A RESEARCH ON THE EFFECT OF AD-HOC COMMUNICATION ON MOBILE DISTRIBUTED TRANSACTION PROCESSING

Tome Dimovski
St. Kliment Ohridski University
Faculty of Technical Sciences
Bitola, Macedonia

Pece Mitrevski
St. Kliment Ohridski University
Faculty of Technical Sciences
Bitola, Macedonia

ABSTRACT

Mobile embedded systems increasingly use transactions for applications like mobile commerce, banking or commercial applications. A transaction in mobile distributed environment may involve multiple mobile devices, beside fixed ones, as full participants. In this paper we make a short review of the operation of the Connection Fault-Tolerant Model for mobile distributed transaction processing. We made simulation experiments study for the model. The performance analysis shows that ad-hoc communication support and transaction time-out considerably improve the transaction commit rate.

I. 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 banking 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 the 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 characterized with mobile devices with limited resources like processing, storage, energy capacity and continuously varying properties of the 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. The 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. 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. There are several other protocols, [3, 4, 5, 7], for transaction execution in distributed mobile environment, but almost all consider limited number of communication models.

The main contribution of this paper is performance analysis of the Connection Fault-Tolerant Model [8] for mobile distributed transaction processing. We study the impact of the level of ad-hoc communication support and transaction time-out on the mobile transaction commit rate.

The paper is organized as follows. Section II gives a survey of related work. In Section III we present model of the mobile environment, and in Section IV transaction model. In Section V we present our Connection Fault-Tolerant Model. In Section VI, we present simulation results and analysis. Section VII discusses conclusion.

II. RELATED WORK

All *Transaction Commit On Timeout (TCOT)* [3] protocol is based on 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 number of wireless messages in the execution. TCOT 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 execution of transaction and they are sending 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 the transaction. This protocol does not provide adequate management of mobility and failures caused by the network disconnection. M-2PC also does not provide mechanism to control the competitiveness of distributed transactions.

Fault-Tolerant Pre-Phase Transaction Commit (FT-PPTC) [5] protocol provides mechanisms for dealing with disturbances in the systems 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 execution of transaction. No mechanisms are developed for competition in the mobile distributed transactions. FT-PPTC doesn’t provide adequate management of mobility because when mobile hosts are disconnected from fixed network for a long time they can block resources on the fixed participants. This leads to an increased number of mobile transaction aborts.

In concurrency control without locking [7], concurrency control mechanism in mobile environment is proposed, by introducing the concept of *Absolute Validity Interval (AVI)* which is time period in which the data item is said to be valid. It doesn't use the traditional locking mechanism. 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 the transaction and resources are not unnecessary locked.

III. MODEL OF THE MOBILE ENVIRONMENT

In this paper we consider system model for a 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 fixed part and the mobile part of the network is conveyed via *Mobile Support Stations (MSS).* MSSs 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.

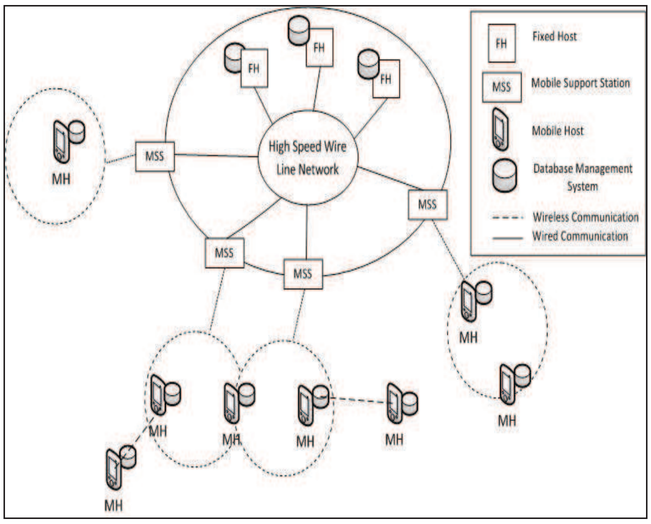

Figure 1: Model of the Mobile Environment

In the considered system model, first, MHs can communicate with the FHs through 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, to access database servers on fixed network they may connect through a neighboring MH which is in the covering area of any MSS.

In our scenario, we consider mobile distributed environment where MHs can communicate with each other, or with FHs through MSS, and MHs can ad-hoc communicate with neighboring MHs to reach fixed network.

We assume that database servers are installed on each FH, and each MH has a mobile database server installed.

IV. 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).*

In our model 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 participants about its decision. The CO should be executed on the fixed host or hosts. That means logs will be kept more safely.

V. CONNECTION FAULT-TOLERANT MODEL

A. 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 we proposed Connection Fault-Tolerant (CFT) Model [8] for distributed transaction processing in mobile computing environment. The CFT Model ensures the atomicity property.

The CFT model considers two communication scenarios:

1. The first scenario is a *Standard communication scenario* when MHs can directly connect to fixed network through MSSs.
2. The second scenario is an *Ad-hoc communication scenario* when MHs cannot directly connect to the fixed network through any MSS. In this scenario MHs can ad-hoc communicate with neighboring MHs to reach the fixed network.

In the Standard scenario, similar to [6], to minimize the use of the wireless communication and conservation of the resources of MHs we assign a *Mobile Host Agent (MH-Ag)*, which we add in the fixed network, to each MH. 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 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 are created for each participating MH.

In our CFT model we define additional function to MH-Ag that we called *Decision Algorithm (DAlg).* DAlg is used during the execution of a transaction when MH-Ag cannot directly or ad-hoc communicate 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 FIFO (First In First Out) queue list and makes a decision for MH to send "Yes" vote to the transaction CO. When the connection between MH and 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 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 MH to send "No" vote to the transaction coordinator.

The second communication scenario 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 through 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 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 his MH-Ag of the fixed network via *MH-RAg* of the neighboring MH which is in coverage area of any MSS.

B. *Connection Fault-Tolerant Model operation*

In this section we make a short review of the operation of the CFT model that has been presented in [8]. Fig. 2 illustrates the execution of a mobile transaction for the proposed model, but without the functions of the Decision Algorithm because in this paper our interest is concentrated on the Ad-hoc communication impact on mobile transaction execution.

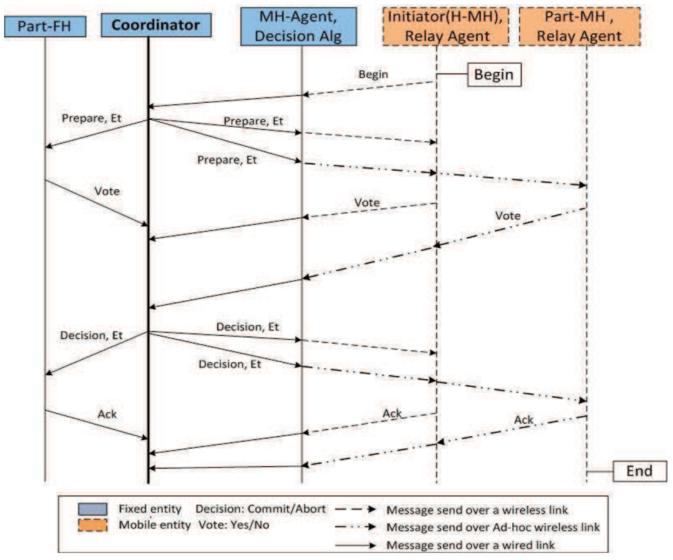

Figure 2: Mobile transaction processing

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

Transaction coordinator computes the execution time-out (E_t), which is a time limit for all participants to complete the execution of the TPFs and send a VOTE to coordinator. After that coordinator sends E_t and TPFs to all participating FHs and MH-Ags, asks them to prepare to commit the transaction, and enters the wait state. MH-Ags try to send TPFs to their participating MHs. If MH-Ag cannot communicate with a MH trough standard communication, it tries to connect through ad-hoc communication to any neighboring MH which is in the covering area of any MSS. That function is assigned to the MH-Relay Agent.

When the participants receive the "prepare" message, they check if they could commit the transaction. If so, participants send YES vote to the coordinator. MH sends vote to coordinator through its corresponding MH-Ag via standard or ad-hoc communication.

After the coordinator 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 execution time–out expires, the coordinator decides to abort the transaction and sends "Abort" message to all participants. Otherwise, if all received votes are "Yes" and execution time–out is not expired, the coordinator decides to commit the transaction and sends "Commit" message, with reset execution time–out to all participants. The participants need to acknowledge the coordinator decision before execution time–out expires.

VI. SIMULATION RESULTS AND ANALISYS

The simulation experiments study the impact of ad-hoc communication in CFT Model on the mobile transaction commit rate. For the simulations, we used SimPy [9], a process-based discrete-event simulation package based on standard Python programming language [10]. Table 1 summarizes our simulation parameters.

Table 1: Simulation settings

Parameter	Value
Number of Part-MHs	5-10
Fragment execution time (MH)	0.5s
Fragment execution time (CO)	0.3s
Transmission delay (wireless link)	0.4s
Transmission delay (wireless ad-hoc link)	0.9s
Transmission delay (wired link)	0.2s
Disconnection Rate	0 - 95%
Ad-hoc support	0 - 80%

Hence, disconnection rate is defined as the ratio of time where the participating MH is disconnected from the fixed network, against the total simulation time. Ad-hoc support is the ratio of time where 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 much lower compared to other. For that reason, in our simulation we define three groups that represent different parts of the wireless network with different level of ad-hoc support. Every MH in the wireless network is a member of one of the defined groups.

A simulation run is set to simulate 10 hours. Transactions are generated with exponentially distributed in-between intervals, with an average of 30 seconds. We assume that all transactions are of similar length, but experience different connection conditions.

The main parameter for performance evaluation is the mobile transaction commit rate. Figs. 3-5 show mobile transaction commit rate against different disconnection rates, and different ad-hoc support values. Transaction time-out is set to 5 seconds. For results shown in Fig. 3 we assume that every MH is a member of the same group. This means that the level of ad-hoc support is the same in any part of the wireless network. It is evident that ad-hoc support in the CFT model considerably improves the transaction commit rate. The ad-hoc communication impact is higher for networks where disconnection rate is higher.

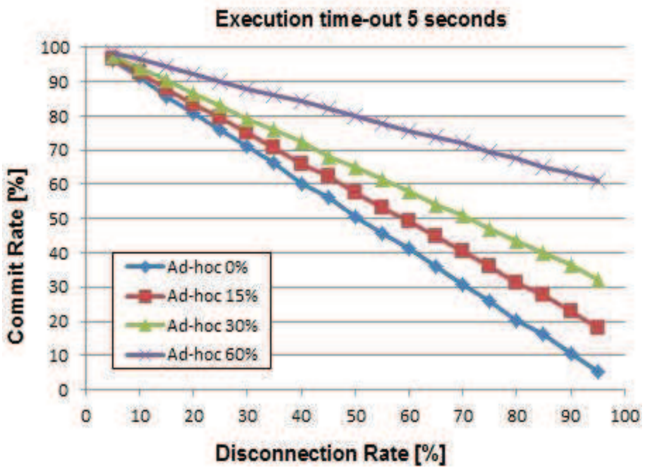

Figure 3: Impact of ad-hoc communication on commit rate (single group)

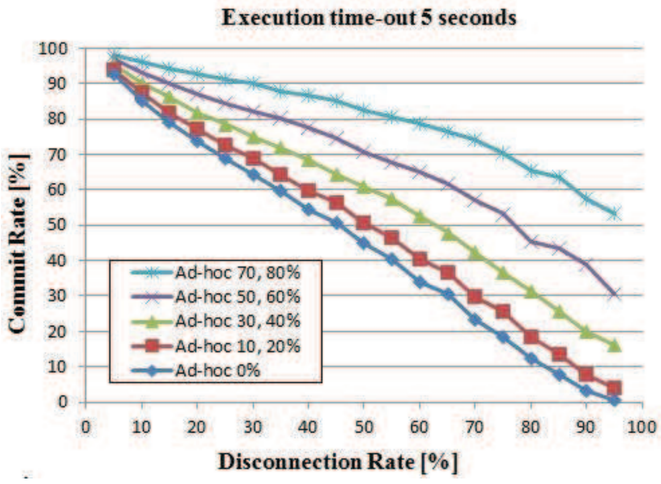

Figure 4: Impact of ad-hoc communication on commit rate (two groups)

Fig. 4 shows simulation results where MHs are classified in two different groups that have different levels of ad-hoc support. It is evident that ad-hoc support improves the transaction commit rate, but the percentage of improvement is lower compared to the previous scenario where the level of ad-hoc support was the same in each part of the wireless network. Compared to the previous scenario, commit rate slightly decreases when the disconnection rate rises all the way from 60 up to 95 seconds.

To present the influence of ad-hoc support in a highly dynamic wireless network, we classify all MHs in three groups that have different levels of ad-hoc support. From the chart in Fig. 5, one can conclude that commit rate increment is not evident as before, e.g. the highest improvement of the commit rate is about 11%.

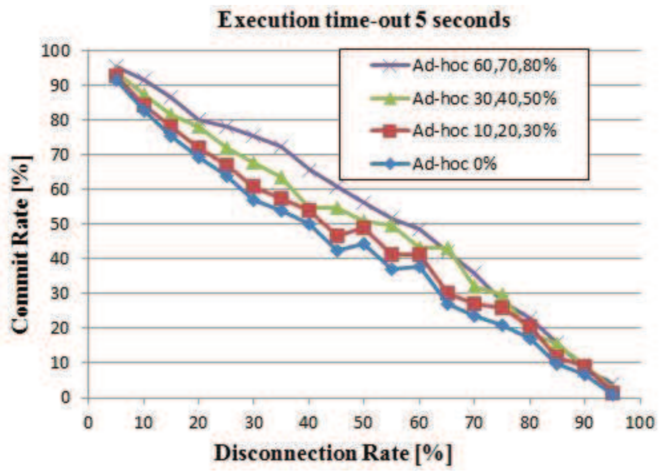

Figure 5: Impact of ad-hoc communication on commit rate (three groups)

This improvement comes with the increase in the number of wireless massages as shown in Fig. 6.

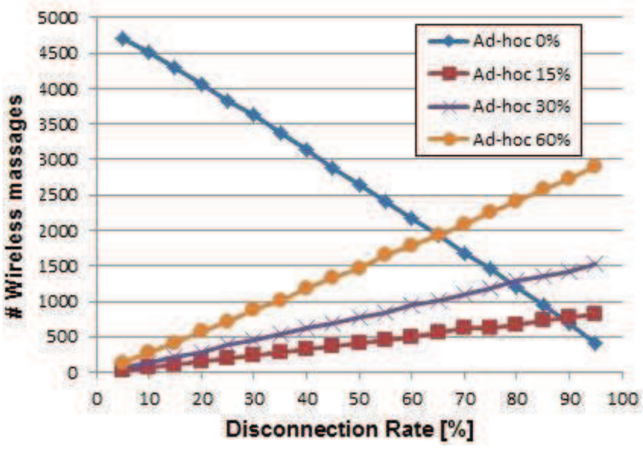

Figure 6: Number of wireless massages

To evaluate transaction time-out impact on the commit rate we fix the disconnection rate to 60%, and repeat the simulations for transaction life-times from 5 to 95 seconds with 5s increments. From the chart in Fig. 7 can be concluded that commit rate increment is more evident for increments of transaction life time up to 50-60 seconds. After that point, the commit rate slightly increases when the transaction life time rises all the way from 60 up to 95 seconds.

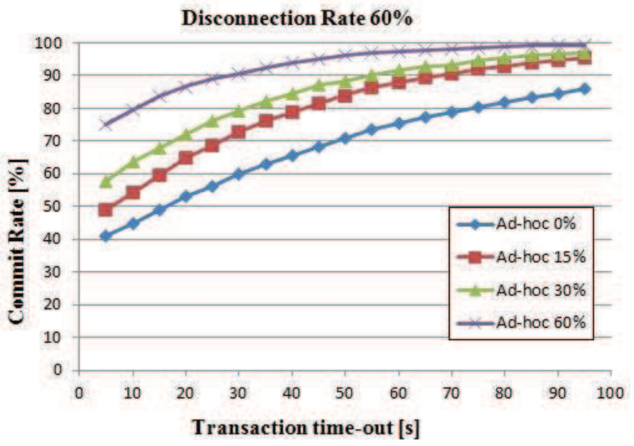

Figure 7: Impact of transaction time-out on commit rate

VII. CONCLUSIONS

In this paper we made a short review of the operation of the Connection Fault-Tolerant Model [8] for mobile distributed transaction processing. It is developed to increase the number of committed mobile transactions and shows resilience to connection failures of the mobile devices.

The performance analysis shows that ad-hoc support in the CFT model considerably improves the transaction commit rate. It is evident that the impact of ad-hoc support is higher for less dynamic wireless networks. This improvement comes with the increase in the number of wireless ad-hoc massages.

We also show the transaction time-out impact on the mobile transaction commit rate.

In our future work we plan to evaluate the performance of a model that includes a decision algorithm, as well, for which we expect to have positive impact on mobile transaction commit rate.

REFERENCES

[1] J. Gray, “Notes on Data Base Operating Systems”, Operating Systems, An Advanced Course, 1978.

[2] N. Santos, P. Ferreira, “Making Distributed Transactions Resilient to Intermittent Network Connections”, Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks. IEEE Computer Society, Washington, 2006.

[3] V. Kumar, “A Timeout-Based Mobile Transaction Commitment Protocol”, Proceedings of the East-European Conference on Advances in Databases and Information Systems, 2000.

[4] N. Nouali, A. Doucet, H. Drias, “A Two-Phase Commit Protocol for Mobile Wireless Environment”, Proc. 16th Australasian Database Conference, 2005.

[5] B. Ayari, A. Khelil, N. Suri, “FT-PPTC: An efficient and fault-tolerant commit protocol for mobile environments”, Proc. of SRDS, 2006.

[6] L. Xiang, Z. Yue-long, C. Song-qiao, Y. Xiao-li, “Scheduling Transactions in mobile distributed real-time database systems”, Journal of Central South University of Technology, 2008.

[7] S.A. Moiz, M.K. Nizamudin, “Concurrency Control without Locking in Mobile Environments”, First International Conference on Emerging Trends in Engineering and Technology, Nagpur, Maharashtra, 2008.

[8] T. Dimovski, P. Mitrevski, “Connection Fault-Tolerant Model for distributed transaction processing in mobile computing environment”, Information Technology Interfaces, 33rd International Conference, Dubrovnik, Croatia, 2011.

[9] URL < http://simpy.sourceforge.net/>

[10] URL < http://www.python.org/>

[11] M.H. Dunham, “A Mobile Transaction Model That Captures Both the Data and Movement Behaviour”, Mobile Networks and Applications, 1997.

[12] P.K. Chrysanthis, “Transaction Processing in Mobile Computing Environment”, IEEE Workshop on Advances in Parallel and Distributed Systems, 1993.

[13] E. Pitoura, “Maintaining consistency of data in mobile distributed environments”, Proc. 15th ICDCS, 1995.

[14] S.K. Madria, “A Transaction Model to Improve Data Availability in Mobile Computing”, Distributed Parallel Databases, 2001.

[15] G. LI, B. YANG, J. CHEN, “Efficient optimistic concurrency control for mobile real-time transactions in a wireless data broadcast environment”, The 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, IEEE Computer Society Press, Washington, 2005.

[16] M. Salman, R. Lakshmi, “Single Lock Manager Approach for Achieving Concurrency in Mobile Environments”, Proceedings of 14th IEEE International Conference on High Performance Computing, Springer, 2007.

[17] N. Hien, N. Mads, “A transaction framework for mobile data sharing services”, The second International conference on Mobile Ubiquitous Computing, Systems, Services, and Technologies, IEEE, 2008.

[18] G.K. Attaluri, K. Salem, “The Presumed-Either Two-Phase Commit Protocol”, Transactions on Knowledge and Data Engineering 14(5), IEEE, 2002.