Checkpointing and Roll back Recovery Protocols in Wireless Ad hoc Networks: A Review

JAWAHAR THAKUR, ARVIND KALIA
Department of Computer Science, H. P. University, Shimla
and
LALIT AWASTHI
National Institute of Technology, Hamirpur

The main focus in a single process checkpointing protocol is on finding optimal checkpoint interval to minimize the loss due to any fault, but in a distributed environment the main focus is on finding out and saving a global consistent state of the system. The challenge in finding a global consistent state is that interprocess communication creates dependencies that must be factored, otherwise the global checkpoint becomes useless. Mobile ad hoc networks throw up a plethora of challenges in tracking interprocess dependences including how to reliably save checkpoints in face of transience and node failures, where to save the checkpoints and how to reconstruct the stable global state from the nodes which are available after the fault. To add high availability and reliability to mobile networks, checkpoint based rollback recovery techniques are widely applicable. Checkpointing methods for traditional distributed systems cannot be applied directly to the mobile networks. This paper provides an overview of the available checkpointing strategies for mobile networks, comparing them on the various parameters. We conclude that no single strategy is optimal in all fault scenarios and that the perfect strategy may still be in the works.

Keywords: mobile networks, checkpointing, rollback recovery

1. INTRODUCTION

In Mobile Networks the computing device and the communication link are both susceptible to failure prompting the deployment of fault tolerance procedures for fast recovery. Rollback recovery mechanisms [give reference] based on checkpointing [give reference] are widely employed to effectively recover from faults in distributed systems. Haerder and Reuter's [add reference] define a checkpoint as a "collection of information in a safe place, which has the effect of defining and limiting the amount of REDO recovery required after a crash," Chekpointing is the process of saving the state of a process on stable storage and a checkpoint is typically defined as a designated place in a program at which normal processing is interrupted specifically to preserve the status information necessary to allow resumption of processing at a later time. Checkpointing and rollback recovery is popular and most widely used techniques for fault tolerance in wireless and mobile distributed computing system. [[Park et al. 2003], [Prakash and Singhal 1996b], [Quaglia et al. 2006], [Men et al. 2007]].

Mobile ad hoc networks are challenging in terms of designing effective and viable strategies for fault tolerance primarily due to its infrastructure-less operation and temporary topology due to node transience, but also due to:

- dynamically changing topologies or routes,
- variable mobility pattern of nodes,
- variable link capacity,
- packet loss due to error in transmission,
- disconnection/partition/join are frequent,
- unpredictable synchronous and asynchronous links,

- communication bandwidth is limited,
- broadcast communication,
- limited battery lifetime,
- no stable storage,
- no centralized control,
- limited capacities of system,
- lack of mobility awareness by system and mobile node and
- prone to hardware, software failure and physical security threats.

Checkpointing-based rollback recovery restores the system state to the most recent checkpoint whenever failure occurs [Randill 1975]. Figure 1 provides a classification of rollback recovery strategies. Roll back recovery treats a distributed system as a collection of process that communicates over a network. It achieves fault tolerance by periodically saving either the state of the process or its control message exchanges during failure free execution. This enables the process to be restarted from the saved state to reduce the amount of lost work due to failure. The set of saved states or messages are called checkpoints.

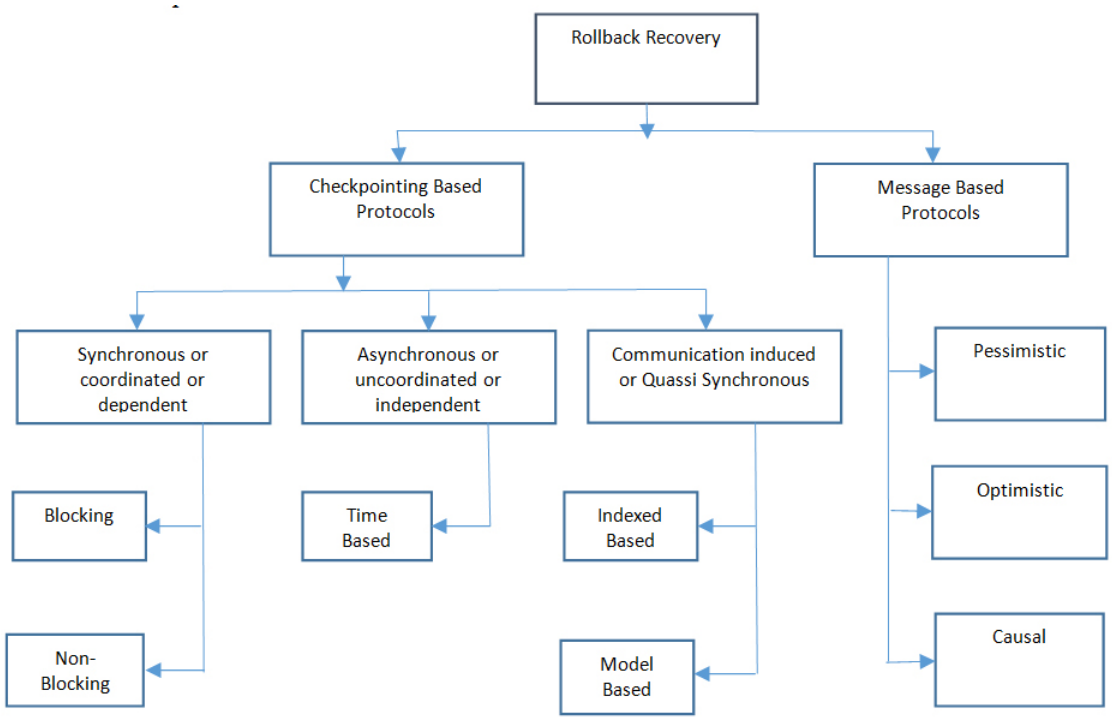


Figure 1: Classification of Rollback Recovery Schemes

Existing rollback recovery protocols has their pros and cons which are summarized in Table 1.

The objective of this paper is to present a review of the checkpointing and rollback recovery strategies in the mobile distributed computing networks and ad-hoc networks based on the various attributes.

The rest of the paper is organized as follows: in section II review of literature is carried out, in section III comparison and analysis of existing schemes is presented and finally the paper is concluded in section IV with possible future developments.

S.No.	Rollback Recovery Techniques	Pros	Cons
01	Synchronous or co-ordinated or Dependent Checkpointing [[Awasthi and Kumar 2007b],[Cao and Singhal 2001],[Prakash and Singhal 1996a],[Elnozahy et al. 2002]]	• Only one checkpoint per process. • No coordination required during recovery process. • No domino effect.	• Synchronization overhead due to message. • Blocking nature.
02	Asynchronous or uncoordinated or independent [Elnozahy et al. 2002]	• No Synchronization message is required. • Non-blocking nature. • Autonomy is maximum.	• Domino effect. • Avalanche-effect. • Useless checkpoint taken. • Overhead due to garbage collection. • Overhead due to recovery line computation. • More than one checkpoint per process.
03	Communication induced or Quassi-Synchronous checkpointing [[Elnozahy et al. 2002],[Acharya and Badrinath 1994]]	• No synchronization message. • Non-blocking nature. • Domino-free.	• Governed by communication pattern. • Forced checkpoint.
04	Hybrid checkpointing [Elnozahy et al. 2002]	• It takes advantage of both synchronous and asynchronous checkpointing in consideration.	• Process and recovery overheads due to hybrid approach.
05	Log based [Elnozahy et al. 2002]	• No synchronization. • Roll forward possible. • Can handle outside world interaction.	• Garbage collection overhead. • Message logging overhead.
06	Log-based [Elnozahy et al. 2002] (Pessimistic)	• No orphan message. • Only one checkpoint per process.	• Blocking nature protocol.
07	Log-based [Elnozahy et al. 2002] (Optimistic)	• Non-blocking.	• Orphans message. • Multiple checkpoint. • Recovery line computation overhead.
08	Log-based [Elnozahy et al. 2002] (Causal)	• No orphan message. • Non-blocking.	• Require tagging of extra data on normal application message.

Table I: Pros and Cons of Rollback Recovery Protocols.

2. LITERATURE REVIEW

Research on fault tolerance for distributed systems has received tremendous interest in the recent past. But these schemes cannot be applied directly to ad hoc networks due to lack of central control, no fixed stable host or mobile support station, necessitating development of specific schemes.

[Biswas et al. 2012] proposed mobility based checkpointing and trust based rollback recovery protocol to provide fault tolerance in Mobile Ad hoc networks. Wireless ad hoc network devices are failure prone and security attack prone. Hence, the authors propose adding security of checkpointing in mobile host as a factor to calculate the trust factor of mobile host. Mobility based checkpointing limits the recovery time and trust based recovery increases recovery probability of failed mobile hosts.

[Awasthi and Kumar 2007b] introduced a synchronous checkpointing protocol for mobile distributed system to make it fault tolerant. The protocol reduces redundant checkpoints and blocking of process during checkpointing by using a probabilistic approach. In this scheme a process takes an induced checkpoint if the probability that it will get a checkpoint request in current initiation is high.

[Tuli and Kumar 2011a] present an asynchronous checkpointing and optimistic logging strategy for mobile ad hoc networks. In a wireless ad hoc network due to unreliable mobile hosts and network connections only checkpointing is not sufficient to ensure reliability. Hence message logging is also included which is typically carried out by cluster heads. In this scheme each mobile host takes checkpoints independently and messages delivered to the mobile host are routed through the respective cluster head which logs the message on its own stable storage. The algorithm operates inspite of mobile host failures and disconnections from cluster, especially due to the handoff procedure with detailed sequence of events helps in recovery of information transfer.

[Jiang et al. 2008] suggest a novel communication-induced ckeckpointing algorithm that makes every checkpoint belong to a consistent global checkpoint, where every process stores the tentative checkpoint in memory first and then flushes it to stable storage when there is no contention for stable storage or after finalizing the tentative checkpoint. Messages sent and received after a process takes a tentative checkpoint are finalized. The tentative checkpoint can be flushed to stable storage any time.

[Jaggi and Singh 2010] propose a log-based recovery protocol for application in large-scale mobile computing environment. The scheme employs sender-based message logging along with movement based checkpointing to reduce the number of checkpoint taken by a mobile host. The mobility of a node is used for deciding when a checkpoint needs to be taken. A base transceiver station is used to store the checkpoints and message logs of the mobile hosts.

[Tuli and Kumar 2011b] propose a message induced soft checkpointing for recovery in mobile environments. The protocol takes soft checkpoints saved locally on the mobile host. Before disconnecting from the mobile support station, these soft checkpoints are converted to hard checkpoints and sent to the mobile support station to be saved on stable storage. Taking the soft checkpoint avoids the overhead of transferring large amount of data to the stable storage. The proposed protocol is non-blocking and adaptive.

[Men et al. 2008] present a checkpointing and rollback recovery scheme for the cluster-based multi-channel ad hoc wireless networks where the cluster head controls the mobile hosts to take checkpoints during checkpoint beacon intervals and to rollback to consistent state in case of failure. This scheme is capable of handling ordinary host failure including crash of gateway between two neighbor clusters. Each cluster head uses beacon packet which contains clock data, traffic indication messages and data window apart from variables such as checkpoint index, ordinary node queue and reply messages. The recovery scheme has no domino effect and the failure process can start from its latest local consistent checkpoint then replay the messages to make the gateway consistent again. Simulation results show that this scheme has fast recovery upon transient failures with low additional overheads.

[Lim 2011] propose a new checkpointing scheme based on the concept of checkpoint agents and recovery distance. By combining both, the protocol provides the capability of semantics-aware checkpointing. The proposed scheme allows the distributed mobile application to specify its level of its checkpointing strictness. The strictness is defined by maximum rollback distance that defines the number of recent local checkpoints that can be rolled back in the worst case.

[Kumar and Khunteta 2010] propose a minimum process, non-blocking coordinated checkpointing algorithm for mobile distributed system. Selective processes are blocked for a short duration while being allowed to do their normal computation and send messages during the blocking period. The algorithm incurs no useless checkpoints while optimizing the blocking of processes. It thus involves low cost of maintaining exact dependencies among processes employing piggybacking checkpoint sequence number and dependency vectors onto normal computation messages.

[Jaggi and Singh 2011] present a staggered checkpointing scheme to evade synchronous contention for stable storage in mobile ad hoc networks. The scheme is particularly required for coordinated checkpointing where the number of processes taking their checkpoints simultaneously is large. The scheme is designed ito work effectively with limited storage and a non FIFO channel. Simultaneous checkpoint initiation is allowed by design with capability to handle multiple failures.

[Saluja and Kumar 2011] contribute a new minimum process checkpointing approach for mobile ad hoc networks based on the cluster-based routing protocol, reducing routing traffic and flooding in route discovery. The checkpoint is initiated by any process on the mobile host by taking the tentative checkpoint before sending message forwarding the request to the cluster head, which then coordinates checkpointing with other processes on behalf of the mobile host. Only those processes that participate in checkpointing operation with the initiator are included in the minimum processes set created with Z-dependencies notion. The scheme ensures that number of coordinated messages between a cluster head and its ordinary members is small.

[Morita and Higaki 2001] present an approach to mission critical applications where the system can have both mobile and fixed station. Due to the limitations of mobile stations, checkpointing is recorded asynchronously, whereas fixed stations perform checkpointing synchronously. During recovery stage the mobile station retrieves the local state from the consistent set along with message logs stored in stable storage. Communication and synchronization overheads are minimized as this algorithm separates content and order of information.

[Sharma and Awasthi 2013] propose a log based fault tolerance scheme for cell based Mobile Distributed Computing System that takes advantage of fixed infrastructure of the cellular network. As each message is bound to be routed to the Mobile Host (MH) through its current MSS (Provide FULL FORM OF MSS) which also acts as the stable storage for MH, the message can be logged in advance at MSS. The messages resulting from the outside world interaction by the MH are then merged into this message log. Checkpoints are taken on periodic basis so as the old messages of a MH can be purged. The proposed scheme is novel and requires minimum overheads on wireless channels and MHs, i.e., overhead of checkpointing and transferring the checkpoints to the MSS. Rest of the overhead of logging the events and checkpoints on stable storage is transferred to much powerful MSS side of the system.

[Awasthi and Kumar 2007a] have proposed a synchronous checkpointing protocol where only interacting processes are needed to maintain checkpoints. The initiator MSS collects dependencies of all processes, computes the tentative minimum set, and broadcast the tentative minimum set along with the checkpoint request to all MSSs. Initiator MSSs broadcasts exact minimum set along with commit request on the static network. However, this approach leads to blocking of processes.

[Awasthi et al. 2010] present a weighted checkpointing protocol for mobile distributed systems that significantly reduces checkpointing overheads on mobile nodes. The protocol requires no synchronization messages and decreases checkpoints taken by MHs. The faults at MH which are frequent typically entail less cost as compared to faults at the MSS and are assigned variable weights. The weighted method leads to considerable improvements and makes the protocol adaptable. If the weights are low, most of the checkpoints will be skipped and protocol will behave as log based protocol and if the weights assigned to the events are high, the protocol will behave as NRAS [Add full form and reference] protocol.

On the basis of the review we have identified attributes such as checkpointing techniques, features, channel allocation, retrieval process, parallel initiation and stable storage on which to compare different strategies. The process of checkpointing in a distributed system can be coordinated, uncoordinated, communication induced and log based. Different rollback-recovery protocols offer different tradeoffs with respect to performance overhead, latency of output commit, storage overhead, ease of garbage collection, simplicity of recovery, freedom from domino effect, freedom from orphan processes, and the extent of rollback. Several studies have shown that these protocols perform reasonably well in practice, and that several factors such as checkpointing frequency, machine speed, infrastructure-based or infrastructure-less network, network bandwidth and stable storage etc. play more important roles than the fundamental aspects of a particular protocol.

3. COMPARISON OF ROLLBACK RECOVER PROTOCOLS

The comparative analysis of various checkpointing and rollback recovery protocols is presented in tabular form in Table 2.

Algorithm	**Checkpointing Techniques**	**Attributes**	**Channel**	**Retrieval Process**	**Parallel Initiation**	**Stable storage**
[Biswas et al. 2012]	Asynchronous	• Ensure consistent recovery. • No orphan and lost message. • Minimum checkpoint and log overhead. • Checkpoint only through trusted host.	FIFO	Through intermediate mobile host	Possible	Intermediate node
[Awasthi and Kumar 2007b]	Synchronous	• Protocol reduces the useless checkpoint. • It uses the probabilistic approach. • If the probability is not good then process buffers message till it take checkpoint or receives commit message.	FIFO		Not Possible	Mobile support station

[Tuli and Kumar 2011a]	Asynchronous	• It uses asynchronous checkpointing and optimistic message logging in cluster based system. • This scheme avoids overhead of message logging at mobile host. • It handle both failure and disconnection are planned.	Non FIFO	Only for Ordinary Mobile Nodes	Not Possible	Mobile Support station at Cluster Head
D. Manivannan et.al [[Jiang et al. 2008]]	Communication Induced or Quasii- Synchronous	• Overcome the disadvantages of synchronous and asynchronous, and have the advantages of both synchronous and asynchronous algorithms. • It prolong the response time due to before processing a received message. • Optimistic mean saving the checkpoint and message log first and then flushes from the stable storage to prevent contention for stable storage.	FIFO	Fixed and mobile hosts	Possible	Mobile Support Station
[Jaggi and Singh 2010]	Synchronous	• It combine the movement based checkpointing with message logging. • Recovery related message are reduced and free from domino effect.	FIFO	Fixed host and mobile support station	Not Possible	Mobile Support Station and fixed hosts

[Tuli and Kumar 2011b]	Synchronous	• It uses message induced soft checkpointing for recovery. • Soft checkpoint saved locally in mobile host and hard checkpoint saved on stable storage permanently. • It is non-blocking and adaptive due to checkpoint sequence number.	FIFO	Mobile host and mobile support station	Not Possible	Mobile host and mobile support stations
[Men et al. 2008]	Synchronous	• Cluster based multi-channel management protocol. • Local consistent checkpoint two consecutive beacon interval. • Rollback recovery one beacon interval.	FIFO	Only for ordinary mobile nodes and crash of gateway	Not Possible	Mobile support station at cluster head
[Lim 2011]	Synchronous	• It uses tunable checkpointing scheme for mobile environment by strictly defining the maximum rollback distance. • It is based on the concept of R-distance, which reduces the average cost for creating the consistent global checkpoint. • Logging agent can reduce the amount of checkpoint related data transferred between MSS's and MH's.	FIFO	Fixed and mobile hosts	Possible	Mobile Support stations and Mobile Hosts

[Kumar and Khunteta 2010]	Synchronous	• Only minimum number of processes is required to take their checkpoints • This scheme forces zero useless checkpoints at the cost of very small blocking.	FIFO	Possible		Mobile support station
[Jaggi and Singh 2011]	Asynchronous	• Wireless bandwidth Suitable for small sized message log. • Handle multiple failure. • Cluster head storage contention removed.	Non FIFO	Only for cluster head and nodes for multiple failures	Possible	Own memory
[Saluja and Kumar 2011]	Synchronous	• Ensure zero blocking time. • Reduced message complexities. • Useless checkpoints are minimize by maintaining exact dependencies among process. • Piggybacking checkpointing sequence number and dependency vector on to the normal messages.	FIFO	Only for ordinary mobile host	Possible	Cluster head

[Morita and Higaki 2001]	Hybrid Checkpointing	• Hybrid checkpointing protocol support both fixed and mobile stations. • It support consistent recover for both fixed and mobile stations. • It reduced communication and synchronization overheads.	FIFO	Fixed and mobile hosts	Not Possible	Mobile support station
[Sharma and Awasthi 2013]	Hybrid Checkpointing	• Checkpointing and rollback recovery schemes are used. • Checkpoints are taken periodically and old message can be purged.	FIFO	Mobile host and MSS	Not Possible	Mobile support station
[Awasthi and Kumar 2007a]	Synchronous	• Minimum process synchronous checkpointing. • Checkpoint and blocking are reduced by probabilistic approach. • Low memory and computation overheads on MHs. • Low communication overheads on wireless channels.	FIFO	Through local MHs and local MSS	Not Possible	Local MHs and mobile support station

[Awasthi et al. 2010]	Quasi-synchronous	• It is Hybrid of quasi-synchronous and log-based approach. • No message synchronization. • Reduced checkpointing overheads at mobile nodes. • Dimino free. • Purging cost and large stable storage needed. • Recovery is complex.	FIFO	Through MHs and MSSs	Possible	Local MHs and mobile support station

Table II: Comparison of rollback recovery protocols

3.1 CONCLUSIONS

It is challenging to design the perfect fault-tolerance strategy based on rollback recovery for mobile and wireless ad hoc networks. This paper explored different checkpointing and rollback recovery strategies with respect to a set of attributes including performance overhead, storage overhead, and ease of recovery, freedom from domino effect, freedom from orphan processes and the extent of rollback. It is observed that Synchronous checkpointing which requires the participating processes to synchronize their checkpoints to form a globally consistent system state, simplifies recovery and yields good performance in practice. Asynchronous checkpointing does not require the processes synchronize their checkpoints, but it suffers from potential domino effect, complicates recovery and still requires synchronization to perform output commit or garbage collection. On the other hand Quassi synchronous or communication induced checkpointing schemes relies on the communication patterns of the application to trigger checkpoints. These schemes do not suffer from domino effect and do not require synchronization. Log based rollback recovery is often a natural choice for applications that frequently interact with external applications.

In practice, depending upon the class of application, its criticality and performance requirements, a hybrid approach might work best. However, real-world testing and benchmarking requires to be carried out for different application categories to determine the best applicable schemes. Future recovery schemes could be self-aware, predict failures based on past-analysis and intelligently respond to dynamic changes in environment. Rapidly increasing processing power, storage and battery life of mobile devices will help in making such schemes feasible in future.

REFERENCES

Acharya, A. and Badrinath, B. R. 1994. checkpointing distributed applications on mobile computers. In *IEEE international conference on parallel and distributed information systems*. pp.73–80.

Awasthi, L. and Kumar, P. 2007a. Asynchronous checkpointing protocols for mobile distributed systems: Probablistic approach. *International Journal of Information and Computer Security Vol.4,* No.3, pp.298–314.

Awasthi, L. K. and Kumar, P. 2007b. A synchronous checkpointing protocol for mobile distributed systems: Probabilistic approach. *International Journal of Information and Computer Security*.

Awasthi, L. K., Misra, M., and Joshi, R. C. 2010. A weighted checkpointing protocol for mobile distributed systems. *International Journal of Ad-hoc and Ubiquitous Computing Vol.5*, pp.137–147.

Biswas, S., Neogy, S., and Dey, P. 2012. Mobility based checkpointing and trust based recovery in manet. *IJWMN Vol.4,* No.4, pp.53–69.

Cao, G. and Singhal, M. 2001. Mutable chekpoints: A new checkpointing approach for mobile computing systems. *IEEE Transation on parallel and distributed systems Vol.12,* No.2, pp.157–172.

ELNOZAHY, K. N., ALVASI, L., WANG, Y. M., AND JOHNSON, D. B. 2002. A survey of rollback recovery protocols in message passing systems. *ACM computing surveys Vol.34,* No.3, pp.375–408.

JAGGI, P. K. AND SINGH, A. K. 2010. Log-based recovery with low overheads for large mobile computing system. *JISE*.

JAGGI, P. K. AND SINGH, A. K. 2011. Staggered checkpointing and recovery in cluster based mobile ad hoc network. *Advances in Parallel Distributed Computing Communications in Computer and Information Science Vol.203*, pp: 122–134.

JIANG, Q., LUO, Y., AND MANIVANNAM, D. 2008. An optimistic checkpointing and message logging approach for consistent global checkpoint collection in distributed system. *Journal of Parallel Distributed Computing, Elsevier Vol.68*, pp.1575–1589.

KUMAR, P. AND KHUNTETA, A. 2010. A minimum process coordinated checkpointing protocol for mobile distributed system. *International Journal of Computer Science (IJCSI) Vol.7,* No.4.

LIM, S. 2011. A tunable checkpointing algorithm for the distributed mobile environment. *International Journal of Computer Science Issue (IJCSI) Vol.8,* No.3.

MEN, C., XU, Z., AND LI, X. 2008. An efficient checkpointing and rollback recovery scheme for cluster-based multi-channel ad hoc wireless networks. In *ISPA, IEEE International Symposium on Parallel and Distributed Processing with Applications*. pp: 371–378.

MEN, C., XU, Z., AND WANG, D. 2007. An efficient handoff strategy for mobile computing checkpoint system. *EUC, LNCS Vol.4808*, pp.410–421.

MORITA, Y. AND HIGAKI, H. 2001. Hybrid checkpointing protocol for supporting mobile to mobile communication information networking. In *15th International conference*. pp.529–536.

PARK, T., WOO, N., AND YEOM, H. Y. 2003. An efficient recovery scheme for fault tolerant mobile computing system. *future Generation Computer System Vol.19*, pp.39–53.

PRAKASH, R. AND SINGHAL, M. 1996a. Low cost checkpointing and failure recovery in mobile computing systems. In *IEEE trans. On parallel and distributed system*. pp.1035–1048.

PRAKASH, R. AND SINGHAL, M. 1996b. Low cost checkpointing and future recovery in mobile computing system. *IEEE Transaction on Parallel and Distributed System Vol.7*, pp.1–38.

QUAGLIA, F., CICIANI, B., AND BALDONI, R. 2006. Checkpointing protocols in distributed systems with mobile host: A performance analysis. In *Workshop on Fault Tolerant Parallel and Distributed Systems*. pp. 743–755.

RANDILL, B. 1975. System structure for software fault tolerance. *IEEE Trans on software Engineering Vol.1*, pp.220–232.

SALUJA, K. AND KUMAR, P. 2011. Transitive dependencies tracking in minimum-process checkpointing protocol for mobile ad ho networks. *International Journal of Computer Science and Communication Technologies Vol.4*, pp.700–704.

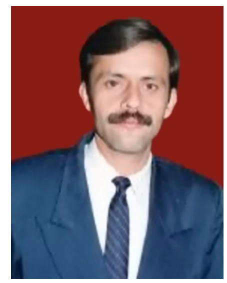
SHARMA, R. AND AWASTHI, L. K. 2013. A log-based recovery protocol for mobile distributed computing systems. *Journal of Bioinformatics and Intelligent Control Vol.1*, pp.1–10.

TULI, R. AND KUMAR, P. 2011a. Asynchronous checkpointing and optimistic message logging for mobile ad hoc networks. *International Journal of Advanced Computer Science and Applications Vol.2,* No.10, pp.70–76.

TULI, R. AND KUMAR, P. 2011b. Message induced soft checkpointing for recovery in mobile environments. *International Journal on Computer Science and Engineering (IJCSE) Vol.3*, pp. 1798–1803.

Jawahar Thakur is currently pursuing his Ph.D. Degree in Department of Computer Science, Himachal Pradesh University, Shimla, India. He received his B. Tech. and M. Tech. Degrees in Computer Science and Engineering from National Institute of Technology, Hamirpur in 1996 and PU Chandigarh in 2010, respectively. He is presently Associate Professor in the Department of Computer Science of Himachal Pradesh University. His research interests include design and performance evaluation of communication protocols, rollback recovery protocol for wireless ad hoc and sensor networks. He is a life member of Computer Society of India.

Arvind Kalia received his Master of Computer Application (MCA) from Thapar University Patiala Punjab in 1988 and Ph.D. from Punjabi University Patiala in 2001. He is currently Professor in Department of Computer Science, Himachal Pradesh University, Shimla. His research interests include Software Engineering, Computer Networks, and Data Mining. He has contributed more than 80 research articles in national and international journals as well as conferences. He authored two books and guided several Ph.D. theses and ME/ M. Tech. dissertations. He is a life member of Indian Science Congress and CSI.



Lalit kumar Awasthi received his M. Tech. (CS&E) from IIT Delhi in 1993 and Ph.D. from IIT Roorkee in 2003. He is professor in Department of Computer Science and Engineering, National Institute of Technology, Hamirpur (HP). His research interests include mobile distributed systems, Checkpointing based fault tolerance and ad hoc networks. He has contributed more than 100 articles in national and international journal and conferences. Currently he is Director in Jawaharlal Nehru Government Engineering College (JNGEC) Sundernagar, Mandi, Himachal Pradesh. He has guided several Ph.D. theses, M. Tech. / ME dissertations and completed various sponsored projects.