Volume 2, Issue 9, September 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Backup Two Phase Commit Protocol (B2PC) renders the Trustworthy coordination problem over Distributed Transactions

P. Jamuna* *Department of CSE Pondicherry University*
S. Sathiyadevi *Department of CSE Pondicherry University*
S. Tamilarasi *Department of CSE Pondicherry University*

Abstract— Today all the business transactions are being deployed over the Internet using Web Service Atomic Transaction specification (WS_AT). For such distributed transactions always a trustworthy coordinator is needed and this trustworthy coordinator should able to tolerate the Byzantine Fault. In order to overcome such fault, this paper proposes the Byzantine Fault Tolerance (BFT) Technique using Backup Two Phase Commit protocol (B2PC). In which our B2PC protocol reduces the Byzantine Agreement usage, because using Byzantine Agreement in each and every phase of operation is impossible and it could be prohibitively very expensive. Our B2PC protocol overcomes the major issue faced by the atomic Two Phase Commit protocol is resource blocking problem and this problem can be overcome by adding a backup phase in normal 2 Phase Commit protocol. According to WS_AT specification, the coordinator of transaction provides a set of core services to initiator and to participant. Our proposed BFT Technique can achieves high degree of availability, reliability, security and also less expensive.

Keywords— Distributed Transactions, Web Services, Atomic Transactions, Byzantine Fault Tolerance Technique, Byzantine Fault, and Resource blocking Problem.

I. INTRODUCTION

In distributed transaction environment, a trustworthy coordinator should be tolerable towards Byzantine Fault (i.e. it is an arbitrary fault). In order to overcome such problem, this paper proposes a Byzantine Fault Tolerance Technique (BFT) with the usage of lightweight protocol named Backup Two Phase Commit protocol (B2PC). In the existing system they have used normal atomic Two Phase commit protocol, where it has some issue such as Blocking problem i.e. if the coordinator fails to operate and at the same time some participants has confirmed itself to commit state. In this circumstance to end the blocked transaction, the participant should wait until the coordinator gets recovery. The blocked transactions continue to keep all the resources until they obtain the final decision from the coordinator after its recovery. Thus, the blocking problem reduces the high degree of data availability. To overcome this issue, three-phase commit (3PC) protocol [13] [14] was proposed. On the other hand 3PC protocol involves an extra round of message communication to achieve non-blocking problem. Due to extra round of message communication it degrades the system performance when compared to 2PC protocol.

Our proposed Byzantine Fault Tolerance Technique which comprises a set of protocols, mechanism and some core services such as Activation service, Registration service, Completion and Coordination service are incorporated into a framework. This BFT framework uses the Backup Two Phase Commit Protocol (B2PC) instead of running a Byzantine Agreement during the registration of each participant. Only when the participants are non faulty they can enter into Backup Two Phase Commit protocol.

In this paper, we propose Backup Two Phase Commit Protocol (B2PC) by including backup phase to atomic Two Phase Commit protocol. In this, a backup site is employed for each coordinator. After receiving reply from all participants in the first phase, its starts the 2nd phase of backup phase in which the coordinator will communicates its final decision to the backup site. Afterwards, it broadcast it's the final decision to participants. When the blocking problem occurs due to a coordinator site failure, the participant sites will discuss with the coordinator's backup site and then it follows the termination protocols. In BFT framework, the coordinator provides some set of core services to participant and to initiator, if we harden those core services the coordinator can be a trustworthy entity even in untrusted environment over the Internet. However, in awful case if both the coordinator site and the Backup sites get failed means the participant should wait of recovery of either of coordinator site or backup site and this Backup Two Phase Commit protocol works closely like 2PC.

II. DISTRIBUTED COMMIT PROTOCOLS

In the literature, various commit protocols have been proposed, most of the protocols are based on 2PC protocol. Some of the familiar variants of 2PC protocol are presumed commit and presumed abort protocols [3]. The other

protocols consist of early prepare [6], coordinator log [12]. Since the centralized 2PC protocol having the communication is only between the coordinator and the participants alone. In this paper, we do not talk about other protocols since they are not worried with the resource blocking problem. In this section, we briefly explain the 2PC, 3PC and Enhanced Three Phase Commit (E3PC) protocols.

A. Two Phase Commit Protocol (2PC)

The atomic Two Phase Commit (2PC) protocol or blocking protocol is a typical distributed commit protocol that involves two phase are Prepare phase and Commit phase. Consider the scenario that 2PC protocol does not have any failures and the operations are as follows:

1) First phase (Prepare phase)

Coordinator: Initially the coordinator will broadcast the Begin_commit request message to all participants and enters into wait state.

Participant: When the participant receive the request message, If the participant want to commit the transaction means it respond with the Vote_commit message to the coordinator and enters into ready state. Otherwise, the participant responds with the Vote_abort message to the coordinator.

Coordinator: When the coordinator receives the reply from participant it starts 2nd phase.

2) Second phase (Commit phase)

Coordinator: If the participants reply with Vote_commit message, the coordinator decided to commit the transaction or abort the transaction and it will inform the participant about the outcome of the transaction.

Participant: The Participant follows the coordinator's command and it will acknowledge the coordinator.

However 2PC protocol having less communication overhead and less expensive, it has a main drawbacks is blocking problem. The Blocking problem is described with the given circumstances is, if the coordinator fails to operate and at the same time some participants has confirmed itself to commit state. In this circumstance to end the blocked transaction, the participant should wait until the coordinator gets recovery. The blocked transactions continue to keep all the resources until they obtain the final decision from the coordinator after its recovery.

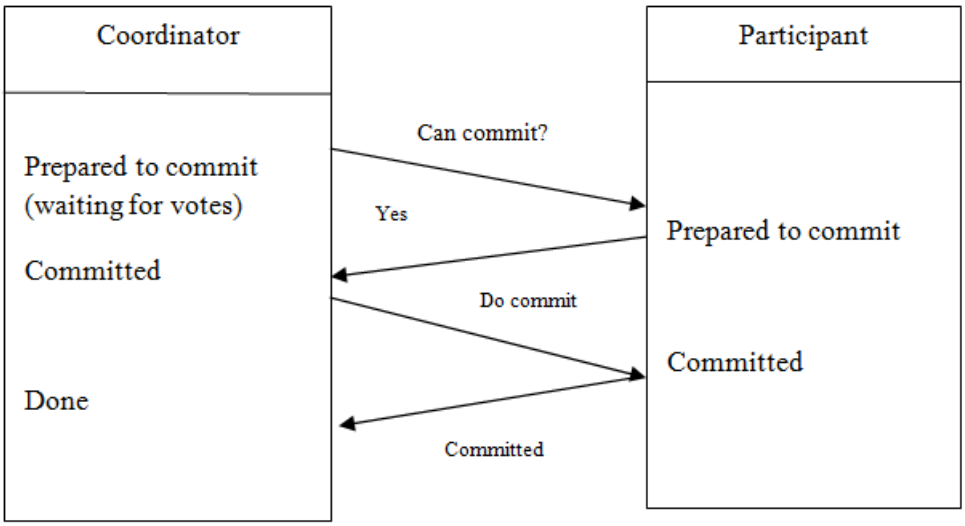

Figure1. Two Phase Commit Protocol

B. Three Phase Commit Protocol (3PC)

Three Phase Commit protocol (3PC)or non-blocking protocols which eliminate the blocking problem faced by 2PC protocol. Three Phase Commit protocol operations is similar with the two phase commit protocol, only difference is it has extra phase called Pre_commit phase where it takes the preliminary decision.

The Three Phase Commit protocol (3PC) performs the operation in three phases are Prepare phase, Pre-commit phase, Commit/Abort phase. Among the three phases Pre-Commit phase which eliminates the blocking problem.

1) First phase (Prepare)

Coordinator: Initially the coordinator will broadcast the Begin_commit request message to all participants and enters into wait state.

Participant: When the participant receive the request message, If the participant want to commit the transaction means it respond with the Vote_commit message to the coordinator and enters into ready state. Otherwise, the participant responds with the Vote_abort message to the coordinator.

Coordinator: When the coordinator receives the reply from participant it starts 2nd phase.

2) Second phase (Pre-Commit or Buffering)

Coordinator: When the coordinator receives Vote_commit message within the time from the participant, the coordinator broadcast the Pre-Commit message to all participants (At this phase preliminary decision can be made) and it moves to prepared state.

Participant: When the participant accept the Pre_commit message it will send acknowledge message the coordinator.

Coordinator: When the Coordinator receive ACK message from participant it starts 3rd phase.

3) Third phase (Commit/Abort phase)

Coordinator: The coordinator decided to commit the transaction or abort the transaction and it will inform the participant about the outcome of the transaction.

Even though the three Phase Commit protocol eliminates the blocking problem it has some disadvantage such as complicated to implement, having more communication overhead, maintains inconsistency towards network partitioning.

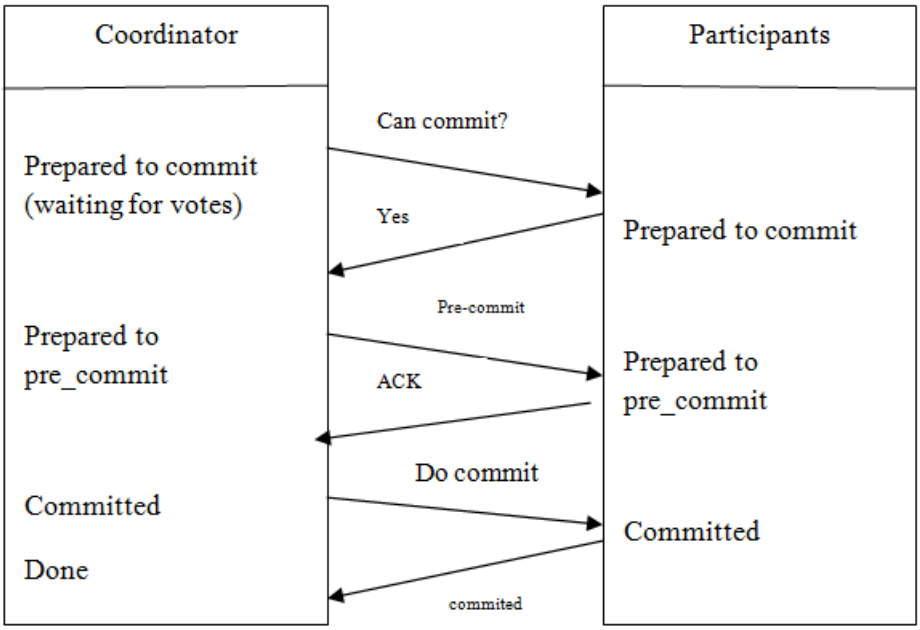

Figure2. Three Phase Commit Protocol

C. Enhanced Three Phase Commit Protocol (E3PC)

The Enhanced Three Phase Commit protocol **(**E3PC) is enhanced from quorum based 3PC. The quorum based 3PC allows a quorum to make growth in case of single failure. If the failure cascades the quorum gets lost even if we connect the quorum later it still remains blocked. The E3PC will eliminate some of the negative aspect faced by 2PC, 3PC such as blocking problem, inconsistency towards the network partitioning, more communication overhead.

E3PC recovery system includes, at every invocation the site try to elect new coordinator if the existing site is subjected to be byzantine fault. Thus E3PC reduces the overhead of View change Algorithm practice. Here we described the outline of view change algorithm with the following circumstances when a faulty primary replica broadcast contradictory message to different replica where the view change algorithm is used and the backup replica will begin this view change algorithm. This algorithm is used to elect a new primary replica when the existing primary is subjected to be Byzantine faulty. The E3PC protocol achieves higher availability of data than 3PC by maintaining two extra counters.

Pre_Elected:- state its Initial value is 0, It denotes the number of election took part in the site and the variable value is updated when new coordinator is elected.

Pre_Attempt:- state its Initial value is 0. It denotes the Election number in previous election.

1) First phase

Consider that the primary coordinator gets failed, in such case elect a new coordinator.

Coordinator: Initially the coordinator gets the value of Pre_Elected and Pre_Attempt from all sites. From that value the coordinator determines the Tot_Elected and Tot_Attempt value and sets Pre_Elected value to Tot_Elected+1 and broadcast the Tot_Elected value to all Participants.

Participants: After receiving the Tot_Elected value from coordinator, the participant sets Pre_Elected value to Tot_Elected+1and informs to coordinator.

2) *Second phase*

Coordinator: After receiving value from all participants the coordinator makes the decision according to decision rule.

TABLE I
DECISION TABLE

Collected States	Decision
∃ABORTED ∃COMMITTED Tot_Attempt_Committable (true) Tot_Attempt_Committable (False) Otherwise	ABORT COMMIT PRE-COMMIT PRE-ABORT BLOCK

where Tot_Attempt_commitable is true iff all the members of the quorum in non-final state and Pre_Attempt is equal to Tot_Attempt. Upon the decision the coordinator sets the Pre_Attempt value to Pre_Elected value and it broadcast its value to all participants.

Participants: Upon receiving the decision (Pre_Abort or Pre_Commit) from coordinator, the participant sets the value Pre_Attempt to Pre_Elected ans send acknowledgement to coordinator.

3) *Third phase*

Coordinator: Upon receiving the acknowledgement, the coordinator multicast its decision to participant.

Participant: Upon receiving decision message from coordinator, it processes the transaction.

However The E3PC protocol eliminates some drawback such as blocking problem, maintaining consistency over network partitioning it has some issues to be resolved are complex to implement, and having communication overhead.

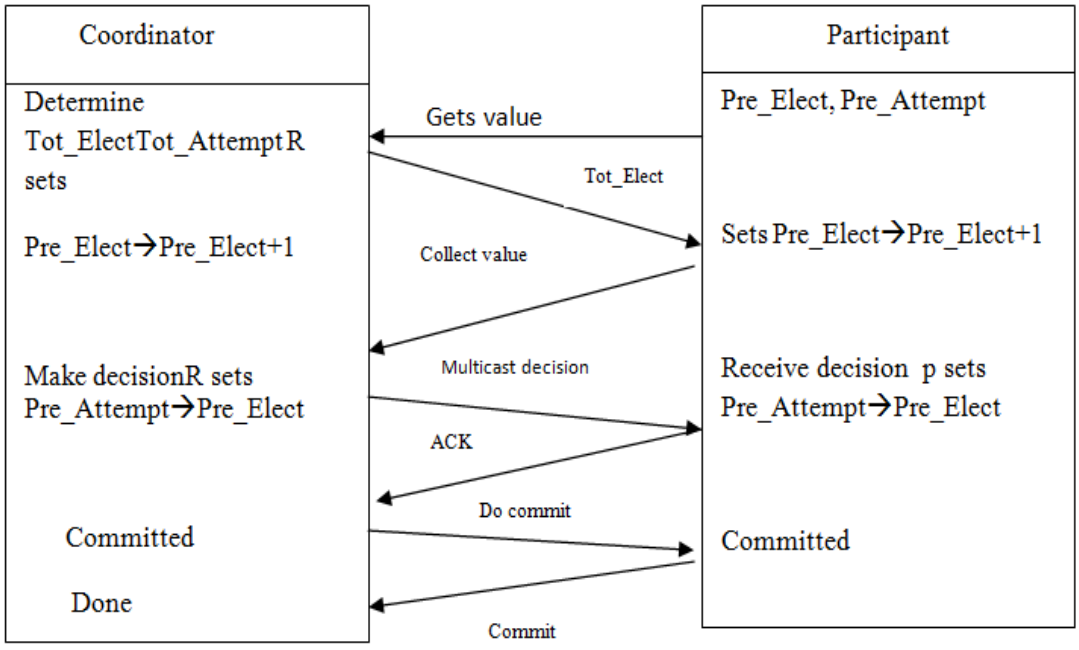

Figure3. Enhanced Three Phase Commit Protocol

III. PROPOSED WORK

Our proposed Backup Two phase commit (B2PC) protocol overcomes the major issue faced by the atomic Two Phase Commit protocol is resource blocking problem and this problem can be overcome by adding a backup phase in normal 2 Phase Commit protocol. B2PC protocol reduces the Byzantine Agreement usage because using Byzantine Agreement in each and every phase of operation is impossible and it could be prohibitively very expensive. In this protocol, a backup site is employed for each coordinator. After receiving notification from all participants in the first phase, its starts the 2nd

phase of backup phase in which the coordinator will communicates its final decision to the backup site. Afterwards, it broadcast it's the final decision to all participants.

1. *First phase*

   Coordinator: Initially the coordinator will broadcast the Begin_commit request message to all participants and enters into wait state.

   Participant: When the participant receive the request message, If the participant want to commit the transaction means it respond with the Vote_commit message to the coordinator and enters into ready state. Otherwise, the participant responds with the Vote_abort message to the coordinator.

   Coordinator: When the coordinator receives the reply from participant it starts 2nd phase.

2. *Second phase*

   Coordinator: when the coordinator receives Vote_commit message , it writes the Decided_to_commit message to its corresponding backup site. Otherwise if the coordinator receives Vote_abort message from the participant, its send the global abort message to all participants.

   Backup site: After receiving Decided_to_commit message from the coordinator, the backup site send the Recorded_commit message to corresponding coordinator.

3. *Third phase*

   Coordinator: After receiving Decided_to_commit message from the backup site, the coordinator broadcast the global commit message to all participants.

   Participants: The Participant follows the coordinator's command and it will acknowledge the coordinator.

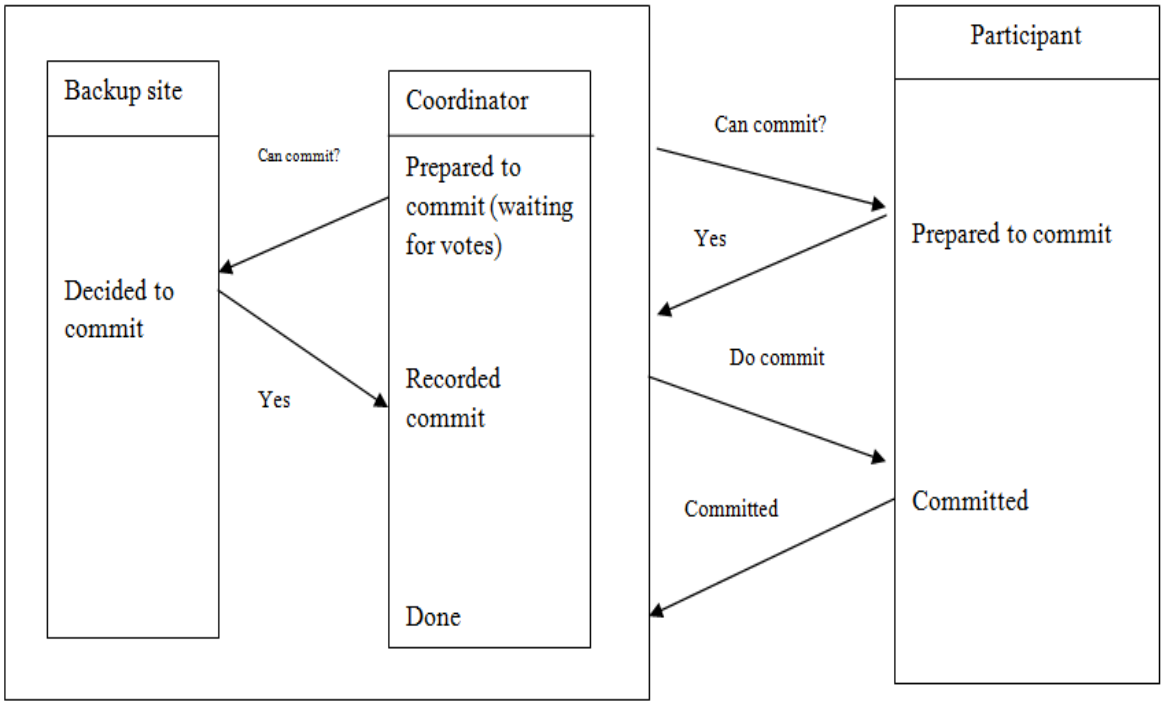

Figure4. Backup Two Phase Commit protocol

h

IV. CORE SERVICES

Our proposed Byzantine Fault Tolerance Technique which comprises a set of protocols, mechanism and some core services such as Activation service, Registration service, Completion and Coordination service are incorporated into a framework. According to Web Service Atomic Specification, the coordinator provides a set of services to Participants and to the Initiator are Registration service, Activation service, Completion service and coordination service. By making core service as harden, the coordinator can be trustworthy even in the untrusted environment of the Internet.

a. Registration service

The Registration service allows the Participants and the Initiator to get register their endpoint and it has no inter replica communication since it has no message to exchange between the replicas. For example, If a new user wants to create a new account means at the time the bank (initiator) send the request message to the registration service ,the registration service validate all information and response back to bank and bank will response back to user.

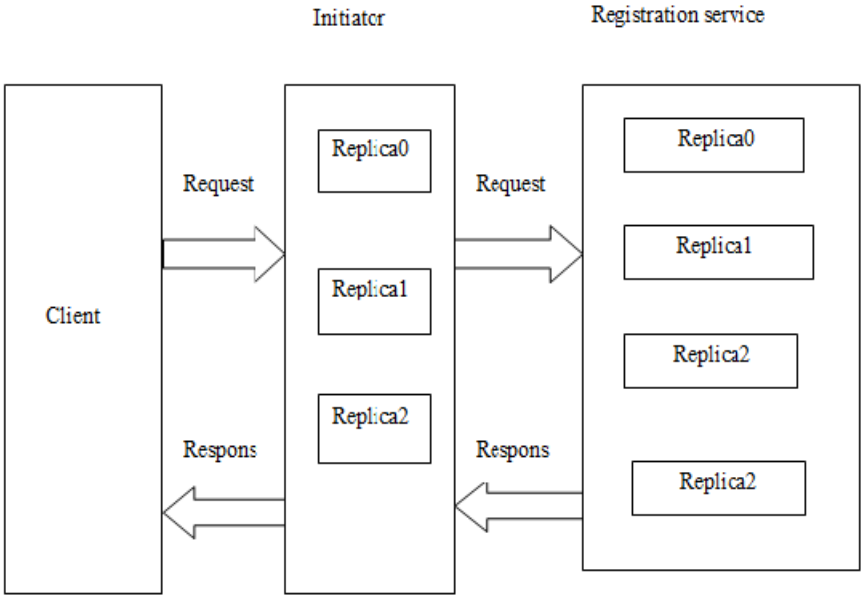

Figure 5. Registration service

b. Activation service:

Activation Service creates coordination object and unique transaction id where Transaction id is a component of coordination object. In order to preserve security transaction id can be selected from random number so that it cannot be guessed by other backup replicas. The Activation service can allow the inter replica communication it allows to create the unique transaction id. In this paper, we run the Practical Byzantine Fault Tolerance (PBFT) Algorithm in the Byzantine Agreement phase in the implementation .For example, when we requesting for any transaction operation to be done at that time the activation service can activate the required operation.

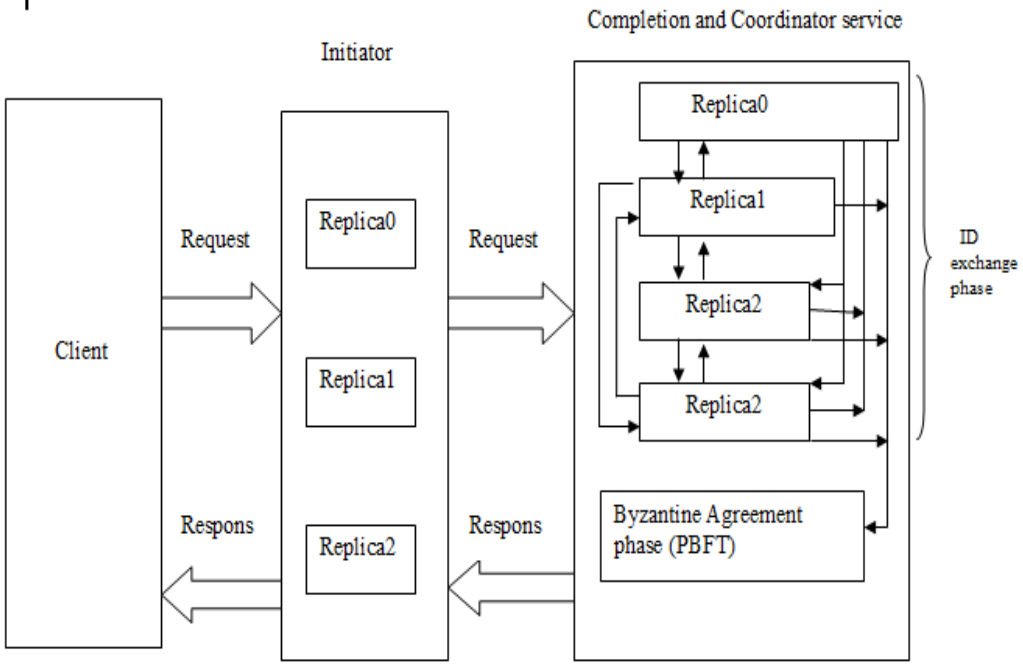

Figure6. Activation service

c. Completion and Coordination service

Completion service can turn on a successful commits operation alone. For example If a user want to credit an amount at the moment it send the request message to the coordinator service, the coordinator will validate the account details and reply back to the completion service then the completion service send back the response to bank service and bank will response to the user. Coordination service perform if any service is fault means it will redirect the request to other replica.

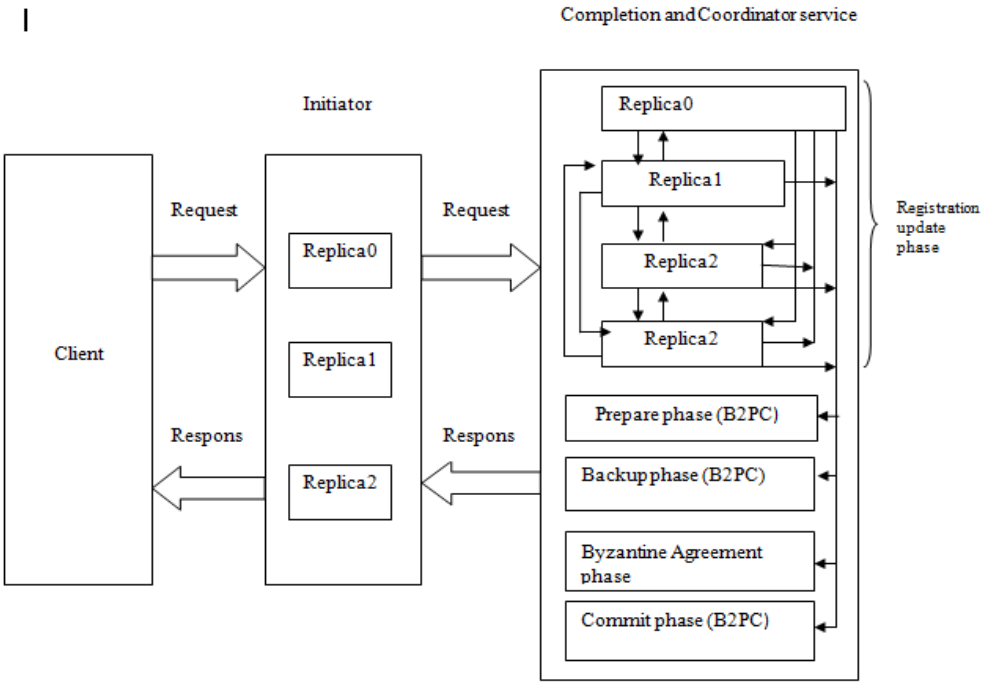

Figure7.Completion and Coordination service

V. CONCLUSION

In this paper, we proposed the Byzantine Fault Tolerance technique using Backup Two Phase Commit protocol (B2PC) which reduces the Byzantine Agreement usage since it is very expensive. Our B2PC protocol can eliminate blocking problem faced by the atomic Two Phase Commit protocol by adding an extra hardware (i.e. backup site) and it can be easily implemented in normal 2 Phase Commit protocol. According to WS_AT specification, the coordinator of transaction provides a set of core services to initiator and to participant, if we make harden this services the transaction can be trustworthy even in untrusted situation over internet. Thus our proposed BFT Technique can achieves high degree of availability, reliability, security and also less expensive.

REFERENCES

[1] M.Castro and B. Liskov, "Practical Byzantine Fault Tolerance and Proactive Recovery,"*ACM Trans. Computer Systems,vol.20, no. 4,pp. 398-461,Nov.2002.*

[2] Y, Amir, B.A. Coan, J.Kirsch, and J. Lane," Byzantine Replication under Attack," *Proc. IEEE Int'l Conf. Dependable Systems and Networks, pp. 105-144, June 2008.*

[3] C.Mohan, B.Lindsay and R.Obermark, ''Transaction management in the R*distributed database management system'', ACM Transactions on Database Systems, vol.11, no.4, 1994

[4] L. Lamport, R. Shostak, and M. Pease, "The Byzantine Generals Problem," ACM Trans. Programming languages and Systems, vol. 4, no. 3,pp. 382-401, July 1982.

[5] J. Cowling, D. Myers, B. Liskov, R. Rodrigues and L. Shri," HQ Replication: A Hybrid Quorum Protocol for Byzantine Fault Tolerance," *Proc. Seventh Symp. Operating Systems and Implementation, pp. 177-190, Nov. 2006.*

[6] P.Spiro, A.Joshi, and T.K.Rangarajan, ''Designing an optimized transaction commit protocol'', Digital Technical Journal, 1991.

[7] R. Kotla, L. Alvisi, M. Dahlin, A.Clement, and E. Wong,"Zyzzyva: Speculative Byzantine Fault Tolerance," *Proc. 21st ACM Symp. Operating Systems Principles,pp. 45-58,Oct.2007.*

[8] IditKeidar, Danny Dolev," Increasing the Resilience of Distributed and Replicated Database Systems*", J. Comut. Systems Science, December 1998*.

[9] Wenbing Zhao,"Byzantine Fault Tolerant Distributed Commit Protocol," *Dependable, Autonomic and Secure Computing, 2007. DASC 2007.*

[10] Honglei Zhang, Hua Chai, Wenbing Zhao, P.M.Melliar-Smith, L.E. Moser, "Trustworthy Coordination of Web Services Atomic Transaction", IEEE Transactions on Volume: 23 , Issue: 8

[11] P.Krishna Reddy and Masaru Kitsuregawa,"Reducing the blocking in Two-phase commit protocol employing backup sites.

[12] J.Stamos and F.Cristian, ''Coordinator log transaction execution protocol'', Journal of Distributed and Parallel Databases, 1993, pp.383-408

[13] D.Skeen, ''Nonblocking commit protocols'', ACM SIGMOD, June 1981.

[14] D.Skeen, ''A quorum-based commit protocol'', in proc. of 6th Berkeley Workshop on Distributed Data Management and Computer Networks, February 1982, pp. 69-80.

