Comparison Study of Commit Protocols for Mobile Environment

Bharati Harsoor[1]
[1]Dept of CSE, University College of Engg,
OU, Hyderabad, India

Dr.S.Ramachandram[2]
[2]Dept of CSE, University College of Engg,
OU, Hyderabad, India

***Abstract*—This paper presents a study of protocols to commit the transactions distributed over several mobile and fixed units and provides the method to handle mobility at the application layer. It describes the solutions to defeat the dilemma related to principle implementation of the Two Phase Commit (2PC) protocol which is essential to ensure the consistent commitment of distributed transactions. The paper surveys different approaches proposed for mobile transaction and outline how the conventional commitment are revisited in order to fit the needs of mobile environment. This approach deals with the frequency disconnections and the movement of mobile devices. This paper also proposes Single Phase Reliable Timeout Based Commit (SPRTBC) protocol that preserves the 2PC principle and it lessens the impact of unreliable wireless communications.**

Keywords- Mobile Transactions; Transaction Log; Transaction Recovery; Network disconnection; handoff; ACID properties.

I. Introduction

Due to the mobile computing standard, the mobile users can access information independent of their physical location through wireless connections. However, accessing and manipulating the information without confining the users to definite locations complicates the processing of data. Mobility and disconnected computing are two major issues in such environment. With the advancement in the distributed technology, consistency mechanism for a mobile transaction has become easier and manageable with more than one participant. To preserve data consistency *all* or *nothing* effect of transaction execution is usually enforced at commit time.

To ensure consistent termination of distributed transactions regardless of communication and site failure we use the following various commit protocols for mobile transactions. Along with study, we propose a new execution framework providing an efficient extension that supports the reliable execution of mobile transactions called Single Phase Reliable Timeout Based Commit (SPRTBC) protocol.

It is one phase commit protocol, which makes use of only *decision phase* to perform commitment of transactions. With the proposed model, during first step, no resources are blocked due to timeout approach. It confirms ACI (Atomicity, Concurrency, Isolation) properties, during second step, it preserves durability property; hence it supports disconnections and handoff having reduced blocking situations.

II. Comparison Study Of Commit Protocols For Mobile Environments

A. Two Phase Commit Protocol (2PC)

In distributed systems, an Atomic Commitment Protocol is required to terminate the distributed transactions. The most commonly used and standardized mechanism dealing with the commitment problem is Two Phase Commit protocol (2PC) [4][7]. It is the simplest and most used Atomic Commit Protocol. Generally, it follows two phases, voting and decision phase. In voting phase the coordinator requests all the participants to prepare to commit the transaction, if any of the participant responds No, the coordinator decides to abort and inform every participant to abort their local transaction, otherwise if all the participants votes Yes then CO decides to commit and informs all the participants to make their local transaction durable or permanent. The participants acknowledge the coordinator.

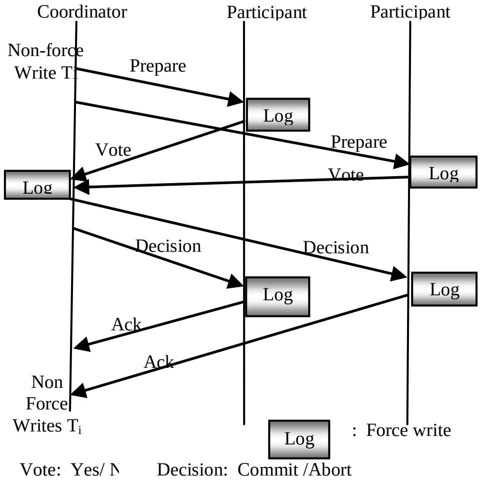

Figure 1. 2PC-Two-Phase Commit Protocol

Fig. 1 illustrates sequence of operations carried out in Two Phase Commit Protocol. The issues related to the Two Phase Commit Protocol with the mobile environment are, Inaccurate global decision ; means if the coordinator does not receive all votes before its timeout expiration it may decide to terminate globally in case where global commit is possible and Blocking situations; mean the following blocking conditions may arise in Two Phase Commit Protocol.

- The coordinator waits until the reception of every acknowledgement messages from the participants. Here no data is blocked.
- A participant waits after voting commit until reception of global decision. This situation may be constrained where the participant's local resources remain locked during that time. Such a participant is not permitted to unilaterally terminate the local transaction.

B. *Mobile - 2PC (M - 2PC)*

The aim of M-2PC (Mobile-2PC) [7] protocol is to globally commit mobile transaction T_m which is being executed over more than one host. Suppose that a transaction T_m is issued at MH called as Home-MH and which is attached to a BS called Home-BS. As MH transfers from one cell to next cell and it joins to a new BS that is called Current-BS. At commit time a commit demand is issued from the Home-MH, hence its current-BS (either it will be the Home-BS) becomes the commit-BS. The M-2PC protocol may terminate either with same cell or in a new cell protected by new BS. Fig. 2 illustrates the sequence of transaction executions used by M-2PC. The transaction execution is split into two phases; the initial one is almost equal to traditional 2PC, while the next phase controls the mobile wireless part.

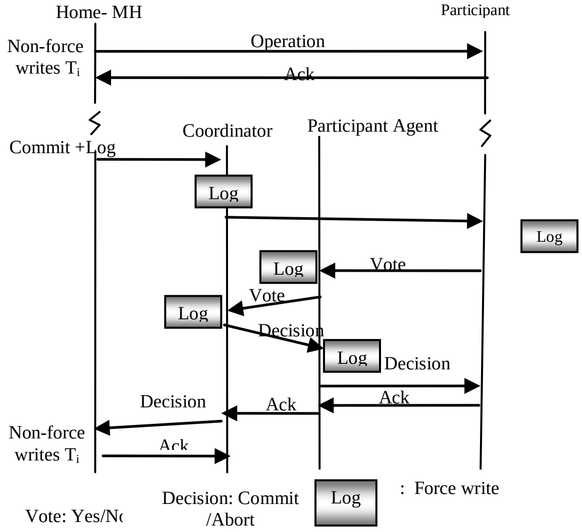

Figure 2. M-2PC – Mobile 2PC Protocol

The Home-MH (participant) sends the transaction to be carried out in batches to BS (Coordinator). When hand-off occurs the incomplete transaction information is transferred to the new BS which becomes the new coordinator. With each change in location, the MH may require to send the message to inform to the old BS that handoff may need to be achieved. While carrying out the process the participants and the coordinator may communicate between each other, so that all are involved during commit.

This solution may give a way to deal with mobility at application layer and embeds the mobility mechanism in the protocol. In M-2PC no message concerned to the protocol execution must be lost during a disconnection or a handoff. During disconnections the continuity of service is guaranteed because of three-tier architecture, where the agents (Coordinator for the mobile client and participant-agent for the mobile server) execute on behalf of the MHs.

During handoff, the MHs are in charge of telling their correspondents about the new location by transferring them a message after registering in a new cell. Also no loss of messages appears during a period of handoff processing (supports the disconnection handling and mobility control). This solves the problem of the address change. The MH must record the identity and location information of the correspondent as it needs when it registers in new BS. Also, there will be no loss of messages, while on the handoff processing. The drawback of this protocol is that it is not capable to handle disconnection and handoffs simultaneously.

C. *Unilateral Commit Protocol (UCM)*

The Unilateral Commit Protocol for Mobile (UCM) [8] environment supports off-line transaction execution and decreases the risk of abortion of such transaction during reconnection moment. It also supports the disconnection of one or more participants while commitment of the execution of the protocol and is particularly designed for mobile environment, which is based on the idea of single phase commit protocol.

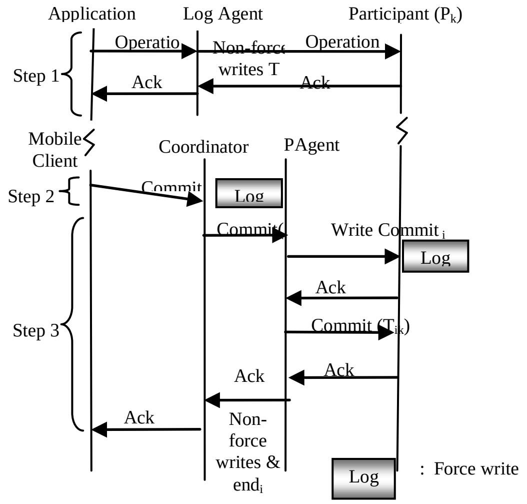

Figure 3. UCM – Unilateral Commit Mobile Protocol

Its message complexity is quite low (a single phase to commit the transaction), thereby saving an essential communication cost in wireless environment. UCM removes the voting phase of 2PC during which the coordinator verifies that participants can guaranty ACID properties or not. Having these properties assured at commit time at each participant site p_K, these operations are logged by log register (force write) and locally executed.

Each operation is acknowledged up to the request. Once all the acknowledgements are received by the application, it issues a commit request. The transaction operations and their acknowledgements are commonly logged to make sure the atomicity. If the transaction reaches validation phase then the global decision is commit. If any problem arises with the global transaction it is immediately aborted

At this point, ACID properties are locally guaranteed by the participants for all the local transaction branches. It reduces the cost of wireless communication by reducing the message complexity. A global commit is performed in a single phase, the decision phase. It is initiated by the transaction's operation log transfer from the application to the coordinator. Fig. 3 shows the sequence of operations carried out by the UCM.

The major issues related to UCM are, blocking situations: UCM coordinator waits if at least single Ack message is missing. Handoff/Mobility: with the UCM the handoff/Mobility problem was not particularly taken care.

D. Timeout Based Commit Protocol (TCOT)

"Transaction Commit on Timeout (TCOT) [4]," is based on a "timeout" approach for Mobile Database Systems, which is generally used to reach a final transaction termination decision (e.g. commit, abort, etc) in any message-oriented system. The transaction is being initiated and fragmented by the MH; the initial fragment is executed at MH while the remaining will be sent to coordinator. The coordinator distributes these left over fragments among the relevant DBSs (Data Base Server).

Let E_t being an upper bound of the execution time, just long enough to allow a fragment to successively finish its execution on participant site and S_t be the upper bound of data shipping time from MH to DBS. If timeout (Max (E_t +S_t)) occurs before the log arrives or not all the commit messages are received, the coordinator informs to all the participants about a global abort decision. A participant can unilaterally abort and inform the coordinator. A global commit is decided by the coordinator if it receives the updates log from MH before S_t expires and the commit messages from all participants. Moreover a static or moving coordinator is feasible in case of mobility. Fig. 4 Illustrates the sequence of transaction execution carried out by TCOT.

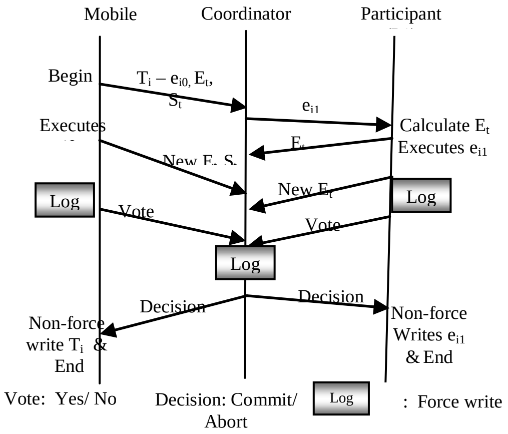

Figure 4. TCOT –Timeout Based Commit Protocol

TCOT is specifically suited for wireless environment; timeout mechanism is the only way to reduce the impact of slow and unreliable wireless link. In case if it's quickly moving and frequent disconnection the abort rate increases, also the message rate can increase. With increase of MPL (Multi Programming Level) the performance degrades. TCOT performs well in an environment where the communication over wireless connection which is highly available and reliable. Timeout not only enforces the termination condition but also the entire execution period as well.

TCOT commits transaction in minimum number of uplinks (user to server direction) by permitting every processing host participating in the transaction to have independent decision making capability based on the timeout mechanism. TCOT is designed for a system offering a connectivity mode known as, Mobile connectivity which permits the users to remain connected all the time to the network by the wireless channel. In Intermittent connectivity mode, the user voluntarily decides as to connect/ disconnect from/to network.

E. Reliable Timeout Based Commit Protocol (RTBCP)

We have proposed the commit protocol to implement the mobile transactions called Reliable Timeout Based Commit Protocol (RTBCP) [9], which is based on the "timeout" approach for mobile database systems to reach transaction global decision. This execution model has the Mobile Host (MH**)** and the Base Station (BS) communicating with each other through messages**.** The Mobile Transaction (MT) is initiated by the MH and is executed either at mobile host or at the fixed hosts. Hence it uses distributed mode of execution between MH and the data base servers (DBS) available at wired network, henceforth these data base servers are called as Fixed Cohort Units (FCUs).

The designed algorithm in [9] depicts Transaction execution at Mobile Host (MH), initiates transaction T_i and split T_i into set of fragments, the first fragment is executed at MH and the remaining fragments of T_i are sent to coordinator (CO) available at the base station. The CO distributes these fragments among various fixed cohorts (FCU) at the wired network. Let E_t being an upper bound of the execution time, i.e. just long enough to allow a fragment to successfully finish its entire execution on participant site. Upon receipt of their respective fragment, each participant calculates e_{tk} (time required to execute fragment at site k), Since we use logs and databases locally it is not necessary to calculate data shipping time S_t [as in TCOT] from the MH to the DBS. If the timeout (Max (E_t)) expires before all the commit messages are received, the coordinator informs all the participants about global abort decision. A participant can unilaterally abort and inform the coordinator. A global commit is decided by the coordinator if it receives commit messages from all the participants.

Determining the value of E_t practically need more rational verification. In case of handoffs and frequent disconnection the abort rate increases, accordingly the message rate also increases. With increase in number of transactions the performance degrades. RTBCP mainly suits for a wireless environment with highly available and reliable wireless link, with timeout mechanism it avoids blocking situations and by maintaining logs locally it reduces the commit time which produces good performance over slow and unreliable wireless link.

At this point, ACI properties are locally guaranteed by the participants for all the local transaction branches. However, since participants are not aware of the termination of the transaction, they cannot guarantee the Durability property.

Durability is ensured by the coordinator itself based on the messages received from all the participants within timeout duration, which gets the *T_i* log produced by all the nodes and force-writes on stable storage and at the same time, the coordinator then broadcasts the *Commit* decision to all participants and forgets the transaction and will not wait for their acknowledgments. Hence there is no problem of blocking I/O. Based on the global decision MH & FCU's update their databases. Once this is achieved, the ACID properties are guaranteed altogether for all the transaction branches. If T_i fails to commit, then it may initiate cascade rollback.

The absence of an abort message after E_t expires indicates a global abort. Thus, the commit time is the time indicated by E_t. A premature abort is indicated by an abort message. In 2PC [7], the FH waits for messages from participants to make any decision. If the wait is over unsuccessfully, then it aborts the transaction. In TCOT, the absence of a message is enough to make a decision, thus no additional phase is necessary. The RTBCP also works on timeout, all the participants (MH & FCUs) decide to abort and they will not wait for any coordinator decision. Fig. 5 shows the transaction's execution at MH & FCUs.

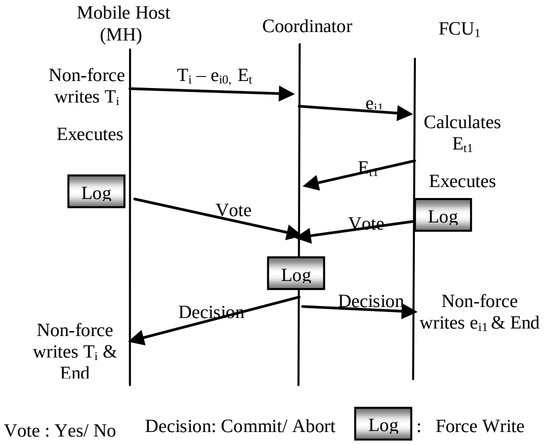

Figure 5. RTBCP -Transaction executions at MH

F. Modified Reliable Timeout Based Commit Protocol (MRTBCP)

The Modified Reliable Timeout Based Commit Protocol is one-phase commit protocol which is an extension to the RTBCP. It supports off line execution, disconnection and mobility. It eliminates voting phase of 2PC during which the coordinator verifies that the participants can guarantees ACID properties.

The MRTBCP initiates and fragments the transaction (T_i) at transaction manager at MH(TM-MH), the first fragment e_{i0} is being executed at MH and the remaining fragments of T_i i.e. T_i - e_{i0} are sent to the various participants (MH and part-FHs) for execution. Once the participants receive their respective fragments, they compute and send E_t to the TM-MH, after receiving all E_ts , the MH calculates maximum Time T_m= Max (E_{t0},E_{t1},…..,E_{tn}) required to execute the transaction at MH & Part-FHs.

While executing at Participant, if time expires before it acknowledge TM-MH, then the TM-MH decides to abort and issues an abort request to CO. The participant can unilaterally abort the transaction, if it does not receive the acknowledgement for commit before time expiry.

A global commit is decided by the MH, if it receives an acknowledgement from all the participants before time expiry. Once all the acknowledgments are received by the participants, the TM-MH issues a commit request to CO. The coordinator force-writes and delegates the commit messages to participants at wired network and waits for an acknowledgement. After receiving all acknowledgments the coordinator informs the TM-MH, about the decision.

The transaction's commit and acknowledgment messages are continuously logged at Log Agent to ensure atomicity. If the transaction reaches validation phase then the global decision is to commit or else the transaction is immediately aborted. The maintenance of databases and log autonomously at each participant insures the proper recovery in case of failure. At this point, ACID properties are guaranteed by the participants for all the local transaction branches. Fig. 6 outlines the sequences of execution transaction at MH and FH using MRTBCP.

The MRTBCP also takes care of handoff problem, specifically to handle the situations in case of mobility. Although the handoff process leads to decrease in the performance with MH, increasing the frequency of handoffs does not introduce an additional degradation in performance.

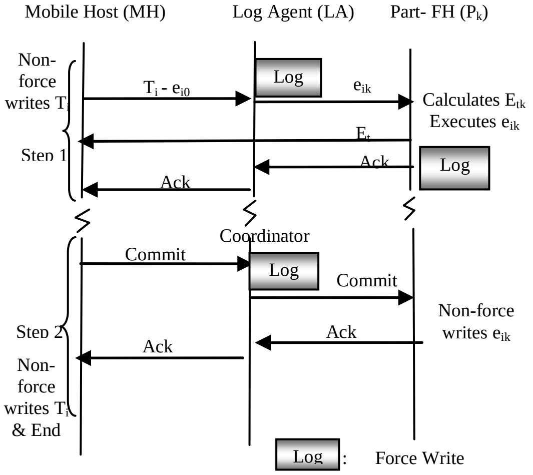

Figure 6. MRTBCP- Modified Reliable Timeout Based Commit Protocol

The performance with participants is not affected by the frequency of the handoff process. This is because, after the submission of a mobile transaction to its participant, there is no need to transmit messages between the coordinator site and the participant until the transaction is completed. Therefore the search process for the coordinator site is required too rarely to affect the performance. It is mainly suited for a wireless environment; a timeout mechanism avoids the blocking situations, since we use logs and databases locally, it reduces the commit time producing good performance metrics over slow and unreliable wireless link. Figure 6.3 shows the handoff management approach in case of mobility with MRTBCP.

G. *Single Phase Reliable Timeout Based Commit Protocol (SPRTBCP)*

The basic idea of SPRTBCP is to eliminate the voting phase of the 2PC by introducing the properties of the local databases. In this context, a transaction is initiated by the TM-MH (Transaction Manager at MH) and this transaction is assured to be committed in a failure free environment by distributing the fragments at various participants (Part-FHs and MHs). When the acknowledgments for all fragments of a transaction T_i are received by the TM-MH, it means, the transaction fragments i.e. $e_{i0,}$ $e_{i1,}$ $e_{i2,\ldots,}$ e_{in} have been successfully executed till completion. At this point, TM-MH submits its positive commit message to the CO which can directly ask each participant host accessed by the transaction T_i to commit, with no synchronization between the sites. If a transaction fragment, say e_{ik} is aborted by Participant$_k$ during its execution for any problem, the CO simply asks each accessed participant to abort that transaction. Assume that Participant$_k$ crashes during the one-phase commit of transaction T_i during which T_i may have been committed at other hosts. To ensure T_i's atomicity, the effects of the transaction branch e_{ik} have to be forward recovered in Participant$_k$.

The participants executing their respective fragments launch positive acknowledgement and also update their local logs that contain physical redo log records generated during the execution of this operation along with the respective Log Sequence Number (LSN). The CO registers the commit decision in its own log. Once Participant$_k$ recovers from its crash, it redoes set of operations using local log records with highest LSN and reinstalls them in the database.

To enforce transaction atomicity with the site autonomy, SPRTBCP utilizes logging schemes introduced in their respective participant's database systems. On each participant site, local logs keep up each operation sent to it before its execution. During the decision phase, when a participant receives the commit decision, it updates the local database.

If the local database crashes before completing the commit, it will abort the transaction. After the database recovery, the Participant re-executes all operations found in its log and belonging to the globally committed transaction. This approach guarantees global atomicity while preserving site autonomy. To achieve high performance and throughput, transactions are to be interleaved and executed concurrently. We assume that the concurrent executions of transactions are coordinated such that there is no interference among them.

In order to recover from failures, SPRTBCP maintains logs locally with each of the participants. Indeed, maintaining the logs locally, the CO must guarantee that the decision must be non-force written in stable storage before broadcasting its decision. In case of a participant crash during the one-phase commit, the failed transaction branches will be re-executed due to the operations registered in their respective redo logs.

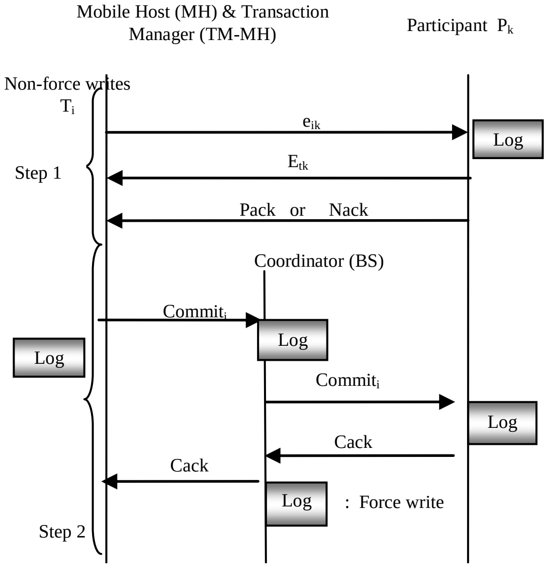

Figure 7 Single Phase Reliable Timeout Based Commit Protocol (SPRTBCP)

Fig. 7 gives the sequence of executions carried out during the transaction processing and shows the series of operations scenario introduced by the SPRTBC protocol.

1) *Disconnection*

In the presence of disconnections, a large number of transactions can be expected to abort. While executing any transaction, the coordinator mainly concerns with the wireless network, MHs and its current location. During processing, the MH might be either connected to the network fully or totally disconnected or partially connected or weak connection (very low bandwidth).

Specifically, all transaction fragments which undergo a disconnection will be either aborted by the coordinator when it fails to receive a response from a participant site before timeouts or blocked if coordinator cannot proceed until it has collected all the necessary responses**.**

The designed model proposes the system to operate independently even during total disconnection. In case, a mobile host physically detaches from the network, SPRTBCP has enough information locally available for its autonomous operation during disconnection. SPRTBCP maintains log and database locally at each host supporting offline execution providing less number of aborts during disconnection.

2) *Handoff/ Mobility Management*

The mobile host on movement from one mobile cell to another, it connects to the new MSS. The movement of the local transactions execution should support the mobility across different mobile cells; while the shared transactions support the mobility of standard transactions across different mobile sharing areas when the mobile host is moving across different mobile cells.

III. Performance Analysis Of Various Protocols

This segment represents the presentation study of above mentioned commit protocols used for mobile environment. A study provides the performance metrics used to evaluate the performance of commit protocols viz. by means of number of message transfers, writing disk of log records (force writes), blocking property, and real-time atomic property, impact of frequency disconnection, latency, handoff association.

The performance of SPRTBCP is compared with 2PC, M-2PC, UCM, TCOT, RTBCP and MRTBCP. The 2PC is the most well-known blocking commit protocol, while M-2PC is improved version of 2PC. UCM is one phase commit protocol that has been proposed for light weight processing and TCOT is timeout based non-blocking commit protocol.

The number of messages presented includes the execution and the commitment phases. In TCOT with normal execution only 2 message rounds are required whereas M-2PC requires minimum of 3 message rounds but RTBCP message complexity is almost similar to TCOT. The UCM and MRTBC Protocols use only one phase for commitment of transaction, since they add logging messages at base station, during commitment phase of transaction, due to which the failures are handled effectively but there is a small increase in the message complexity.

To overcome the problem, we have designed an extended version of MRTBCP called "Single Phase Reliable Timeout Based Commit Protocol (SPRTBCP)". It maintains log locally at each mobile and fixed host participant. SPRTBCP proposes one phase, reliable, efficient and non-blocking atomic transaction commit protocol.

Table I summarizes the principal properties of protocols studied over. To commit a transaction, the best protocol in terms of wireless messages is UCM. This is obtained at price of making strong assumptions about the local concurrency and recovery mechanisms. This may limit its usability in arbitrary heterogeneous systems. TCOT adopt the latest approaches which are completely different from 1PC or 2PC protocols. The other protocols preserve 2PC principles and try to optimize it to fit mobile environment requirements.

When the number of nodes increases in a cell, the performance of the transmission channel decreases as soon as flow threshold is reached. The conflict rate also increases leading to decreasing throughputs. The TCOT protocol gives very good performance when the MPL (Multiple programming level) is low. With an increasing MPL its performance deteriorates significantly, hence it becomes the less powerful protocol. With an increase in MPL the throughput of 2PC becomes closer to UCM & M-2PC.

TCOT has the best latency in case, with and without mobility of nodes, because the timeout limits the processing time of a transaction in all cases. But high value timeouts may lead to good throughput but increase latency.

A small value of Δ_t (Extension requests) may generate a large number of T_i aborts or request for the extension of Δ_t may affects throughput and message cost. Table 1 describes the impact on various commit protocols due to the disconnection.

Impact of handoff on M-2PC cannot be included as it is designed for supporting mobility management. UCM does not support mobility. TCOT handles mobility and disconnections but compared to RTBCP, it increases commit time [9] hence reduces overall throughput. MRTBCP also supports for the mobility, disconnections and handoff. It is one phase and timeout protocol hence produces better performance by reducing latency, message complexity and increased throughput over other protocols.

TCOT, RTBCP, MRTBCP and SPRTBCP are semantic based commit protocols, they eliminate the uncertainty period of transaction termination and the blocking effects, where they allow a participant to unilaterally commit transaction and release the resources it holds.

If the final decision is global abort, compensation is used semantically to undo the aborted transaction effects. The protocols like UCM, 2PC follow strict atomicity where they follow traditional ACID requirements. Table 1 gives the detailed analysis of the performance of various protocols used for the commitment of mobile transactions.

A. *Performance metrics*

The proposed technique is simulated and performance metrics are analyzed. The results of experiment are presented to verify the performance of the SPRTBCP protocol

1) *Effect of Disconnection on Commit Rate*

Disconnection may also occur involuntarily and unpredictably. Figures 8(a) and (b) depict the effect of disconnection on commit rate. SPRTBCP commit rate is compared based on timeouts with TCOT, RTBCP and MRTBCP (Figure 8(a)). SPRTBCP makes use of single phase operation and maintains log and database locally at each host supporting offline execution. Because of which, SPRTBCP completes transaction execution efficiently in case of disconnections producing higher commit rate compared to other protocols.

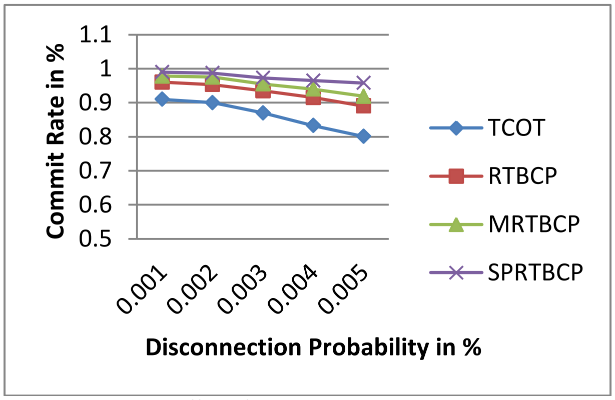

Figure 8(a) Effect of Disconnection on Commit Rate

Figure 8 (b) shows the commit rate compared with 2PC, M-2PC, UCM, RTBCP and MRTBCP considering disconnection as abort. The results show that SPRTBCP provides almost more than 95% of commit rate with the disconnection probability of 0.005%. If the traditional 2PC is executed in mobile environment, the number of disconnections increase, leading to transaction aborts i.e. the CO tries to communicate with a disconnected MH will cause blocking of resources. As a result, the commit rate reduces. In case of M-2PC and UCM, message overheads lead to decrease in the commit rate.

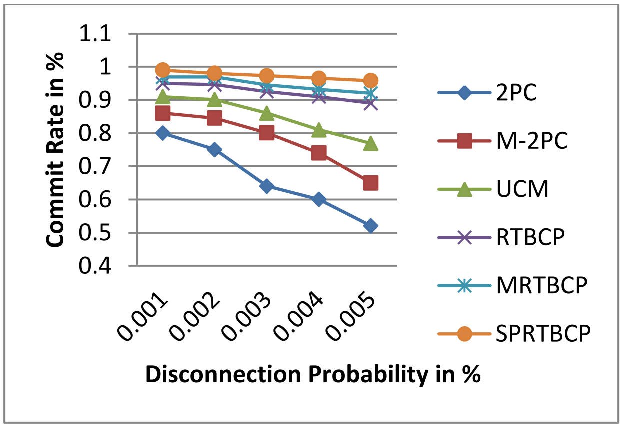

Figure 8(b) Effect of Disconnection on Commit Rate

2) Effect of Disconnection on Abort Rate

As frequent are disconnections, as transaction abortions are. This is not acceptable in mobile environments because frequent disconnections are not exceptions but rather are part of the normal mode of operation, so they should not be treated as failures. Contrary to the traditional 2PC, a protocol must not account on MHs to be continuously available to participate in the transaction commitment. Due to maintenance of log and database locally SPRTBCP tolerates disconnections providing less number of aborts.

Figure 9(a) shows the effect of disconnection on abort rate in comparision with TCOT, RTBCP and MRTBCP. It is observed that with the single phase commit operation and having offline execution without Log Agent at the wired network, SPRTBCP produces less number of aborts compared to the other protocols.

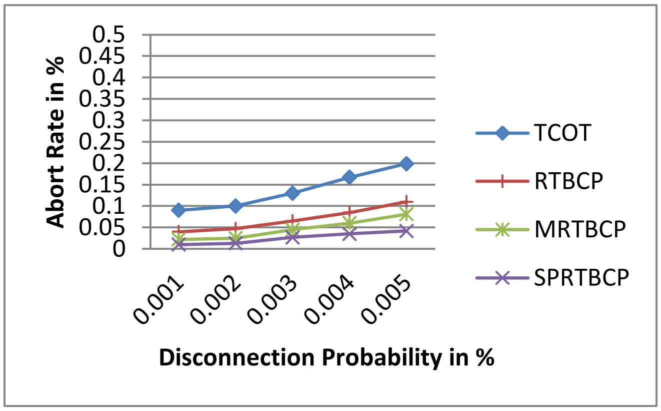

Figure 9(a) Effect of Disconnection on Abort Rate

Figure 9(b) shows the effect of disconnection on abort rate compared to the other protocols. We can observe that as the probability of disconnection increases the abort rate will also increase.

It is verified that SPRTBCP produces more efficient results (having very less abort rate) compared to 2PC, M-2PC, UCM, TCOT, RTBCP and MRTBCP and proved that SPRTBCP is more reliable in case of disconnections.

TABLE I: PERFORMANCE OF COMMIT PROTOCOLS

Protocol	**No. of phases**	**Atomicity**	**Site of transaction execution**	**Message complexity**	**Impact of frequency disconnection**	**Impact of latency**	**Handoff management**
2PC	2	Strict	FH	4n	Increase in number of aborts	Bad	Registration level
M-2PC	2	Strict	MH & FH	4n-1	Increase in number of Aborts until resources released	Medium	Protocol level
UCM	1	Strict	MH & FH	2n	Delay local transactions	Good	-----
TCOT	1	Semantic	MH & FH	$(2n-1) + \Delta_t$ Δ_t: no. of timeout extensions	Increase in number of Aborts	Good	Registration and protocol level
RTBCP	1	Semantic	MH & FH	$(2n-1)+ \Delta_t$	Increase in number. of Aborts	Good	Registration and protocol level
MRTBCP	1	Semantic	MH & FH	$(2n-1) + \Delta_t$	Increase no. of Aborts	Good	Registration and protocol level
SPRTBCP	1	Semantic	MH & FH	$(2n-1)+ \Delta_t$	Increase in number of aborts	Good	Registration and protocol level

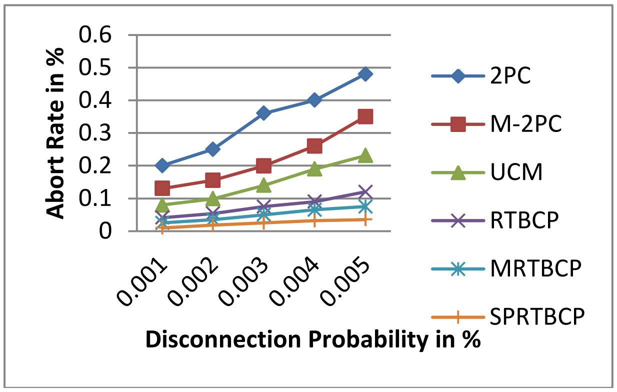

Figure 9(b) Effect of Disconnection on Abort Rate

IV. CONCLUSION

Transactions are main building blocks to have reliable systems. It could be provided by guaranteeing data consistency, concurrency control and recovery in case of failures due to disconnections and handoff.

This paper presents an overall comparison analysis of commit protocols for mobile environment. The study is mainly based on the performance metrics Viz. impact of frequency disconnection, latency, handoff management etc.

This revise presents the ongoing research which consists of the design and experiment of dedicated protocol that satisfies as several requirements of mobile surroundings as possible. To therapy to this condition a new atomic commitment protocol devoted to mobile and distributed computing is extremely desirable.

Hence, we proposed Single Phase Reliable Timeout Based Commit Protocol as extension to Modified Reliable Timeout Based Commit Protocol that increases a new commitment protocol, which suits for mobile transactions, designed to preserve all the above performance metrics. The protocol aims at handling new challenges including site failures and message loss blocking-free manner.

In addition to MRTBCP, it is single phase commit protocol without the Log Agent, due to which it reduces message complexity and average commit time. It is proved that, even in case of disconnections, failures and during mobility SPRTBCP produces better performance and reliable execution of transactions compared to the existing ACPs like 2PC, M-2PC, UCM and TCOT.

REFERENCES

[1] L. Amanton, B. Sadeg, S. Saad, Bouzefrane, “Disconnection Tolerance in Soft Real Time Mobile Databases,” Proceedings of 16th International Conference on Computers and Their Applications, USA. 2001,pp. 98-101.

[2] B. Bose, S. Sane, “Distributed Timeout Based Transaction Commit Protocol for Mobile Database Systems,” in Proceedings of International Conference and Workshop on Emerging trends in Technology (ICWET-2010),2010, pp. 518-523.

[3] C. Bobineau, C. Labb’e, C. Roncancio, P. Serrano- Alvarado, “Comparing Transaction Commit Protocols for Mobile Environments,” in Proceedings of the 15th International Workshop on Database and Expert Systems Applications (DEXA’04) 1529-4188/04 IEEE.

[4] V. Kumar, N. Prabhu, M. H. Dunham, and A. Y. Seydim, “TCOT- A Timeout-Based Mobile Transaction Commitment Protocol,” *IEEE Transactions on Computers*, 51(10), 2002.

[5] P. Serrano, C. Roncancico, M. Adiba, “A Survey of MobileTransactions,” DAPD Jnl., 16(2), 2004.

[6] M. M. Goreyand , R. K. Ghosh, “The Recovery of Mobile Transactions,” in Proceedings of the 11th International Workshop on Database and Expert Systems Applications (DEXA’00) 2000 pp. 23, IEEE.

[7] N. Nouali, A. Doucet, and H. Drias. “A two-phase commit protocol for mobile wireless environment.” in H. E.Williams and G. Dobbie, editors, *Sixteenth Australasian Database Conference (ADC2005)*, volume 39 of *CRPIT*, pages 135–144, Newcastle, Australia, 2005. ACS.

[8] C. Bobineau, P. Pucheral, and M. Abdallah. “A Unilateral Commit Protocol for Mobile and Disconnected Computing”, In *PDCS*, USA, 2000.

[9] B. Harsoor, S. Ramachandram, “Reliable Timeout Based Commit Protocol”, in Proceedings of 2nd International Workshop on Trust Management in P2P Systems (IWTMP2PS-2010) CNSA-2010, Springer Verlag 2010, pp. 417-423.

AUTHORS PROFILE

Mrs. Bharati Harsoor received her bachelor’s degree in CSE (1995), Masters in Computer Science (2001). She is a Research Scholar at Osmania University; Hyderabad.She is presently working as Associate Professor, Department of Information Science, PDA College of Engineering, Gulbarga, Karnataka State, India and Published papers in various National/International Conference and Journals. Her areas of interests include Mobile Computing, Databases, and Software Engineering. She is member of Institute of Electronics and Telecommunication Engineers (IETE).

Dr. S. Ramachandram (1959) received his bachelor’s degree in Electronics and Communication (1983), Masters in Computer Science (1985) and a Ph.D. in Computer Science (2005). He is presently working as a Professor and Head, Department of Computer Science, University College of Engineering, Osmania University, Hyderabad, India. His research areas include Mobile Computing, Grid Computing, Server Virtualization and Software Engineering. He has authored several books on Software Engineering, handled several national & international projects and published several research papers at international and national level. He also held several positions in the university as a Chairman Board of Studies, Nodal officer for World Bank Projects and chair of Tutorials Committee. He is a member of Institute of Electrical and Electronic Engineers (IEEE), Computer Society of India (CSI) and Institute of Electronics and Telecommunication Engineers (IETE).