A Study of Coordinated Check Pointing in Mobile Environment

Amit Chaturvedi, Ph.D
Head, MCA Deptt,,
Govt Engg. College, Ajmer
Rajasthan, India

Kapila Pareek
Lecturer
Sophia Girls College,Ajmer
Rajasthan, India

ABSTRACT

Mobile networks have become immensely popular. There are a few limitations in case of such networks like battery backup nodes being mobile and lack of fixed infrastructure. Checkpointing is essential because it takes care of fault tolerane and system performance in case of mobility of nodes. Uncoordinated checkpointing is when any process can take a checkpoint independent of other processes. In case of coordinated checkpoint synchronization between processes helps in assumimg a global checkpoint. The overheads in this case need to be taken care of .The various measures for the same are reducing the number of synchronizing processes and piggybacking.

In this paper, we have analyzed various coordinated checkpointing algorithms and found that minimum process algorithm is more suitable than others because it reduces the overheads of storing checkpoints as no useless checkpoints are created and not all processes are blocked.

Keywords

mobile networks, coordinated checkpointing, uncoordinated checkpointing, Non-blocking, MSS, MN

1. INTRODUCTION

A mobile system is a distributed system where in a collection of processes geographically separated communicate with each other by exchanging messages[1][2][3][4].These processes are running on mobile hosts(MH) and communicate with mobile support stations(MSS) by wireless networks. The MSS are connected to other MSS by wired networks. Due to the mobility of MHs and the constraints of wireless networks like low bandwidth of wireless network, lack of stable storage, limited battery life of MN, frequent disconnection and limited range of wireless networks[5][6][7][2]the design of checkpointing is difficult. Checkpoint is actually a snapshot of the current state of a process which saves information in non-volatile stable storage. If an error occurs then process could be resumed from the current checkpoint rather than starting process from the start. It is a recovery mechanism and a fault tolerant technique which helps the self controlling and self manageable mobile networks without requiring additional efforts from the programmer.

In case of a failure, the system rolls back to a consistent set of checkpoints. Now the checkpoints could be decided in two major ways: If all the processes take checkpoints at the same time then the set of checkpoints would be consistent. But this is difficult so a time interval is given to the processes wherein they can take their checkpoint. Processes may take temporary checkpoints to synchronize themselves with other processes which are made permanent, when all processes agree. Total checkpointing time or TCT is the time from the initiation of checkpointing to the time the last process takes its checkpoint (may be temporary)). Checkpointing latency is the time from initiation to when all checkpoints made permanent this signifies the finishing of checkpointing process. After failure a set of checkpoints, with one checkpoint for every process, is said to be consistent global checkpointing state (CGS) .A CGS will not have orphan, lost or duplicate processes. Orphan Messages are messages whose reception has been recorded, but the record of their transmission has been lost. This situation arises when the sender node rolls back to a state prior to sending the message while the receiver node still has the record of its reception. Lost Messages are messages whose transmission has been recorded, but the record of their reception has been lost. This happens if the receiver rolls back to a state prior to the reception of the message, while the sender does not roll back to a state prior to their sending. Duplicate Messages happens when more than one copy of the same message arrives at a node; perhaps one corresponding to the original and other generated during recovery phase.

Checkpointing is divided under two broad categories: (a) coordinated and (b) uncoordinated. In uncoordinated checkpointing each process takes checkpoints independently, without bothering about other processes [8] [9] [10]. In case of a failure, after recovery, a CGS is found among the existing checkpoints and the system restarts from there. Thus finding a CGS is quite tricky. The choice of checkpoints depends on the manner in which these processes are dependent on each other mutually. The common approach is to use rollback dependent graph or checkpoint graph [8][12][14][1][15]. In case of coordinated checkpointing [12][16][18][19][9] [1] [21] all processes have to synchronize through control messages and piggybacked information before taking checkpoints. Both the uncoordinated and coordinated algorithms have their shortcomings. In case of uncoordinated algorithms some useless CGS may be created. Also these algorithms suffer from domino effect In which case a rollback propagation occurs, consider the situation where the sender of a message P rolls back to a state that lies before the sending of P .Now to maintain the consistency in the system the receiver of P must also roll back to a state that precedes P' receipt; otherwise, the states of the two processes would be inconsistent as it would mean a message P received without being sent. This phenomenon of cascading rollbacks is called the domino effect.

In case of coordinated algorithms the processes need to send messages which add to the overheads. In this paper we analyze the various algorithms proposed for coordinated checkpointing.

2. RELATED WORK

Related work of this field have been covered on the topics blocking coordinated checkpointing, Non-Blocking Coordinated checkpointing algorithm, All-process coordinated checkpointing algorithms, and Minimum-process algorithms.

2.1 Blocking coordinated checkpointing

In case of blocking algorithms, the processes need to stop their computation during synchronization to prevent orphan messages. They remain blocked until the entire checkpointing activity is complete. In this case a coordinator takes a checkpoint and broadcasts a request message to all processes, asking them to take a checkpoint. When a process receives the message, it stops its executions, flushes all the communication channels, takes a tentative checkpoint, and sends an acknowledgement message back to the coordinator. After the coordinator receives acknowledgements from all processes, it broadcasts a commit message. On receiving commit, a process converts its tentative checkpoint into permanent one and discards its old permanent checkpoint, if any. The process is then free to resume execution and exchange messages with other processes.

These algorithms force all relevant processes in the system to stop their computation during synchronizing and hence degrade system performance [18] [23].

Dang and Park [24] proposed an algorithm to address both orphan messages and lost messages. Kim and Park [25] proposed an improved scheme to address failures during checkpointing. It allows the new checkpoints in some subtrees to be committed. In the approach, a process commits its tentative checkpoint if none of the processes, on which it transitively depends, fails; and the consistent recovery line is advanced for those processes that committed their checkpoints. The initiator and other processes which transitively depend on the failed process have to abort.

Leu and Bhargava,[26] proposed an algorithm which is resilient to multiple process failures and does not assume that the channel is FIFO. The algorithm does not consider lost messages in checkpointing and recovery and assumes a sliding window kind of scheme to deal with message loss problem.

Li and Shu [28] designed an algorithm to reduce blocking time for checkpointing operation, in which each process maintains a set of processes .A process is included in this set if it has sent at least one message to the process in current checkpoint interval. Checkpointing dependency information is transferred from sending process to destination process during normal message transmission. So when a process starts a checkpointing procedure it knows in advance the processes on which it depends.

The useless checkpoints and blocking of processes during checkpointing were reduced by a synchronous checkpointing protocol for mobile distributed systems proposed by Kumar and Kumar [46]. A process takes an induced checkpoint if the probability that it will get a checkpoint request in current initiation is high.

Biswas and Neogy [47] gave a blocking coordinated scheme which each MSSp is required to maintain an array A[n] where n is the number of mobile hosts from 0 to n-1. A [1] is 1 when MH1 is near MSSp . A MH calculates its dependency vector D initiates checkpointing procedure and sends request to all the MH who have bit 1 in dependency vector D via its MSS. If such a MH is present in vicinity of current MSS, then checkpoint request is send directly to MH. Else this MSS will broadcast checkpoint request message to other MSS which reaches all those processes whose bit is 1 in dependency vector D. Thus all these processes take checkpoint and sends information to initiator via their local MSS.

Lotfi, Motamedi and Bandarabadi proposed a two-level blocking checkpointing algorithms [27] in which local and global checkpoint are taken. Nodes take local checkpoint according to checkpoint interval calculated previously based on failure rate and save it in their local disk.

These checkpoints when sent to stable storage become global checkpoint. Local checkpoints are used to recover from more probable failures where as global checkpoints are used to recover from less probable failures. If the time taken to recover in case of global checkpoint being taken is less than the time taken if global checkpoint is not taken system stores global checkpoint else it would take local checkpoint.

Biswas & Neogy [2] proposed a checkpointing and failure recovery algorithm where mobile hosts save checkpoints based on mobility and movement patterns. Mobile hosts save checkpoints when number of hand-offs exceed a predefined handoff threshold value. They introduced the concept of migration checkpoint An MH upon saving migration checkpoint, sends it attached with migration message to its current MSS before disconnection.

Kumar ,and Garg [29] and Kumar [30] gave the concept of hybrid checkpointing algorithm, where in an all-process coordinated checkpoint is taken after the execution of coordinated checkpointing..

Synchronising using loosely synchronized clocks are given by Cristian and Jahanian [48], Neves and Fuchs [50], [Ramanathan and Shin], [52].

Ssu et al. [49] also gave a loosely synchronized coordinated checkpointing protocol that removes the overhead of synchronization. This approach assumes that the clocks at the processes are loosely synchronized which can trigger the local checkpoints at all the processes roughly at the same time without a coordinator.

2.2 Non-Blocking Coordinated checkpointing algorithm

In non-blocking algorithms, no blocking of processes is required for checkpointing [26][18]. In this approach the processes need not stop their execution while taking checkpoints. But the inherent problem here is coordinated checkpointing is to prevent a process from receiving application messages that could make the checkpoint inconsistent [12][16][1][22].

Cao and Singhal [13] presented a non-blocking coordinated checkpointing algorithm with the concept of "Mutable Checkpoint" which can either be converted to temporary checkpoint or discarded later and can also be saved either on the main memory or local disk of MHs. In this scheme only dependent processes are forced to take checkpoints. Thus the advantage of this scheme is that taking mutable checkpoint avoids the overhead of transferring large amounts of data to the stable storage at MSSs over the wireless network.

An algorithm for mobile computing systems producing consistent set of checkpoints without the overhead of temporary checkpoints was given by Bidyut, Rahimi and Liu [54].This algorithm presented a single phase non-blocking coordinated checkpointing.

Cao, Zhang and He [53] proposed an algorithm for Hybrid Systems integrating independent and coordinated checkpointing for application running on a hybrid distributed system.

Singh [38]] proposed a non blocking algorithm in which a predefined checkpoint interval *T* is set on timers of all the MHs which is a deadline to take next checkpoint if process has sent any message in current checkpoint interval.

Mandal and Mukhopadhyaya [35] proposed a non blocking algorithm that uses the concept of mobile agent to handle multiple initiations of checkpointing. Mobile Agent has the same id same as its initiator and it migrates among processes, take actions and then moves to other node together with required information. Each process takes initial permanent checkpoint and sets version number of checkpoint to 0. Process sends application message m by piggybacking it with version number of its latest checkpoint. Receiver compares application message's version number with its own current checkpoint version number to decide whether to take checkpoint first or simply to process message only. There is a DFS which is maintained by each process which contains id of neighbors on which the process depends.

Continuing with their work for non-blocking checkpointing and recovery algorithms Bidyut, Rahimi and Liu [34] proposed an algorithm for bidirectional ring networks. The proposed algorithm allowed the process to take permanent checkpoints directly, without taking temporary checkpoint global snapshot algorithms for large scale distributed systems. Whenever a process is busy it takes a checkpoint after completing its current procedure.

A single phase non blocking coordinated algorithm proposed a single-phase non-blocking coordinated checkpointing algorithm suitable for mobile computing environments was given by Kumar, Chauhan and Kumar [36] in which processes take permanent checkpoints directly without taking temporary checkpoints and whenever a process is busy, the process takes a checkpoint after completing the current procedure.

Subha Rao and Naidu [37] introduced an algorithm for handling orphan messages and lost messages. Active interval (AI) is the time that elapses between the events of sending messages" prepare checkpoint" and "take checkpoint" by the initiator to all the processes. Message is said to be lost if it is sent in active interval of a process but received after active interval of the process or is not received at all. A message becomes an orphan message if it is sent by sender after its active interval and received by receiver before or in the active interval. Number of messages that are lost can be determined so that they can be replayed from sender side log. Orphan messages are handled by allowing receiver to maintain for each message in its latest checkpoint a count of number of messages received by the process until the last checkpoint. If sender tries to replay any message whose SSN is less than or equal to SSN of receiver, receiver discard it as orphan message.

2.3 All-process coordinated checkpointing algorithms

Every process is required to take its checkpoint in an initiation. Chandy and Lamport algorithm is one of the earliest nonblocking all-process coordinated checkpointing algorithm for static nodes but works with FIFO channels only[16]. Elnozahy,Johnson.and Zwaenepoel proposed an algorithm which uses use checkpoint sequence numbers to identify orphan messages, thus avoiding the need for processes to be blocked during checkpointing and allows piggybacking of integer CSN (checkpoint sequence number) along with normal messages[14].

2.4 Minimum-process algorithms

These algorithms have minimum interacting processes .These processes are required to take their checkpoints in an initiation period. Prakash and Singhal [1] were first to give minimum-process nonintrusive coordinated checkpointing protocol for mobile distributed systems. But their algorithm did not handle inconsistencies.Koo and Toueg [18] proposed a minimum-process coordinated checkpointing protocol which relaxes the assumption that all communications are atomic. It reduces the number of synchronization messages and minimize the number of processes to checkpoints They further added that if any of the relevant process is not able to take its checkpoint in an initiation, then the entire checkpointing process of that particular initiation is aborted Kim and Park [25] proposed an improved scheme to address failures during checkpointing. It allows the new checkpoints in some subtrees to be committed

Thus, in case of a node failure during checkpointing; total abort of the checkpointing is avoided .In the algorithm proposed by Silva and Silva [22], the processes which did not communicate with others during the previous checkpointing interval did not need to take new Checkpoints. Kumar and Khunteta [17] proposed an algorithm for deterministic mobile distributed systems The features of the proposed algorithm were that no blocking of processes takes place, no useless checkpoints are created and anti-messages of very few messages are logged .Hence the loss during checkpointing reduces when a process fails to coordinate with other processes.

In case of cluster based multi channel ad hoc wireless systems ,a fast checkpointing and recovery algorithm with low overheads was given by Chaoguan-Zhenpeng Xiang algorithm[31].Here a cluster head uses a beacon packet to manage cluster for information like In this algorithm the checkpointing clock data, sizes of announcement traffic indication message and data window. It does so by channel assignment, scheduling intra cluster traffic, and communicating data..It also maintains system variables like a checkpoint index, an ordinary node queue, and a variable storing the number of reply messages. The head delivers a request for checkpointing to all nodes together with all parameters .After the receipt of the checkpoint request, nodes busy communicating will get the required checkpoint during next beacon and new checkpoints will be given to those nodes which are not allowed to communicate will take a new checkpoint so the checkpointing process is completed within two consecutive beacon intervals and then start rollback recovery in one beacon interval. Talking on the same lines a Concurrent checkpointing and recovery was proposed by Singh and Jaggi [39]. This algorithm discussed an approach causing events usually happing at the same time to happen at different time. This eliminates contention for resources. The main features of this algorithm are that there is no need for FIFO channels, only minimum numbers of messages are logged and overlapping failures are handled efficiently.

In case of The Minimum Process Coordinated Checkpointing Scheme proposed by Tuli and Kumar [41] A base station saves routing and data information of cluster head .If this CH fails then the fault is detected by the base station which makes a new cluster head .the base station on receiving checkpoint request receives a checkpoint and then on receiving commit it makes this checkpoint as permanent. . The important feature of this algorithm is that only participating hosts need to receive checkpoints.Saluja and Kumar [42] proposed a minimum process checkpointing scheme on cluster based routing protocol which reduces the number of useless checkpoints. Tuli and Kumar [45] introduced the concept of “Soft checkpoint”. Soft checkpoints can be saved anywhere, e.g., the main memory or local disk of MHs. Before disconnecting from the MSS, these soft checkpoints are converted to hard checkpoints and are sent to MSSs stable storage. In this way, taking a soft checkpoint avoids the overhead of transferring large amounts of data to the stable storage at MSSs over the wireless network. We have also shown that our soft checkpointing scheme also adapts its behaviour to the characteristics of network.

3. SYSTEM MODEL

The system model of a mobile computing system consists of a set of mobile hosts (MHs) and mobile support stations (MSSs). The static MSS provides various services to support the MHs and a region covered by a MSS is called a cell. A wireless communication link is established between a MH and a MSS; and a high speed wired communication link is assumed between any two MSSs. The wireless links support FIFO communication in both directions between a MSS and the MHs in the cell wired links. A distributed computation is performed by a set of MHs or MSSs in the network.

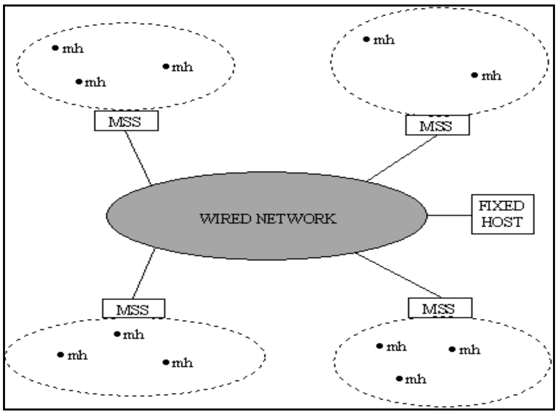

Fig 1: Mobile network

4. FINDINGS

Coordinated checkpointing avoids domino effects and minimizes the stable storage requirement hence it seems better than asynchronous checkpointing. It could be further blocked or unblocked .Blocking degrades the system performance and causes overheads .These overheads could be reduced by non blocking and further if only minimum number of processes take part in checkpointing. Checkpoints could be temporary and after commit happens these are converted to permanent and stored in stable storage. This reduces overheads of storing checkpoints the overheads for synchronizing messages could further be reduced by piggybacking.

5. CONCLUSION

Checkpointing is essential for load balancing and monitoring. Recovery after failures is the main function of checkpointing. In coordinated or synchronous checkpointing, processes take checkpoints in such a manner that the resulting global state is consistent. The Coordinated algorithms are classified as blocking, nonblocking, all process and minimum process algorithms. Mobile systems suffer from issues like lack of stable storage, mobility, frequent disconnections, finite power source, etc. We have tried to analyse various coordinated checkpointing algorithms in this paper. All robust systems need to have efficient fault tolerance capabilities and same is true for mobile systems also. The synchronization through messages adds to the overheads in case of coordinated algorithms which needs to be improved.

6. REFERENCES

[1]. Prakash, R. and Singhal, M.1996. Low Cost Checkpointing & Failure Recovery in Mobile Computing System. IEEE Transactions on Parallel & Distributed Systems(Vol. 7 No. 10), 1035–1048.

[2] Biswas, S. and Sarmistha, N. 2010. Mobility based Checkpointing Protocol for Mobile Computing System. International Journal of Computer Science & Information Technology (IJCSIT) (Vol. 2. No. 1), 135–151.

[3] Gahlan, P. and Kumar, P.2010. A Low Overhead Minimum Process Coordinated Checkpointing Algorithm for Mobile Distributed System. International Journal of Computer Applications(Vol. 3,.No. 1) 17–21.

[4] Khatri, Y. 2012. Distance based Asynchronous Recovery Approach in Mobile Computing Environment. International Journal of Distributed and Parallel Systems (IJDPS)(Vol. 3, No. 3).

[5] George, S. E., Chen I-R and Jin, Y.2006.Movement-based Checkpointing and Logging for Recovery in Mobile Computing Systems. In Proceedings of the 5th ACM International Workshop on Data Engineering for Wireless and Mobile Access,51–58.

[6] Yeom, H.Y.and Park, T. 2000.An Asynchronous Recovery Scheme based on Optimistic Message Logging for Mobile Computing Systems. 20th International Conference on Distributed Computing Systems (ICDCS 2000), 436–443.

[7]Woo, N., Park, T. and Heon Y.Y. 2002. An Efficient Optimistic Message Logging Scheme for Recoverable Mobile Computing Systems. IEEE Transactions on Mobile Computing(Vol. 1, No. 4), 265–277.

[8] Bhargava, B. and Lian, S.R..1998. Independent checkpointing and concurrent rollback for recovery in distributed systems-an optimistic approach.In Proceedings of the Seventh IEEE Symposium on Reliable Distributed System,3–12.

[9] Meth, K.Z and Tuel, W. G. 2000.Parallel heckpoint/restart without message logging.In Proceedings of the IEEE 28th International Conference

on Parallel Processing (ICPP '00), (August ,2000)253–258 .

[10] Strom, R.E.and Yemini, S. 1985. Optimistic recovery in distributed systems,ACM Trans. Comput. Systems 3 (3), 204–226.

[11] Park, T. Woo, N. and. Yeom, H.Y. 2001.An Efficient Recovery Scheme for Mobile Computing Environment. In Proceedings of 8th International Conference on Parallel & Distributed Systems (ICPADS), 53–60.

[12] Cao, G. and Singhal, M.1998. On coordinated checkpointing in distributed systems, IEEE Trans. Parallel Distrib. Systems 9 (12) ,1213–1225.

[13] Cao, G. and Singhal, M. 2001.Mutable Checkpoints: A New Checkpointing Approach for Mobile Computing Systems.IEEE Transactions on Parallel and Distributed system(vol.12, Issue 2, Feb., 2001)157-172, ISSN: 1045-9219.

[14] Elnozahy, E.N., Alvisi, L., Wang, Y. M. and Johnson, D. B..2002. A survey of rollback-recovery protocols in message-passing systems.ACM Comput. Surveys 34 (3). 375–408.

[15]. Wang, Y. M. 1997. Consistent global checkpoints that contain a given set of local checkpoints, IEEE Trans. Comput. 46 (4) (Apr. 1997) ,456–468.

[16] Chandy, K.M.,and Lamport, L. 1985. Distributed snapshots: determining global states of distributed systems, ACM Trans. Comput. Systems 3 (1) ,63–75.

[17] Khunteta, A. and Kumar,P. 2010.An Analysis of Checkpointing Algorithms for Distributed Mobile Systems", International Journal on Computer Science and Engineering (IJCSE), Vol. 02, No. 04, Pp. 1314–1326.

[18] Koo, R. and Toueg, S. 1987. Checkpointing and rollback-recovery for distributed system, IEEE Trans. Software Eng. 13 (1) 23–31.

[19]Manivannan, D. and Singhal, M. 1996. A low-overhead recovery technique using quasi-synchronous checkpointing, in: Proceedings of the IEEE Sixth International Conference on Distributed Computer Systems, (May 1996)100–107.

[20]Manivannan,D. and Singhal, M. 1999. Quasi-synchronous checkpointing :models, characterization, and classification, IEEE Trans. Parallel Distrib. Systems 10 (7) ,703–713.

[21]Vidya, N.H.1999 .Staggered consistent checkpointing, IEEE Trans. Parallel Distrib. Systems 10 (7) , 694–70.

[22] Silva, L.M. and Silva,J.G. 1992. Global checkpointing for distributed programs. In Proceedings of the 11th Symposium on Reliable Distributed Systems, 115–162.

[23]Tamir, Y. and Sequin,C.H. 1984.Error Recovery in multicomputers using global checkpoints. In Proceedings of the International Conference on Parallel Processing, 32-41.

[24] Dang, Y. and Park, E.K. 1994. Checkpointing and Rollback-Recovery Algorithms in Distributed Systems" J. Systems and Software,(Apr. 1994), pp. 59-71.

[25]Kim, L. and Park, T. 1993.An Efficient Protocol for Checkpointing Recovery in Distributed Systems.,IEEE Trans. Parallel and Distributed Systems (Aug. 1993),955-960.

[26]. Leu, P. Y. and Bhargava, B.1988. Concurrent Robust Checkpointing and Recovery in Distributed Systems.Proc. Fourth IEEE Int'l. Conf. Data Eng., 154-163.

[27] Lotfi, M., Motamedi S.A. and Bandarabadi, M. 2009.Lightweight Blocking Coordinated checkpointing for Cluster Computer Systems. Symposium. On System Theory.

[28] Li, G. and Shu ,L-C..2005. A Low-Latency Checkpointing Scheme for Mobile Computing Systems .Int. Conf. Computer Software and Applications, IEEE.

[29] Kumar, P. and Garg, R.2011.Soft Checkpointing Based Hybrid Synchronous Checkpointing Protocol for Mobile Distributed Systems", International Journal of Distributed Systems and technologies,2(1) (Jan-March, 2011),1-13.

[30] Kumar, P. 2004. A Low-Cost Hybrid Coordinated Checkpointing Protocol for Mobile Distributed Systems.Mobile Information Systems, An International Journal from IOS Press, Netherlands, 13-32, Vol. 4, No. 1, 2007

[31]. Chaoguang, M., Zhenpeng, X. and Xiang, L.2008.An Efficient Checkpointing and Rollback Recovery Scheme for Cluster-based Multi-channel Ad-hoc Wireless Networks.International Symposium on Parallel and Distributed Processing with Applications 978-0-7695-3471-8/08 IEEE ,371-378.

[32]Gupta, B. ,Rahimi, S. and Liu, Z. 2006. A new high performance checkpointing approach for mobile computing systems, International Journal of Computer science and network security.

[33]Saluja, K. and Kumar, P. 2011.A Non-blocking Checkpointing Algorithm for Non-Deterministic Mobile Ad hoc Networks. International Journal of Computer& Organization Trends – Vol 1 no 2),13.

[34]. Gupta, B. ,Rahimi, S. and Liu, Z. 2008. Design of high performance distributed snapshot/ recovery algorithms for ring network, Journal of Computing and information Technology-CIT.

[35] Mandal, P. S. and Mukhopadhyay, K.2007.Mobile Agent based Checkpointing with Concurrent Initiations", International J. of Foundation of Computer Science.

[36]Surender Kumar, R.K. Chauhan and Parveen Kumar, "Minimum process Error discovery algorithm for mobile Distributed system using Global Checkpoint", International Journal of Information Technologyand Knowledge Management, Jan-June, 2008, Vol. 1, No. 1, pp 25-33

[37] Subha Rao, D. V. and Naidu, M.M .2008.A New Efficient Coordinated Checkpointing Protocol Combined with Selective Sender based Message Logging.

[38] Singh, A.K. 2007.On Mobile Checkpointing Using Index and Time Together", World Academy of Science,Engineering and Technology.

[39] Singh, A.K. and Jaggi, P.K. 2011.Staggered Checkpointing and Recovery in Cluster Based Mobile Ad Hoc Networks. International Conference on Parallel, Distributed Computing technologies and Applications (PDCTA-2011) Springer Proceedings.

[40] Kumar, P. and Chauhan, R.K. 2006.A Coordinated Checkpointing Protocol for Mobile ComputingSystems", International Journal of Information and Computing Science(Vol. 9 No. 1) 18-27.

[41] Tuli,, R. and Kumar, P. 2011. Minimum process coordinated Checkpointing scheme for ad hoc Networks, International Journal on AdHoc Networking Systems (IJANS) 9Vol. 1, No. 2)9 October 2011) ,51-63.

[42] Saluja, K. and Kumar, P. 2011.A Non-blocking Checkpointing Algorithm for Non-Deterministic Mobile Ad hoc Networks. International Journal of Computer& Organization Trends (Vol1 No2), 13.

[43]Acharya, A. and Badrinath, B.R. Checkpointing distributed applications on mobile computers.In Proceedings of the Third International Conference on Parallel and Distributed Information Systems (Austin, Texas, Sep, 1994), 73-80.

[45] Tuli,, R. and Kumar, P .2011.The Performance of Soft Chekpointing Approach in Mobile Computing Systems Global Journal of Computer Science and Technology (Vol 11 No 9 Ver 1.0) May 2011.

[46] Kumar, L. and Kumar, P. 2007.A synchronous ckeckpointing protocol for mobile distributed systems: probabilistic Approach. Int Journal of information and computer security .

[47]Biswas, S. and Neogy, S. 2007.A Low Overhead Checkpointing Scheme for Mobile Computing Systems. Int. Conf. Advances Computing and Communications, IEEE.

[48] Cristian, F. and Jahanian, F. 1991. A timestamp-based Checkpointing Protocol for Long Lived Distributed Computations", Proc IEEE Symp. Reliable Distributed Systems, 12-20.

[49] Ssu, K.F., Yao B., Fuchs, W.K. and Neves, N. F.1999.Adaptive Checkpointing with Storage Management for Mobile Environments. IEEE Transactions on Reliability (December 1999) (Vol. 48, No. 4) 315-324.

[50] Neves, N. and Fuchs, W. K.1997.Adaptive Recovery for Mobile Environments.Communications of the ACM (January 1997) (Vol. 40, No.1) ,68-74,

[51] Chauhan, R.K. ,Kumar,P. and Kumar,L. 2006. Checkpointing Distributed Applications on Mobile Computers. Journal of Multidisciplinary Engineering and Technologies(Jan 2006) (Vol. 2 No.1).

[52] Ramanathan, P. and Shin, K.G. 1993.Use of Common Time Base for Checkpointing and Rollback Recovery in a Distributed System. IEEE Trans. Software Engg (June 1993) 571-583.

[53] Cao, J., Chen,Y. Zhang, K. and He, Y. 2004. Checkpointing In Hybrid Distributed Systems,Proceedings of 7th international symposium of Parallel architetures, Algorithms and Network,IEEE, 2004.

[54]]. Gupta, B. ,Rahimi, S. and Liu, Z.2006.A new high performance checkpointing approach for mobile computing systems. International Journal of Computer science and network security.