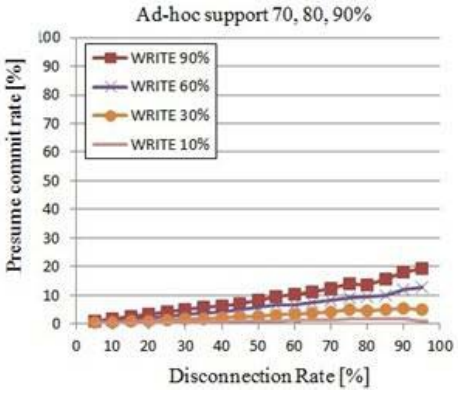

Fig. 7. DAlg contribution over commit rate

Figs. 5 and 7 show mobile transaction "presumed commit" rate against different disconnection rates and different percentages of mobile transactions whose function is WRITE ("presumed commit" rate is the percentage of committed mobile transactions by DAlg). From the charts it can be inferred that if the level of ad-hoc support is lower (Fig. 5), the impact of DAlg in the CFT Model on the mobile transaction commit rate is higher. The other way around, if the level of ad-hoc support is higher (Fig. 7), the impact of DAlg on the mobile transaction commit rate is lower.

7 Conclusion

In this paper we made a detailed review of the operation of a Connection Fault-Tolerant Model for mobile transaction processing. Our simulation-based performance analysis determines the contribution of (i) the existence of ad-hoc communication and (ii) the implementation of a decision algorithm to the mobile transaction commit rate. We also determined the connection timeout values that contribute the most for a high ad-hoc communication impact on transaction commit rate (before DAlg's activation). Simulation results show that the contribution of ad-hoc communication and DAlg on the transaction commit rate is counter proportional: if the level of ad-hoc support is lower, the impact of DAlg in the CFT Model on the mobile transaction commit rate is higher and vice versa. Conjointly, they both increase the transaction commit rate.

In our future work we plan to expand the parameters that the decision algorithm uses for decision making on behalf of the MH, as well as to employ the class of Deterministic and Stochastic Petri Nets (DSPNs) for modeling and analysis of the proposed model (deterministic transitions – to capture the timeout mechanisms, and exponential transitions – to capture the interarrival times of transactions, as well as the stochastic nature of random network disconnections), in order to evaluate a range of availability, reliability, performance and performability measures.

References

1. Gray, J.: Notes on Data Base Operating Systems. In: Operating Systems, An Advanced Course, pp. 393–481 (1978)
2. Santos, N., Ferreira, P.: Making Distributed Transactions Resilient to Intermittent Network Connections. In: Proc. of the 2006 International Symposium on World of Wireless, Mobile and Multimedia Networks, pp. 598 – 602. IEEE Computer Society, Washington (2006)
3. Kumar, V.: A Timeout-Based Mobile Transaction Commitment Protocol. In: Proc. of the East-European Conference on Advances in Databases and Information Systems, pp. 339–345 (2000)
4. Nouali, N., Doucet, A., Drias, H.: A Two-Phase Commit Protocol for Mobile Wireless Environment. In: Proc. of the 16th Australasian Database Conference, pp. 135–143 (2005)
5. Ayari, B., Khelil, A., Suri, N.: FT-PPTC: An efficient and fault-tolerant commit protocol for mobile environments. In: Proc. of SRDS, pp. 96–105 (2006)
6. Moiz, S.A., Nizamudin, M.K.: Concurrency Control without Locking in Mobile Environments. In: Proc. of the First International Conference on Emerging Trends in Engineering and Technology, pp. 1336-1339. Nagpur, Maharashtra (2008)
7. Ayari, B., Khelil, A.,Suri, N.: On the design of perturbation-resilient atomic commit protocols for mobile transactions. J. ACM Transactions on Computer Systems, 29 (2011)
8. Miraclin Joyce Pamila, J.C., Thanushkodi, K.: Framework for transaction management in mobile computing environment. J. ICGST-CNIR. 9, 19--24 (2009)
9. Ravimaran, S., Maluk Mohamed, M. A.: An improved kangaroo transaction model using surrogate objects for distributed mobile system. In: 10th ACM International Workshop on Data Engineering for Wireless and Mobile Access. ACM (2011)
10. Hu, S., Muthusamy, V., Li, G., Jacobsen, H.: Transactional Mobility in Distributed Content-Based Publish/Subscribe Systems. In: 29th IEEE International Conference on Distributed Computing Systems. IEEE (2009)
11. Li, G., Yang, B., Chen, J.: Efficient optimistic concurrency control for mobile real-time transactions in a wireless data broadcast environment. In: Proc. of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, pp. 443 – 446. IEEE Computer Society Press, Washington (2005)
12. Salman, M., Lakshmi , R.: Single Lock Manager Approach for Achieving Concurrency in Mobile Environments. In: Proc. of 14th IEEE International Conference on High Performance Computing, Springer (2007)
13. Hien, N., Mads, N.: A transaction framework for mobile data sharing services. In: Proc. of the 2nd International conference on Mobile Ubiquitous Computing, Systems, Services, and Technologies, pp.109—115. IEEE (2008)
14. Dimovski, T., Mitrevski, P.: Connection Fault-Tolerant Model for distributed transaction processing in mobile computing environment. In: Proc. of the 33rd International Conference on Information Technology Interfaces, pp.145—150. IEEE Conference Publications (2011)
15. Xiang, L., Yue-long, Z., Song-qiao, C., Xiao-li, Y.: Scheduling Transactions in mobile distributed real-time database systems. Journal of Central South University of Technology, pp. 545-551 (2008)
16. SimPy simulation package, http://simpy.sourceforge.net
17. Python Programing Language, http://www.python.org