Volume: 6 Issue: 3 Pages: 2308-2313 (2014) ISSN : 0975-0290

Checkpointing and Rollback Recovery Algorithms for Fault Tolerance in MANETs: A Review

Sushant Patial
Department of Computer Science, Himachal Pradesh University Shimla-5
Email: patialsushant @gmail.com
Jawahar Thakur
Department of Computer Science, Himachal Pradesh University Shimla-5
Email: jawahar.hpu @gmail.com

ABSTRACT

Mobile Ad Hoc Networks (MANETs) are emerging as a major technology in mobile computing. A MANET is a collection of mobile devices or nodes that communicate with each other using wireless links without availability of any static infrastructure or centralized control. A node in such a network should be fault tolerable and failure free execution of processes on the network nodes is vital. In order to make devices fault tolerant checkpoint based recovery technique can be used. Checkpointing is a technique that can be used to make device or node fault tolerant and reduce the recovery time in case of failure. It takes the snapshot of current application state of process and stores it in some memory area and then using it to resume the computation from current checkpoint instead of resuming it from the beginning. Some limitations of MANETs such as mobility, dynamic topology, limited bandwidth of channel, limited storage space and power restrictions makes checkpointing as a major challenge in mobile ad hoc networks. This paper presents the survey of some existing algorithms, which have been proposed for making MANETs fault tolerant and implementing or deploying checkpointing in mobile ad hoc network.

Keywords – **Checkpointing, Dynamic topology, Fault tolerant, MANETs Mobile computing, Mobile Support Station (MSS), Recovery.**

Date of Submission : September 16, 2014 Date of Acceptance : November 03, 2014:

I. INTRODUCTION

Network is a collection of devices called nodes that allow communication among users and shares the resources using some set of rules also called as protocols. Network can be broadly classified into two types. One is the wired networks which are connected through a physical medium or cables, such as Ethernet cables or phone lines. And the other is the wireless networks, using wireless networking cards that send and receive data through the air with the help of radio waves. Wireless networks are gaining much popularity these days since they help in communication in areas where network wiring is almost impossible. On the other hand a distributed system consists of several processes that execute on computers that are separated geographically by some distance and coordinate via message-passing with each other to achieve a common objective [1]. In a traditional distributed system all hosts are stationary. Advances in computers with wireless communication interfaces and satellite services these days have made it possible for mobile users to perform distributed applications and to access information anywhere and at anytime. A new computing environment in which some hosts are mobile computers connected by wireless communication networks and some are stationary computers connected to a fixed network is called as distributed mobile computing environment. Thus, distributed systems can have a special type called distributed mobile system where some of its hosts are not stationary. A distributed mobile system is characterized by the mobility and poor resource of mobile hosts.

Mobile ad-hoc network (MANET) is an autonomous ad hoc wireless networking system which consists of independent nodes that move frequently and changes the network connectivity. MANETs are collection of self-organizing mobile nodes with dynamic topologies and no fixed infrastructure where nodes are autonomous and independent wireless devices. From the fault tolerance prospective the MANETs are highly vulnerable and challenging, basically due to its complex system infrastructure-less network where the wireless mobile nodes are dynamically attached to temporary topology. Nodes do not have to follow any constraint or rules. Nodes can move freely in the network, it indicates that host movement and topology changes frequently.

The advantages of ad hoc network are that they can be easily deployed, their robustness, flexibility and they inherently supports mobility of devices. The topology of ad hoc network is very dynamic because of the host mobility, so MANETs can be very useful where instance communication is required in emergency like military applications, mobile conferencing and inter vehicular communication [2]. When a fault or failures of process

Volume: 6 Issue: 3 Pages: 2308-2313 (2014) ISSN : 0975-0290

occur, an application with mobile hosts must rollback to a consistent global checkpoint as close as possible to the end of the computation. The main constraints of ad-hoc networks includes dynamic network topology, limited bandwidth, variability of the links, low node capability in terms of limited power or battery, no centralized control, broadcasting nature of transmission and packet losses[2]. Some other constraints to these networks are frequent disconnections/partitions/joins of links of nodes, no stable storage or limited stable storage, different mobility pattern of nodes, devices are vulnerable to physical security threats.

As fault-tolerance is an important design issue in building a reliable Ad hoc network, MANETs must be fault tolerant that is they must be able to recover even after a failure occurs. Transient failures in system are the one which stays for short duration time during operation only. If the fault is recognized in the system, the fault tolerance technique allows the system to resume the computation from the last consistent state and thus reducing the recovery time. There are various recovery scheme that have been proposed to make the system fault tolerant such as log based recovery, rollback recovery and checkpointing. This paper has been organized into different sections. Section II gives description about checkpointing and its types. Section III describes about the work done by various research scholars in the field of checkpointing in MANETs. Finally the conclusion is given in the Section IV.

II. CHECKPOINTING AND ITS TYPES

Checkpointing is a technique for inserting fault tolerance into computing systems. It basically consists of taking a snapshot of the current application state, storing it on some memory area and later on using it for restarting the execution from that particular point in case of failure. It is a fault tolerant technique in which normal processing of a process is interrupted specifically to preserve the status information necessary and then to allow resumption of processing at a later time. Computation may be restarted from the current checkpoint instead of repeating it from the beginning if a failure occurs. Checkpoint based rollback recovery is being used as a technique in various areas like scientific computing, mobile computing, distributed database, telecommunication and critical applications in distributed and mobile ad hoc networks.

Checkpoint-based rollback recovery restores the system state to the most recent consistent set of checkpoints whenever a failure occurs [3]. Checkpoint based rollback recovery is not suited for applications that require frequent interactions with the outside world, since such interactions require that the observable behavior of the system through failures. Checkpointing technique can be basically classified into three categories that are: uncoordinated checkpointing, coordinated checkpointing (blocking and non-blocking) and communication-induced check pointing.

- **Uncoordinated Check pointing:** It allows any process can initiate checkpointing. Each process can take a checkpoint in any critical state and does not need to coordinate with other processes in the system. [4].
- **Coordinated Checkpointing:** This type of checkpointing simplifies recovery and with no domino effect, since each process is restarted from its most recent checkpoint not from the beginning. Coordinated checkpointing requires that only one permanent checkpoint is maintained on stable storage by each and every process which helps in eliminating the need for garbage collection and reducing storage overhead[5].
- **Blocking Checkpoint coordination:** These 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 trees and to save the states of processes on some stable storage, which may take some time. Therefore, these algorithms may degrade the performance of system [6].
- **Non-blocking Checkpoint Coordination:** In this protocol, a checkpoint is taken by the initiator and then a checkpoint request is broadcasted to all the processes. When each process receives a request it takes a checkpoint and before sending any application message rebroadcasts the request to all processes. This protocol works on assumption that the channels are reliable and FIFO based [7].
- **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. It can be assured to the process that all the checkpoints belonging to the same coordination session have been taken without the need of exchanging any messages [7].
- **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 [8].
- **Communication - induced Checkpointing**: This type of checkpointing avoids the domino effect while allowing processes to take some of their checkpoints independently [8]. 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 are taken by a process independently are called as local checkpoints, while those that are taken by a process forcibly are called forced checkpoints.
- **Model-based Checkpointing**: It relies on preventing patterns of communications and checkpoints that could result in inconsistent states among the existing checkpoints.[8]

Volume: 6 Issue: 3 Pages: 2308-2313 (2014) ISSN : 0975-0290

- **Index based Communication Induced Checkpointing**: This type of checkpointing works by assigning monotonically increasing indexes to checkpoints, such that the checkpoints having the same index at different processes form a consistent state [8].
- **Hybrid Checkpointing:** There might be some situations where we require two or more checkpointing schemes in one algorithm; such type of checkpointing where combination of checkpointing schemes is used is called as hybrid checkpointing.

III. ANALYSIS OF CHECKPOINTING ALGORITHMS FOR MANETS

There are various checkpointing schemes or algorithms that have been developed for reducing the time for recovery if any failure occurs. The flexibility introduced by mobile computing brings new challenges to the area of fault tolerance. Failures become common which were rare with fixed hosts, fault detection and message coordination are made difficult by frequent host disconnection. Some of the checkpointing algorithms developed for MANETs are as follows:

Masakazu and Hiroaki [9] proposed an approach called Checkpointing by flooding method. According to this protocol ad hoc networks works without any stable storage and enough communication bandwidth. Here, flooding is used to deliver a checkpoint request message. This message carries the state information of a mobile computer and stored into neighboring mobile computers. Intermediate mobile computer stores a candidate of a lost message after its detection on its transmission route.

Singh and Jaggi [10] proposed a Concurrent Checkpointing and Recovery scheme. They presented a staggered approach in their work to avoid resources simultaneous contention. The events which would normally happen at the same time are forced to start or happen at different times by staggering. This protocol logs minimum number of messages and does not need any FIFO channels. It successfully handles the overlapping failures in ad hoc networks and supports concurrent initiation of checkpoints.

Saluja and Kumar [11] in their work discussed a new minimum process checkpointing procedure for mobile ad-hoc networks which is based on the cluster based routing protocol that reduces routing traffic and prohibitive of flooding traffic in discovery of routes. A checkpoint can be initiated by any of the process (MH) in this algorithm, first it takes tentative checkpoint before sending message and then sends request to CH and then on the behalf of MH the CH coordinates checkpointing operation with other processes. Only those process participate in checkpointing operation with the initiator which are present in the minimum processes set created with Z-dependencies notion. This algorithm ensures that blocking of processes does not take place and takes no useless checkpoints as it maintains exact dependencies and piggybacks checkpoint sequence number, dependency vector onto the normal message communication.

Morita and Higaki [12] presented an approach to mission critical application where the system can have both mobile stations and fixed stations. Due to several limitations of mobile stations checkpointing is recorded asynchronously whereas fixed station take checkpointing synchronously. During the recovery process mobile station will get local state from 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.

Juang and Liu [13] provided with an independent checkpointing and rollback recovery technique in multihop communication environment. In the state transition interval called interval index depends on message received by the process and state of process, that give way to the development of dependency matrix considering both types of dependencies that are transitive and direct dependencies. All the communication is transmitted from cluster to cluster goes through the clusterhead node CH which acts as a local coordinator of transmissions within the cluster. CH maintains the dependency matrix and message logs hence no additional overheads are present on MH and also when process fails this scheme covers resending of lost messages.

Biswas and Neogy [14] suggested a mobility aware checkpointing and failure recovery algorithm for cluster based mobile ad hoc networks (MANETs) in which checkpoints of mobile nodes are saved on neighboring nodes if the mobility of a node among the clusters crosses the threshold value and if the failure occurs recovery of node is done through the mobile cluster head. This algorithm shows the minimum checkpoint and log overhead per mobile host per checkpoint interval and no orphan/lost messages.

Tuli and Kumar [15] introduced minimum process coordinated checkpointing scheme for ad hoc network. This scheme allows minimum number of nodes to take checkpoint and uses few control messages to produce consistent global state. Cluster based routing protocol used for the ad hoc network hence containing cluster head and ordinary nodes, additionally cluster head sends aggregated data information to base station which saves cluster head state periodically, If some fault is detected or a cluster head fails, then its failure is detected by the base station (BS) and responsibility of the cluster head is assigned to a new node in the cluster. If a transient fault occur at the cluster head, the cluster can quickly recover from it using checkpointing this approach addresses recovery process for cluster head and ordinary nodes without having any additional overheads.

Men et al. [16] presented a checkpointing and rollback recovery scheme which is best suitable for the cluster-based multi-channel ad-hoc wireless network management where the MHs are controlled by the cluster head to take the checkpoints in checkpoint beacon intervals and in case of failure rollback to a consistent state. Every beacon interval consisting of different phases depicts for checkpointing and recovery scheme capable of handling ordinary host transient failures and also the crash of gateway which are present between two neighboring clusters. Beacon packet is used by CH which contains clock data, traffic indication messages and data window and also holds some other variables such as index of ordinary node queue, checkpoint and reply messages. There is no domino effect in the recovery scheme and the recovery of the failure process can start from its latest local consistent checkpoint then messages are restored and repeated messages for rollback will be discarded to make gateway consistent.

Bhalla [17] asserted global snapshot for host recovery that helps in independent dependency tracking in a mobile ad-hoc computing environment that without any message overheads and delays finds the consistent global state. The process to perform recovery computation is to inform all other processes of its recovery state and then each process verify their highest consistent state, if not satisfied maps the processes to be rolled back to the optimal recovery state. This algorithm assures for each node failure n-1 messages are sent within the system of n nodes. No orphan or no lost messages exist after the failure recovery.

Cao and Singhal [18] introduced the concept of “Mutable Checkpoint”. The Mutable checkpoint is neither a tentative checkpoint nor a permanent checkpoint to design efficient checkpointing algorithms for mobile computing system. We can save these checkpoints anywhere (e.g. in the main memory or local disk of MHs). The overhead of transferring large amount of data to stable storage at MSSs over the wireless network can be avoided by taking a mutable checkpoint. This technique tries to minimize the number of mutable checkpoints. This approach is a non blocking algorithm which avoids the avalanche effect and forces only a minimum number of processes to take their checkpoints on the stable storage.

Neves and Fuchs [19] in their work described a checkpoint protocol which is well adapted to the characteristics of mobile environments. The protocol saves consistent recoverable global states easily without any need of exchanging messages. Whenever a local timer expires a process creates a new checkpoint. The checkpoint timers are kept approximately synchronized by using a simple mechanism. Mobile host locally saves soft checkpoints, and stable storage stores the hard checkpoints. The protocol adapts itself and changes behavior according to different networks by changing the number of soft checkpoints that are created per hard checkpoint.

Table1: Comparison of different checkpointing algorithms for Mobile Ad Hoc Network

Author	Checkpointing Approach	Advantages	Disadvantages	Channel	Stable storage Location
Ono Masakazu and Higaki Hiroaki in [9]	Uncoordinated	• Can be used in mission-critical network applications • Communication overhead for taking global checkpoint is reduced.	• It has additional overheads and control messages associated with it as Checkpoint request message is delivered by flooding.	FIFO	Neighboring mobile devices
A.K.Singh, P .K. Jaggi, [10]	Uncoordinated	• Staggered approach to avoid simultaneous contention for resources. • Successfully handles the multiple failures.	• Suitable only for Small sized message logs.	Non FIFO	Own Memory
Saluja and Kumar [11]	Coordinated	• Minimises useless checkpoints by maintaining exact dependencies among processes. • Ensures Zero blocking time. • Piggybacks checkpointing sequence number and dependency vector on to the normal messages.	• Dependency vector of mobile hosts are maintained at CHs so memory space of Cluster head is wasted.	FIFO	Local mobile support stations (MSS) at cluster head (CH)
Morita and Higaki [12]	Hybrid(Coordinated and uncoordinated)	• Supports both mobile and fixed stations. • Reduced communication and synchronization overheads.	• Overheads may be incurred due to large amount of processing as it involves two different types of checkpointing schemes.	FIFO	Local Mobile support station(MSS)
Juang and Liu [13]	Uncoordinated	• Asynchronous Recovery and an efficient rollback algorithm. • The mobile hosts need to rollback only once and can immediately resume operation without waiting for any	• It has to resend some of the lost messages after finishing the recovery algorithm which can lead to wastage of time and resources.	FIFO	Cluster Head

Volume: 6 Issue: 3 Pages: 2308-2313 (2014) ISSN : 0975-0290

		coordination message from other mobile hosts.			
Biswas and Neogy [14]	Uncoordinated	• Random mobility of cluster members and cluster heads is considered. • Reduce storage overhead of cluster head and supports for efficient recovery. • Threshold value is defined to take checkpoint if MH crosses from its cluster.	• If the node fails, the data has to be searched and retrieved for recovery along with last saved checkpoint. • This search and retrieval cost increases with increasing 'cluster-change-count' and is added to total recovery cost of a failed mobile node.	FIFO	Neighboring nodes
Tuli and Kumar [15]	Coordinated	• It does not consider useless checkpoints • The energy consumption and recovery latency are reduced when a cluster head fails. • Checkpoint is taken by the minimum number of processes.	• As it takes checkpoint for minimum number of processes, it is difficult and time consuming to decide which process should take the checkpoints.	FIFO	Cluster Head
Men et al [16]	Coordinated	• Cluster-basedmulti-channel management protocol. • Local consistent checkpoint-two consecutive beacon interval. • Rollback recovery in one beacon interval.	• Additional power consumption and memory overhead is incurred.	FIFO	Mobile supporting stations (MSS) at cluster head (CH
Bhalla [17]	Independent	• Uses a modified cumulative dependency tracking approach for the recovery process. • And also for the generation of global snapshot.	• For recovery one message needs to be sent to each connected station to inform them about the occurrence of a failure which leads to wastage of time and bandwidth.	FIFO	Nodes own stable memory
Cao and Singhal [18]	Coordinated (Non blocking)	• Checkpoints can be saved anywhere. • Overhead of transferring checkpoint information over the network to the stable storage in Mobile Support Stations is minimized.	• May result in an inconsistency as the number of useless checkpoints in may be exceedingly high in some situations.	FIFO	Anywhere in the main memory or local disks of Mobile host(MH)
Neves and Fuchs[19]	Coordinated (Indirect)	• Uses two different types of checkpoints to adapt to the current network characteristics. • Uses time to indirectly coordinate the creation of recoverable consistent checkpoints. • Saves consistent recoverable global states without any need to exchange messages.	• As it saves two types of checkpoints there is wastage of some memory resource.	FIFO	Soft checkpoint saved locally in the mobile host, hard checkpoints in the stable storage

IV. CONCLUSION

Fault tolerance is a major research area in the Mobile Ad Hoc Networks. No doubt MANETs have a great advantage of being usable in remote areas where the wired communication media cannot reach but still there are many important issues in MANETs to be handled like network stability, low communication bandwidth, power consumption of mobile nodes, time and memory overheads, large stable storage constraints, frequent node disconnections/join and traffic load with the cluster, which makes implementation of fault tolerance techniques difficult in them as compared to distributed system since

Volume: 6 Issue: 3 Pages: 2308-2313 (2014) ISSN : 0975-0290

they do not have constraints like MANETs in them. So the algorithms are developed for less overhead, reducing number of checkpoints for saving both time and memory space by using different approaches. It can also be done that making the techniques implementable on the distributed systems also implementable in MANETs by making some negotiations. We can use a better approach for node arrangement for checkpointing process or a hybrid checkpointing strategy can be used which is a combination of two or more checkpointing schemes.

REFERENCES

[1] Zhonghua Yang, Chengzheng Sun, Abdul Sattar, and Yanyan Yang, Consistent Global States of Distributed Mobile Computations, *Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications*, LasVegas, Nevada, USA, 1998.

[2] Andrea J. Goldsmith, Stephen B. Wicker, Design Challenges for energy constrained ad hoc wireless networks, *IEEE wireless Communications*, 2002.

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

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

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

[6] Guohong Cao, Mukesh Singhal, On Coordinated Checkpointing in Distributed Systems, *IEEE Transactions on parallel and distributed systems, Vol. 9*, No. 12, 1998.

[7] 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, 34(3*), 2002, 375-408.

[8] Franco Zambonelli, On the Effectiveness of Distributed Checkpoint Algorithms for Domino Free Recovery, *IEEE Proceeding of HPDC-7*, Chicago, 1998.

[9] Masakazu Ono, Hiroaki Higaki. Consistent Checkpoint Protocol for Wireless Ad-hoc Networks, *The International Conference on Parallel and Distributed Processing Techniques and Applications*, Las Vegas, Nevada, USA, 2007, 1041-1046.

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

[11] K. Saluja and Praveen Kumar. Transitive Dependencies Tracking in Minimum-Process Checkpointing Protocol for Mobile Ad hoc Networks, *International Journal of Computing Science and Communication Technologies, Vo. 4*, No. 1 (ISSN 0974-3375) , 2011.

[12] Y. Morita and H. Higaki, Hybrid Checkpoint Protocol for Supporting Mobile-to-Mobile Communication, Information Networking, *Proceedings 15th International Conference* on, 2001, 529 – 536.

[13] T.Y.T. Juang and M. C. Liu, An Efficient Asynchronous Recovery Algorithm In Wireless Mobile Ad Hoc Networks, *Journal of Internet Technology, Vol. 3*, No. 2, 2002, 143-152.

[14] S. Biswas and S. Neogy, Checkpointing and Recovery using Node Mobility among Clustersing Mobile Ad Hoc Network, *Advances in Intelligent Systems and Computing, Vol. 176*, 2012, 447-456.

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

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

[17] S. Bhalla, Independent Dependency Tracking in a Mobile Adhoc Computing Environment. Communication System Software and Middleware, *First International Conference on Comsware*, 2006, 1-4.

[18] G. Cao and M. Singhal, Mutable Checkpoints: A New Checkpointing Approach for Mobile Computing Systems, *ACM Symposium on Principles of Distributed Computing*, 1999.

[19] N. Neves and W.K. Fuchs, Adaptive Recovery for Mobile Environments, *Communications of the ACM, vol. 40*, no. 1, 1997, 68-74.