IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

Asynchronous Checkpointing Algorithms for Wireless Ad-hoc Network: A Simulated Analysis

Jawahar Thakur[1], Arvind Kalia[2], L. K. Awasthi[3]

Department of Computer Science, Himachal Pradesh University, Shimla, India [1, 2]

Department of Computer Science & Engineering, National Institute of Technology, Hamirpur, India[3]

Abstract: 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. 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. This paper present the simulated performance analysis of couple of uncoordinated checkpointing protocols for wireless ad-hoc networks on the basis of different size clusters, different number of recovery messages, different numbers of failures, and different recovery time.

Keywords: checkpointing, uncoordinated, recovery, simulation, algorithm, cluster.

I. INTRODUCTION

Mobile Ad hoc networks are the networks in which devices can communicate with each other over the wireless links without using any fixed infrastructure and centralized controller. These are a self-configurable network which means the devices when come in the range of each other form a network on their own.There are various challenges associated with mobile ad-hoc network that makes it difficult to implement the various checkpointing schemes on them. Checkpointing schemes are easily implementable on the distributed systems due to their fixed nature, sufficient power resources, availability of large storage space. These schemes must be also implementable on the ad-hoc networks with some negotiations.Checkpoint can be defined as a fault tolerant technique that is a designated place in a program at which normal processing is interrupted specifically to preserve the status information necessary and then to allow resumption of processing at a later time. If there is a failure, computation may be restarted from the current checkpoint instead of repeating the computation from the beginning.

Aspects of Checkpointing:

Checkpoint-based rollback recovery restores the system state to the most recent consistent set of checkpoints whenever a failure occurs [1]. Checkpoint based rollback recovery is not suited for applications that require frequent interactions with the outside world, since such interactions require that the observable behaviour of the system through failures. Checkpoint technique can be classified into three categories: uncoordinated checkpointing, coordinated checkpointing and communication-induced check pointing.

Uncoordinated checkpointing allows any process to initiate checkpointing. The advantage of this scheme is that each process may take a checkpoint in any critical state [2]. Coordinated checkpointing simplifies recovery and with no domino effect, since every process always restarts from its most recent checkpoint. Coordinated checkpointing requires each process to maintain only one permanent checkpoint on stable storage, reducing storage overhead and eliminating the need for garbage collection [3].Blocking Checkpoint coordination: Blocking algorithms force all relevant processes in the system to block their computation during checkpointing latency and hence degrade system performance. Checkpointing includes the time to trace the dependence tree and to save the states of processes on stable storage, which may be long. Therefore, blocking algorithms may degrade system performance [4].Non-blocking Checkpoint Coordination: In this protocol, the initiator takes a checkpoint and broadcasts a checkpoint request to all processes. Each process takes a checkpoint upon receiving the request and rebroadcasts the request to all processes, before sending any application message. The protocol works assuming the channels are reliable and FIFO [5]. Checkpointing with Synchronized Clocks: A process takes a checkpoint and waits for a period that equals the sum of the maximum deviation between clocks and the maximum time to detect a failure in another process in the system. The process can be assured that all checkpoints belonging to the same coordination session have been taken without the need of exchanging any messages [5]. Minimal Checkpoint Coordination: It is desirable to reduce the number of processes involved in a coordinated checkpointing session. This can be done since only those processes that have communicated with the checkpoint initiator either directly or indirectly since the last checkpoint need to take new checkpoints [6]. Communication-induced checkpointing avoids the domino effect while allowing processes to take some of their checkpoints independently [6]. It forces each process to take checkpoints based on information piggybacked on the application. However, process independence is constrained to guarantee the eventual progress of the recovery line and therefore processes may be forced to take additional checkpoints. The checkpoints that a process takes independently are called local checkpoints, while those that a process is forced to take are called forced checkpoints. Model-based Check pointing: Model-based checkpointing relies on preventing patterns of communications and checkpoints that could result in inconsistent states among the existing checkpoints.[6] Index based Communication Induced Checkpointing: Index-based communication induced check pointing works by assigning monotonically increasing indexes to checkpoints, such that the checkpoints having the same index at different processes form a consistent state [6]. The main objectives of this paper is to do the performance analysis of couple of the uncoordinated checkpointing protocols in different size cluster based wireless ad-hoc networks by using the simulation tool.

The rest of the paper is organized as: part II review of literature where various checkpointing protocols are discussed: Part-III uncoordinated checkpointing protocol are implemented and simulated analysisPart-IV, finally conclusion and future scope.

II. 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. [7] 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. [8] 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. [9] 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. [10] Suggested 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. [11] 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. [12] 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. [13] 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 eff ect 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 transientfailures with low additional overheads.[14] 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. [15] Provides an overview of the available checkpointing strategies for mobile networks, comparing them on the various parameters. They conclude that no

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

single strategy is optimal in all fault scenarios and that the perfect strategy may still be in the works.

III. ASYNCHRONOUS CHECKPOINTING IMPLEMENTATION AND SIMULATED ANALYSIS

Asynchronous checkpointing protocol proposed by Singh and Jaggi [16] are considered for implementation and comparison. The implementation is done in a simulated environment by using the open source network simulator version NS2.35. The system model for the implementation is cluster-based wireless ad-hoc networks is considered.
In [16] the algorithm allows the asynchronous rollback recovery of a process in the ad hoc environment after a crash failure using movement based checkpointing and message logging. The challenges faced by the recovery process due to the following constraints of MANETs: inadequate stable storage, limited wireless bandwidth, dynamic topology and hence network partitions. The MANET is organized into a hierarchical structure by segregating the nodes into disjoint and virtual groups called clusters. The clustering approach is utilized to handle the limitation of stable storage. A self-stabilizing spanning tree is maintained over the resultant topology to reduce the number of recovery related messages. This conserves the wireless bandwidth and also deals with the changing topology or partitions in the network. This scheme combines message logging with movement based checkpointing to limit the overhead due to checkpointing. The results of simulation of the scheme for hosts with different mobility rates and for networks with different number of clusters are described.
It is a movement based checkpointing and message logging algorithm that allows the asynchronous recovery of a mobile host subsequent to a crash failure. Any mobile host in the network is always a member of some cluster in the network. It may be a special node i.e. the Cluster Head or an ordinary member affiliated to the Cluster Head. As the mobile host moves in the network, it may change its cluster and hence the affiliation to a CH. Since stable storage is limited at a CH, the checkpoint and recovery related information of a MH are distributed among the CHs with which the mobile host affiliates as it moves across the network. This approach provides an additional benefit of avoiding the simultaneous access to stable storage present at one place by all the MHs. Each MH uses the stable storage at the CHs with which it affiliates as it moves.

A weight, W, is assigned to each cluster head based on its connectivity with other CHs at a given time. This weight of a given CH is equal to the number of its neighbour CHs. Since the topology of the network is dynamic, the weight of the CH may change over time.
The CH of the first cluster that a MH joins on entering the network becomes its Checkpoint & Movement Coordinator (CMC). The MH saves its initial state at the CMC. The further movement of the MH defines a virtual region comprising the cluster of the CMC and its neighbouring clusters. The messages of a MH are logged at the current CH but till the time the MH is in a virtual region, its checkpoint remains at the CMC. However the message log unifying scheme ensures that the message log of the MH at all CHs except its current and previous to current CHs keeps getting unified at the CMC as the MH moves in a virtual region. Once the MH moves to a CH which is not a neighbour of the CMC, a new checkpoint is taken at that CH and it becomes the new CMC of the MH. Therefore at any given instant, the recovery related data of a MH is distributed amongst the CMC and a maximum of two CHs.

The second protocol proposed by Tuli and Kumar [17] is considered for implementation and comparison. The protocol is based on message logging scheme and well-suited to provide fault tolerance for cluster federations in mobile ad hoc networks.It has been stated that the scheme is based on optimistic message logging for communications in a clustered ad hoc network.Meanwhile the checkpointing-only schemes are not suitable for the mobile environment and also in ad-hoc environments in which unreliable mobile hosts and fragile network connection may hinder any kind of coordination for checkpointing and recovery. In order to cope with the storage problem, the task of logging is assigned to the CH instead of MHs, since each message heading to a MH is routed through the CH. Also, in order to reduce the overhead imposed on mobile hosts, cluster heads take charge of logging and dependency tracking, and mobile hosts maintain only a small amount of information for mobility tracking.

Simulation Parameter
The various simulation parameters that are used for taking the checkpoints are shown in Table 1 with their units.

Table 1: Parameters for checkpointing

Log Size	50 Bytes
Checkpoint Size	3-256 KB
Coordination message size	2 B
Computation message size	50 B
Time to transfer checkpoint per hop through wireless channel	0.10s
Time to transfer coordination message	.0001s
Time to transfer log or computation message for single hop	0.002s
Energy capacity of a node	1000J
Mobility rate	1
Cluster change count threshold	3-17

The parameters that have been used to compare the two algorithms are shown in the Table 2 as below.

Table 2: comparison parameters

Number of clusters	6,12,18
Number of Failures	Different values
Number of Recovery messages	Different values
Recovery Time	Different values

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

Both the algorithms considered for comparison are based on clustered model of mobile ad hoc networks. Comparison of the protocol is implemented on the basis of different clusters size, different numbers of failures, different numbers of recovery messages, and different recovery time.
The tool used for simulation implementation is network simulator NS 2.35. For evaluation, the number of failure is taken along X- axis and number of recovery messages for each algorithm with various clusters is taken along the Y-axis. The detailed result and graphs are discussed in the following Tables and Figures. The variation of number of recovery messages for the same number of failures for both the algorithms in a clustered environment that has 6 clusters is shown in the Table 3.

Table 3: No. of Failures vs No. of Recovery Messages for 6 clusters

No. of failures	No. of recovery messages	
	Algo [1]	Algo [2]
115	55	40
140	96	82
165	110	95
215	120	108
240	135	122

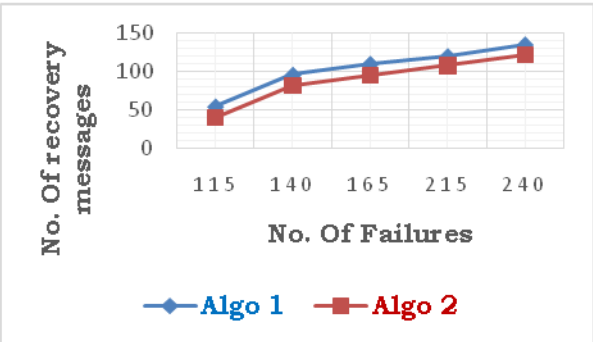

Figure 1: No. of failures vs No. of recovery messages for 6 Clusters.

The variation of number of recovery messages for the same number of failures for both the algorithms in a clustered environment that has 12 clusters is shown in the Table 4.

Table 4: No. of Failures vs No. of Recovery Messages for 12 clusters

No. of failures	No. of recovery messages	
	Algo[1]	Algo[2]
115	50	36
140	54	40
165	61	49
215	84	68
240	100	85

Based on the Table 4 the graph has been plotted as shown in Figure 2

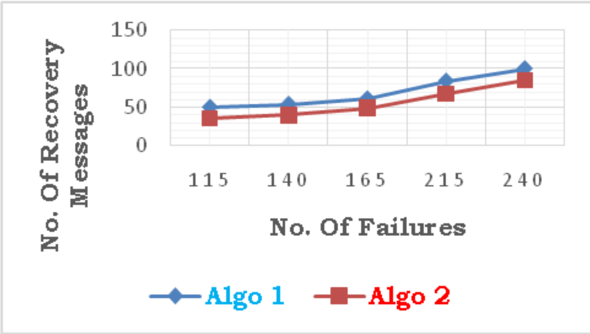

Figure 2: No. of failures vs No. of recovery messages for 12 Clusters

The variation of number of recovery messages for the same number of failures for both the algorithms in a clustered environment that has 12 clusters is shown in the Table 5.

Table5: No. of Failures vs No. of Recovery Messages for 18 clusters.

No. of failures	No. of recovery messages	
	Algo[1]	Algo[2]
115	10	7
140	30	20
165	40	29
215	65	45
240	75	62

Based on the Table 5 the graph has been plotted as shown in Figure 3.

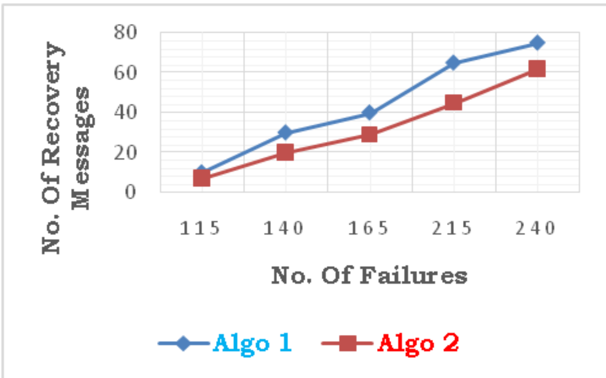

Figure 3: No. of failures vs No. of recovery messages for 18 Clusters

On the basis of analysis of all the above tables and figures it can be concluded that the second algorithm, Algo[2] given by Ruchi Tuli and P. Kumar [17] outperforms the first algorithm, Algo[1] given by A.K.Singh and P.K.Jaggi [16] since it needs less number of recovery messages that implies that it will have less communication overhead in MANET with any number of clusters. The variation of recovery time with the number of clusters for both the algorithms in a clustered environment that has 6, 12, 18 clusters is shown in the Table 6.

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 4, April 2016

Table 6: No. of Clusters vs Recovery Time

No. of Clusters	Time for recovery(in second)	
	Algo[1]	Algo[2]
6	0.51	0.37
12	1.10	0.96
18	1.73	1.54

Based on the Table 6 the graph has been plotted as shown in Figure 4.

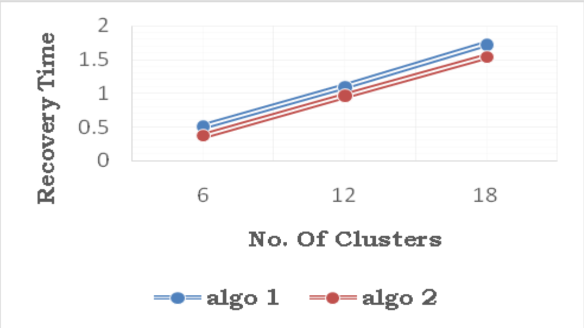

Figure 4: No. of clusters vs Recovery time

From the analysis of above Table 6 and Figure 4 it can be clearly said that the Algo[2] is better than the Algo[1] as the recovery time of a node is less in Algo[2] which means it is faster in terms of recovery of a node which is required for successful operation of MANETs.

IV. CONCLUSION

The work concludes the study of emerging context of checkpointing in MANETs and targeted issues related to the efficient rollback recovery by reducing the recovery time and making system fault tolerant. It also includes the empirical evaluation of some algorithms using software framework NS2 contributing substantially to the investigation of applicability of a specific algorithm for making the checkpointing a better option for recovery during failures.The work attempted to perform a study of various heuristics for checkpointing in distributed systems and mobile ad-hoc networks.The empirical results show that the algorithm given by Ruchi Tuli and B. Kumar [17] algorithm Algo[2] outperforms the algorithm given by A.K.Singh and P.K.Jaggi [16] algorithm Algo[1] in consideration of any number of clusters in a clustered environment as in it less number of recovery messages are required which means it has less message overheads and ultimately lesser bandwidth wastage. And it also takes less time for recovery which is important in a network so that network becomes fault tolerant.There are other checkpointing algorithms that might be better than the chosen one. So implementation of those checkpointing algorithms can be done on different simulators or tools with different parameters which can lead to the discovery of best checkpointing and recovery scheme. In future the algorithm implementation and comparison will be performed on test bed available for this type of research work.

REFERENCES

[1] Randall B., “System Structure for Software Fault Tolerance”, IEEE Trans. On Software Engineering, 1(2), 220-232, 1975.

[2] Y. Wang and W.K. Fuchs, “Lazy Checkpoint Coordination for Bounding Rollback Propagation”, Proc. 12th Symp. Reliable Distributed Systems, pp. 78-85, Oct. 1993.

[3] Tamir Y., Sequin C.H., “Error Recovery in Multicomputers using Global Checkpoints”, In Proceedings of the International Conference on Parallel Processing, pp.32-41, 1984.

[4] Guohong Cao, Mukesh Singhal, “On Coordinated Checkpointing in Distributed Systems”, IEEE transactions on parallel and distributed systems, vol. 9, NO. 12, December 1998.

[5] E.N. Elnozahy, L. Alvisi, Y.M. Wang and D.B.Johnson, “A Survey of Rollback-Recovery Protocols in Message-Passing Systems,” ACM Computing Surveys, Vol 34(3), pp. 375-408, 2002.

[6] Franco Zambonelli, “On the Effectiveness of Distributed Checkpoint Algorithms for Domino Free Recovery”, IEEE, Proceeding of HPDC-7, 98, Chicago, July 1998.

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

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

[9] Tuli, R. and Kumar, P.. Message induced soft checkpointing for recovery in mobile environments. International Journal on Computer Science and Engineering (IJCSE) Vol.3, pp. 1798–1803,2011.

[10] Jiang, Q., Luo, Y., and Manivannam, D.. 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, 2008.

[11] Jaggi, P. K. and Singh, A. K., Log-based recovery with low overheads for large mobile computing system. JISE, 2010.

[12] Tuli, R. and Kumar, P.. Message induced soft checkpointing for recovery in mobile environments. International Journal on Computer Science and Engineering (IJCSE) Vol.3, pp. 1798–1803. 2011.

[13] Men, C., Xu, Z., and Wang, D. An efficient handoff strategy for mobile computing checkpoint system. EUC, LNCS Vol.4808, pp.410–421, 2007..

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

[15] Thakur. J, Kalia. A, and Awasthi. L. K, Checkpointing and Roll back Recovery Protocols in Wireless Ad hoc Networks: A Review, International Journal of Next-Generation Computing, Vol. 6, No. 1, pp. 140-152, 2015.

[16] Awadhesh Kumar Singh, Parmeet Kaur Jaggi, “Asynchronous Rollback Recovery in Cluster Based Multi Hop Mobile Ad Hoc Networks”, International Journal of Enhanced Research in Management & Computer Applications, ISSN: 2319-7471 Vol. 2 Issue 4, pp: (46-55), April-2013.

[17] R. Tuli and P. Kumar, “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, 2011.

[18] C. Men, Z. Xu and X. Li, “An Efficient Checkpointing and Rollback Recovery Scheme for Cluster-based Multi-channel Ad-hoc Wireless Networks”, Proceeding ISPA '08 Proceedings of the IEEE International Symposium on Parallel and Distributed Processing with Applications, IEEE Computer Society Washington, DC, USA, pp.371-378, 2008.

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

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

[21] Rekha Rao, Jawahar Thakur. Coordinated Checkpointing Algorithms: A Comparative Analysis.Current Trends in Information Technology, 5(3): pp,14–20, 2015.