IJCTA, 10(20), 2017, pp. 153-169 ISSN: 0974-5572
© International Science Press

Cross Layer Enhanced Mobile Routing for Maximizing Stability in Manet

Sreeleja* and Bhavani**

ABSTRACT

The Mobile Ad Hoc networks offer new challenges and objectives for tolerating faults since these networks operate in wireless mode. The earlier studies achieved tolerance towards the faults with or without the information related to checkpoint mechanisms. Due to the presence of high dynamic nodes there might occur a possibility that the attackers might degrade the system performance and can cause damage to the entire connections within the networks. A technique called cross layer based on enhanced secure checkpoint is proposed to attain fault tolerance and authentication. The cross layer enhances the lifetime of the network and also its performance. The proposed technique makes use of diskless checkpoint in order to enhance the memory and static backup for the autonomous agents. The checkpoints are initialized based on the sequence numbers and intervals for checkpoints. The suggested technique also overcomes the failure and failure recovery by utilizing rollback and recovery systems. The failure recovery systems make use of sessions. Simulations are performed for efficient checkpoint and evaluated with earlier schemes of ALERT, LOER and MDC for attaining better performance.

Index terms: Mobile Ad Hoc Networks, Diskless Checkpoint, Cross Layer, Failure Recovery, Link Stability.

1. INTRODUCTION

The checkpoint scheme [1] plays a major role in distributed computing. The scheme offers tolerance towards faults without requiring any additional efforts from the programmer. It serves as a picture of the present state of a process which stores the required information into a non – volatile storage so that during the failure of a volatile storage the lost information due to a process failure can be reconstructed using the information held within the non – volatile storages. For a process to communicate with other processes through messages rolling back a process might cause some irregularities.

During a process failure it is the last checkpoint the process might have sent some messages which upon rolling back from that last checkpoint creates orphan messages i.e. the messages whose events that were received gets recorded in the states of the target processes but the sent events will appear lost. Consequently the messages that are received during the rollback time can create problems. The process sending them will have no idea about re-sending these messages where the sending process will record these sent events, but the received events are lost during communication called the missing messages.

The nodes within the distributed mobile computing systems communicates among themselves through open and distributed broadcast channel which are more susceptible to security areas. Moreover the support for multi – hop communication involves that the networks have to depend on sole solutions from every mobile node, resulting in susceptible access, intrusions, eavesdropping and DOS attacks. The focus is upon protected data routing, which faces attacks that purposely disturb the execution of routing protocols and assuring possession of correct information related to topology.

In parallel the security solutions for data – links are employed as a part of wireless standards (WEP / WPA in 802.11) for offering authentication and confidentiality on infrastructure based single – hop wireless

* Research Scholar, Karpagam University.

** Professor and Head, Karpagam University, Coimbatore, India.

networks. However, the solutions proposed at MAC, routing and transport layer only covers a subset of all the possible threats.

A cross-layer design communication use routing and transport protocols to analiyze the report for security issues in all the stages of design protocol. Latest and useful applications, mobile agent systems must also provide additional features to handle the security for the agent and the host, fault tolerance and electronic commerce. Only with those security guarantees, mobile agents will be used in a wide range of applications. The focus should be made upon fault tolerance in mobile agent systems.

Moreover the security problems with the attacks projected, it is important to understand that an agent can get lost due to the network errors or the hosts. If an agent inerentes unconventionally in the network, then there is no guarantee that the next host is correctly reached by the agent and it won't become lost. For eg, SMTP- Simple Mail Transfer Protocol as a transport medium there are no mechanisms for the reliability of the transport or any recovery error handling. Even no regular mechanisms to detect this fault situation. Moreover the errors within the networks, the fault situations can be derived by the problems or due to the host breakdown.

During these failures the agents should posses a suitable mechanism for recovering such failures. Based on the error type either a network or a host can be replicated. These conflicting states of an agent system can create considerable problems which must be addressed.

In the paper, section I describes the introduction about checkpointing in MANET. The section II deals with the related works of various scholars, which is related to checkpoint protocols and schemes. The section III is devoted to the implementation of the proposed scheme. The section IV describes the performance analysis and in the section V concludes the work.

2. RELATED WORK

The contributions of various scholars are studied for analyzing the merits and demerits in order to enhance the consequences for making the system work better.

Suri [1] describes the mobile system as dynamic and fixed nodes linked to one another by the communication network. The system can possibly face limitations like restricted power supply, movement, detachment from hosts and shortage in constant storages. For minimizing the losses in computations during improvement from the node failures cyclic gathering of a reliable snapshot of the system is needed. The author proposed an efficient synchronized checkpoint protocol that is non – blocking and do not force each node to acquire local checkpoint. The proposed work was reliable by gathering global snapshots and the protocol required only less energy and storage.

Jichiang Tsai [2] et al focuses on checkpoint and rollback recovery for distributed systems. The best way to overcome the domino effect is by using checkpoints based on communication. The conventional protocols aims to assure that each checkpoint is a part of reliable global checkpoints. This scheme may introduce high overhead during run – time, possibly by the extreme number of extra forced checkpoints. The author proposed various flexible checkpoint protocols based on communication with domino effect freedom. This technique offers flexibility in trade – offs between the cost of synchronizing the checkpoints and the distance between rollbacks. It is not necessary that all the checkpoints are required to be a part of reliable global checkpoint. The analysis of overhead reveals that the suggested technique can considerably minimize the number of extra compulsory checkpoints.

Jichiang Tsai [3] addressed that the Rollback – Dependency Trackability (RDT) explains that the rollback dependencies between the local checkpoints can be traced through online employing a transitive dependency vector. The RDT is a type of RDT – PXCM paths and it is of two types, namely the U – path and the V – path. The author analyzed several properties of checkpoint protocols based on communication ensuring RDT property. Initially the author focused on an online RDT protocol, which encounters a U – path at a certain point and the communication pattern linked with the distributed computations and also it encounters a V – path over there. Furthermore, if the encountered V – path is doubled unnoticed the related V – path also gets doubled unnoticed. It is final that breaking all unnoticeably doubled U – paths is equal to breaking all the unnoticeably double V – paths for an online RDT protocol. Furthermore, the noticeably doubled U – path should contain a doubled U – cycle in the past. From the obtained results it is realized that various checkpointing protocols actually holds the same behavior for all possible patterns for which the author proposes a systematic method for contrasting the performance of online RDT protocols.

Jichiang Tsai [4] addressed that checkpoint protocols based on communication are employed for avoiding the domino effects. These protocols belongs to a category of indices for attaining better performance. The author proposed an efficient CIC protocol based on indexes. The fully informed protocol (FI) is the best index based CIC protocol because the optimal protocols require to obtain the future information. The author noticed that the improvement using these protocols is not frequently employed based on which that improvement was removed for attaining a new protocol called NMMP. The results of simulation reveal that the proposed protocol is efficient as in FI for some computing environments. Particularly the author showed that these two protocols possess the same behavior over a tree communication network, but the NMMP method piggybacks on every message control information for fixed sizes by not limiting the size.

Subba Rao [6] described that the checkpointing and message logging are the general purpose tools offered for attaining tolerating faults in distributed systems. The diskless checkpointing scheme allows repeated checkpointing without degrading the system performance. The author proposed a scheme called diskless checkpoint based on (N + 1) parity by employing a timeout technique for checkpoint programs with high reference positions. This technique enables the applications with high reference positions to take a cyclic checkpoint. The author proposed a technique by employing a novel message logging technique called partial message logging for synchronizing the checkpoints by both the sender and the receiver. The obtained results estimate the performance of the proposed technique by using a distributed simulator test – bed where the proposed scheme outperforms N + 1 parity technique.

Mehdi Lotfi [7] proposed a novel technique for blocking synchronized checkpointing by using two level checkpointing in high performance cluster computing systems. The first phase of checkpointing is a local checkpointing which is hoarded by the nodes into their local disks based on temporary failure rates. During temporary failures in the computing nodes the process can be recovered from the local disk. The second phase of checkpointing scheme is a global checkpointing where the computing nodes send these checkpoints to a highly dependable global reliable storage within the network based on the permanent failure rates. During the occurrence of permanent failures the computing nodes become useless and the process is recovered from the global storages for utilization of a new computing node. The temporary failures are more likely than the permanent failures and the number of global checkpointing is very low than the local checkpointing. Using this method synchronized checkpointing overheads are minimized and it is proportional to temporary and permanent failure rates of a cluster system.

Doug Hakkarinen [8] et al discussed that the chance that a failure can happen during executing an application is estimated to be much higher than the prevailing systems. For neutralizing these high failure rates needs aggregation of checkpoints based on disks and diskless and also algorithmic tolerance to faults. The diskless checkpoint is an efficient technique for tolerating small number of process failures over a huge parallel and distributed system. The consequent failures occurring in more than N processes are accepted using one – level Reed Solomon checkpointing technique for faults in the N similar processes where the overhead frequently raises quickly as N increases. The author proposed a N level checkpoint scheme for diskless in order to decrease the overheads in order to tolerate a parallel failures upto N processes. Every level is a diskless checkpointing for subsequent failures of ‘i’ processes. The results of simulation signifies that the proposed N level diskless technique attains low overhead for tolerating faults that one level Reed Solomon checkpointing technique for N parallel failures for processes.

Ge – Ming Chiu [9] et al described that the checkpointing for diskless is essential for tolerating faults over a distributed or parallel computing system. The author proposed a novel technique for improving neighbor based diskless checkpointing for tolerating several failures using simple checkpointing and failure reviving operations without depending on the committed diskless processors. The scheme allows each processor to hoard their checkpoints into a set of observed processors called the storage nodes for checkpoints. Here each processor employs XOR operations for saving a group of checkpoints for the processors to which it resembles a checkpoint storage node. The technique is based on a criteria for safe recovery which states the requirement for assuring that any processor failures can be revived in a single step by employing the checkpoint data stored into any of the prevailing processors until no more failures are addressed. The goal is to discover the essential and adequate conditions for assuring safe revival and a technique for modeling a checkpoint storage node set which meets the demands. The proposed technique permits the revival of failures in a shared manner using XOR operations.

Yi Luo [10] discussed that the future generation supercomputers will take over the message passing for distributed systems containing processors hundreds and thousands. With the growth in system, considerably the failure rates gets increased. In order to achieve success and for installing such huge sized systems a checkpoint mechanism which is capable of extending was implemented along with the recovery protocols. The conventional checkpointing and protocols for reviving rollback are utilized for suggesting tolerance towards faults over distributed system which are incapable to expand itself to such huge systems. The author deals with this issue and suggested an expandable group based Hybrid Optimistic checkpoint and a protocol for selective Pessimistic message logging.

Praveen Kumar [11] described the distributed system as a group of unique things that synchronizes for solving a problem that cannot be solved uniquely. The mobile computing system is a scattered system possessing processors that run on a mobile host (MH) which alters its position within the network over time. The number of processes for checkpoints are minimized for removing the initiation of mobile hosts in sleep mode of operation, to decrease battering of mobile hosts with checkpointing behavior and to preserve the restricted battery life of mobile hosts and less bandwidth of wireless channels. In least process checkpointing protocols some unused checkpoints are used. The author proposed a minimum process synchronized checkpointing algorithm for a non – deterministic distributed system which do not utilize unused checkpoints. The proposed attempts to decrease the process jamming and overheads in coordinated messages and also minimizes the failures in processes during its synchronization with others.

Men Chaoguang [13] addressed the challenges faced by the mobile computing systems for a distributed computing systems are less wireless bandwidth, repeated disconnections and shortage of constant storage in mobile hosts. The author proposed a new protocol for checkpointing for effectively minimizing the overheads in synchronization. Employing a communication vector only a few processes contribute in checkpointing where this scheme is capable of conserving the time employed for identifying the dependency tree by communicating the requests for chekpointing to the reliant processes at once. Furthermore, the technique uses a non – blocking processe because the irregularity is resolved using the piggybacking technique due to which redundant and orphan messages can be retarded. By comparing it with the conventional synchronized checkpointing technique the proposed non – blocking algorithm attains the least number of processes to take checkpoints and also reduces the latencies in checkpointing because fetches less overheads to the mobile hosts with restricted resources.

Cheng – Min Lin [14] discussed that the traditional scattered and domino effect free failure revival techniques are unsuitable for dynamic computing systems because each dynamic node is forced to have a fresh checkpoint or else several local checkpoints may be stored into a constant storage. The author proposed a new domino free recovery revival technique for aggregating the advantages of above specified two checkpointing techniques for mobile computing systems. The initial phase uses a synchronized protocol for checkpointing among the mobile supporting station. In the subsequent phase a communication based protocol for checkpointing is employed between each mobile support station and their mobile hosts. The last phase need every mobile support station should communicate a request for checkpoint to their mobile hosts which have not received any communication from mobile support station during the subsequent phase.

The obtained results contrast the proposed algorithm with both a quasi – synchronous fault revival algorithm and a hybrid revival algorithm for checkpointing for mobile computing systems. Based on the comparison the proposed system outperforms the conventional techniques in terms of overhead in checkpointing. Furthermore, the proposed algorithm holds various merits like domino free effect, non – blocking and doubled checkpoint size and expandability.

Anjin Xioang [15] et al conversed the necessity of checkpointing for hoarding the processes of particular states in a constant storage. During failure of a distributed system the system can revive to a global stability state from particular checkpoints, rather than the early states of all the processes concerned. There are many prevailing approaches are available for checkpointing for a distributed system, but they are incapable to adopt for mobile networks due to their independent characteristics like less capability for storage, restricted bandwidth and repeated disconnections.

3. PROPOSED METHODOLOGIES

Based on the investigations from the literature survey of various scholars the solution to the problem is achieved which are classified and presented below.

3.1. Checkpoint Election and Sequence ID Generation

The checkpoint election requires processing of checkpoints in a sequential manner called the checkpoint numbers. The sequence numbers of the previous checkpoint are specified using a constant variable ‘*ck*’. The instance between the current and the previous checkpoint by a process ‘*p*’ is represented using ‘l’. A process with checkpoint sequence number is represented as p_l^{ck} . For a system of ‘n’ process let $C = \{p_1, p_2, ..., p_n\}$be the set of process and $k = \{p_1 c_1, p_2 c_2, ..., p_n c_n\}$ be the set of checkpoints one checkpoint for each of the process ‘*p*’. The checkpoint ‘*k*’ is trustworthy if and only if for packet broadcast from a starting process to a ending process those information are maintained in the checkpoint ‘*k*’ if the transmission of packet $p1$ from a process pi in the checkpoint process. Here, initially the system remains constant and the system remains consistent until a failure occurred.

The checkpoint initialization phase maintains a controller for checkpoints, existing local clocks, derived databases and timestamps for initializing checkpoints.

a) The check point is delivered by the first checkpointing controller & all the working sites in the network receives an initiation message from it.

b) When the initiation message is received, each groups, say M_j, sets the current local clock value as CLC_j and makes an answers to the controller.

c) Reorganized list is created by each site $S_{i,}$ they are the sets of transaction identifiers, on-going at that time.

d) There is a deviation of each site S_i from its checkpoint. The checkpoints are forced as a set of connection between nodes that are updated by the every packet transmission *CLCi* and they are retaining in the main memory up to reach a certain limit.

The checkpoint deviation is managed by the buffer manager for a communication $\tau_k(c)$ with a higher transmission rate than *CLCi* wants to update a node location. In this event, there are two possible cases likely to occur. If that page is not found in the buffer, the database page map table is first checked. If an entry of x is found in the database page map table for any updated version, then the most recent version of z is fetched in the database. Otherwise, the data page in normal database is obtained. After updating the page, it is marked as Derived data page.

If the page is found in the buffer, $\tau_k(c)$ cannot directly update the page due to the database deviation, unless the page have not been updated up till now. However if that page has been read only, then $\tau_k(c)$ updates that page and marked as derived data page. Otherwise, the page is copied into another data page in the main memory and the updates are made on the page as fresh, with the connection between two nodes are maintained in previous neighbor node has been restructured by a communication among better transmission that *CLCi*.

3.2. Initialization of Rollback Session

The session is the interval between the start of processing a normal process and the instance of failure along with the continuation of rollback. After the process of rollback, it resume its execution either as its specific failure or response to another process failure. The operational sessions are ordered by their sequence numbers. Each process 'p' maintains a session number s_{nk} for the current operating session where s_{nk} is initialized to 1. Each time *k* resumes its execution after rolling back to an earlier checkpoint and so 'n' increments s_{np} by one. Each process takes their checkpoints independently according to its own demands.

For instance, a process may take a new checkpoint after a time '*t*' limited clock ticks are elapsed, or after sending away packets. However, in order to have better performance of the system, the strategy of taking checkpoints by each process should be governed by a common principle. For example, if in a system, the checkpoint interval for a process '*k'* is considerably larger than that of process '*l* then '*k'* could obtain a message from q near the end of its long checkpoint interval. If the message is later undone by *'l'*. The process *'p'* would be requested to restore a snapshot taken unreasonably long time ago.

```
Rollback Techniques
For each checkpoint process 'k' in the system
    initialize $c_{nk} = 0$;
Loop
    reset and start counting local clock $CLC_j$;
    increment $c_{nk}$;
    while $t_p < t_{check}$ do
    execute normal operation;
    take a checkpoint k clip;
end loop
```

3.3. Recovering the System

The algorithm for rollback recovery system works in two phases. The first phase is for a initiator rollback to recover from a failure. The second phase is for other non faulty processes to cooperate in recovering failures.

```
Procedure for System Recovery
For the recovering process p:
    increment the session number $s_{nk}$
    restore to the last checkpoint number $k\ c_l J_k$;
    send a rollback Initiating message to every process $l$;
    collect input C information messages from all other processes;
    augment(k);
    restore_ point determination(k);
    for any process q do
    send a rollback_request message from $k$ to $l$;
    restore recovery point of p;
    resume implementation;
end for
```

3.4. Determination of Restart Point

During a recovery phase, the process ‘p’ queues all non system messages. The procedure restart point determination {p} computes the set of processes that must rollback during the failure of a process ‘p’. This is called rollback set of p and is represented by RS(P). Furthermore, themethod determines the respective recovery points for processes in this set.

```
Procedure for restart point
procedure restart_point determination(p)
    RS(p) = (p);
mark $pI$ “visited”;
    for any q do
initialize the recovery point of q determined by p to $qf$;
    dfsearch(p, $pI$);
end restarLpoint_determination;
    procedure dfsearch
```

3.5. Domino Free Property

A domino effect is a failure causing of any process to have two sequential rollbacks without executing any suitable computation among the two. In the rollback recovery algorithm, when a process p is recovering from a failure, instead of just rolling back itself to the last checkpoint, p collects the input_information messages to construct the local system graph *Gp*, then invokes procedure restart point_determination (p) to determine the set of recovery points for processes that need to back up due to the interaction with p. This set of recovery points is globally consistent, since the processes that need to rollback will back up to these consistent points in one step, there is no domino effect.

For a worst case scenario, though no consecutive rollbacks will occur, a process may need to roll back to the beginning of execution in a step. The failures within the distributed systems are infrequent it is usual that the probability for the poorer case to occur are especially slight. Besides, we disagree that even in the worse case, the performance of the independent checkpointing approach may still not be all that bad compared with the coordinated checkpoint method. Suppose that this situation occurs in two systems running the same application. One system uses the independent checkpointing approach, the other uses the synchronized method.

In the autonomous checkpoint method, p and q each takes three confined checkpoints and then p fails. However, in synchronized checkpoint mechanism p and q may have to take six total checkpoints and then p fails. The reason that p and q may take six total checkpoints are as follows: when p takes *pI,* initiates a global checkpoint, q has to take a checkpoint in response to p’s initiation since there is a message exchange between p and q after q took q_0. As q takes q_1, q also initiates a global checkpoint. Since there is a message substitute between p and q after the last global checkpoint, p has to take a checkpoint in response to q’s initiation, etc.

4. CROSS LAYER DESIGN

The Fig [1] shows the infrastructure of a cross layer communication performing between the sender and receiver. In sender part, the measurement of packet dropping is evaluated and the queue length at the receiver part. The reliable transmission of end to end is successfully achieved by the transport layer and crossing network. Including this, the cross layer framework will keep on watching the link quality. Checkpoint Initialization, Session handling, Rollback and Recovery system and Restart point determination are installed in the network and transport layer to provide reliable end to end transmission. The performance of the channel is determined by the MAC layer, the data transmission gets started.

Destination contains a mobile agent at the end observes the power levels of each accepted transmission from the receiver side. After the notification received by the source mobile agent, then the transmission is immediately halts and the duration of predictable fade is determined and future transmissions are schedules consequently. The NAV-Network Allocation Vector at the adjacent is also subjected to update when they

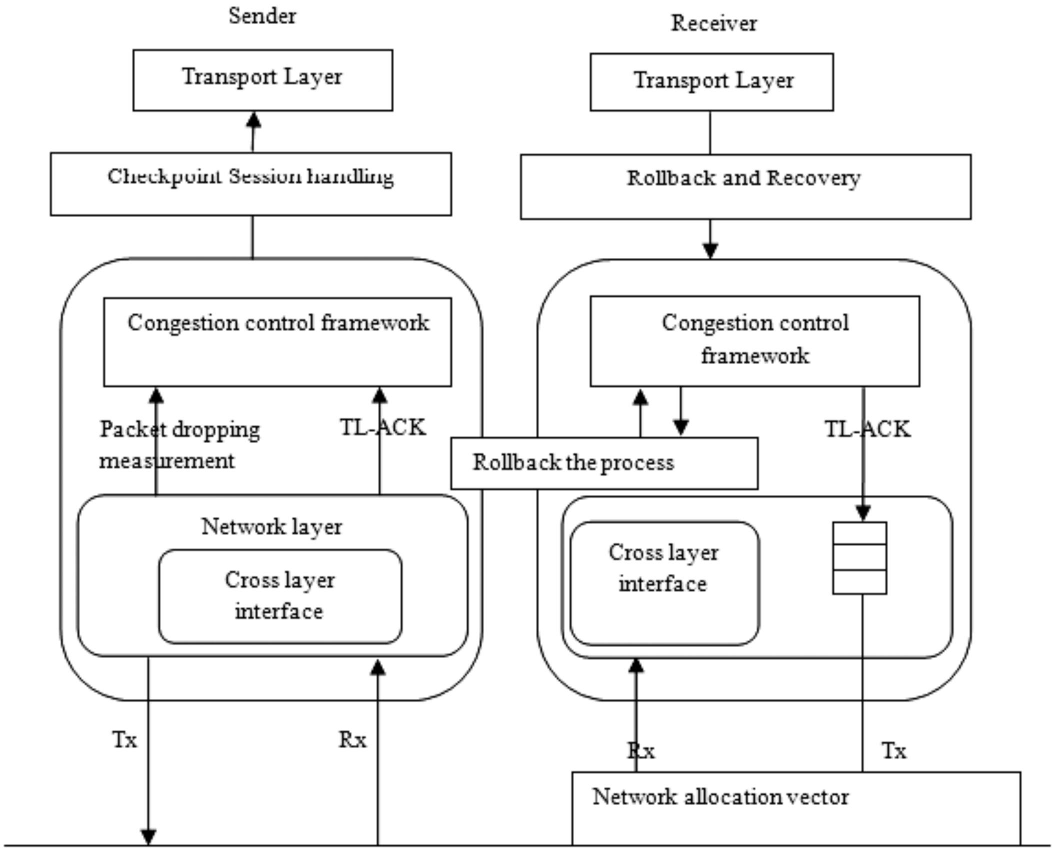
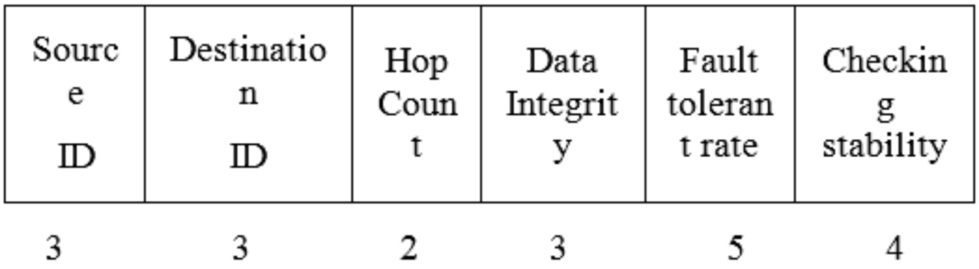


Figure 1: Cross Layer Design Infrastructure

overhear a acknowledgement whose flag bit is noticeable. The simulation results using object oriented discrete event simulator obtained indicate the cross-layer execution executes better than the layer functioning in terms of strength of signal received, delivery ratio, throughput, congestion ratio and fraction of packets dropped.

5. PROPOSED PACKET FORMAT

In the proposed packet format, the source and target node ID each hold 3 bytes. In cluster, the third field hop count regulates the n number of connection ensured to the particular node. It occupies 2bytes. The data integrity consumes 3 bytes encourages the transmission of packets which travels with high integrity from the source node to the destination node. The fault tolerant rate is verified during the route maintenance phase. It occupies 5 bytes. The last field is Checking stability. It provides steady path connection maintenance to achieve efficient transmission.



Sourc e ID	Destinatio n ID	Hop Coun t	Data Integrit y	Fault toleran t rate	Checkin g stability
3	3	2	3	5	4

Figure 2: Cross layer enhanced routing packet format

6. PERFORMANCE ANALYSIS

The proposed cross layer enhanced checkpoint protocol is integrated with the DSR protocol. The Network Simulator (NS 2.34) is used to simulate our proposed crosslayer. Simulation, 101 mobile nodes move in a 1500 meter × 1500 meter square region for 120 seconds simulation time. All agents have the same transmission range of 300 meters. The traffic for simulation is Constant Bit Rate (CBR) and Poisson traffic. The Table [1] presents the parameters considered for the simulation.

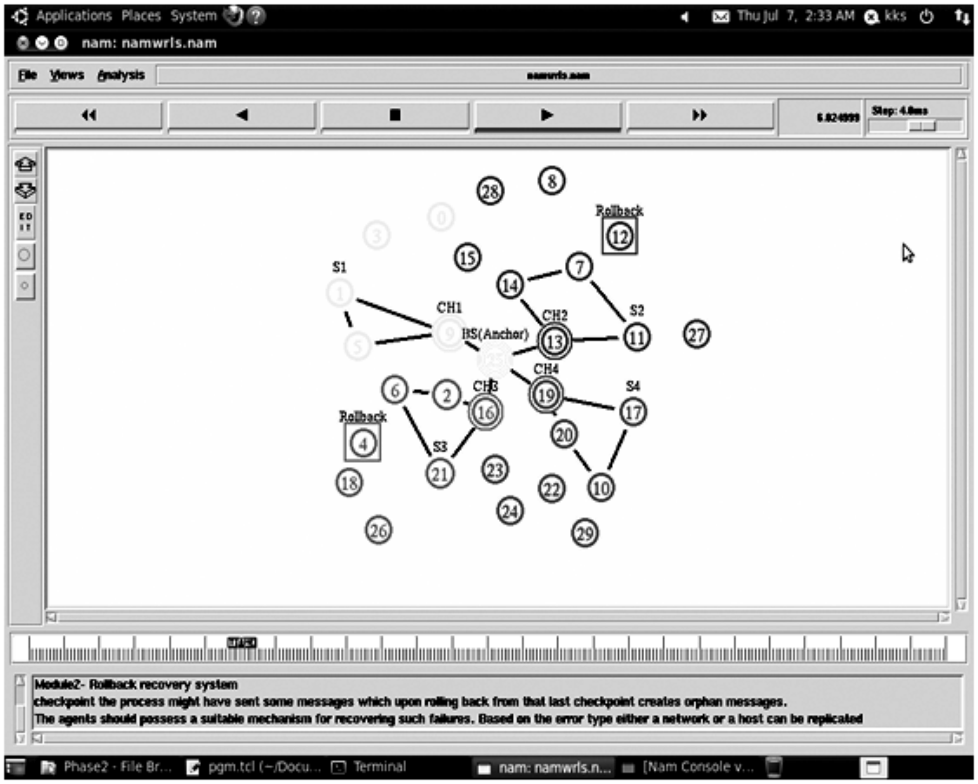
The proposed ECP is compared with the MDC [8] and LOER protocol [11] in presence of congestion environment. The Fig [3] depicts the traffic creation among the nodes. For identifying the packet loss, constant bit rate traffic is implemented. A delay is produced in the form of a packet from source agent to a destination agent via neighbor mobile agents. The source may choose the different paths to achieve a high packet delivery fraction.

Table 1
Simulation Parameters

No. of mobile agents	*101*
Area Size	1500 × 1500
Radio Range	250m
Simulation Time	120 sec
Traffic Source	CBR and Poisson
Packet Size	512 bytes
Mobility Model	Random Way Point
Protocol	Dynamic Source Routing
Pause time	5 msec
Packet Queuing	Drop Tail

Fig [4] shows the results of average node delay for varying nodes between 50 to 200. From the results, the ECP scheme has minimized delay than the ALERT, LOER and MDC schemes because of secure checkpoint protocol. The delay rate determines the improvement in the delivery rate during communication.

Fig [5] shows the results of overheads during communication for varying nodes between 50 to 200. From the results, the ECP scheme has reduced overheads in terms of node count than the ALERT, LOER



```
Module2- Rollback recovery system
checkpoint the process might have sent some messages which upon rolling back from that last checkpoint creates orphan messages.
The agents should possess a suitable mechanism for recovering such failures. Based on the error type either a network or a host can be replicated
```

Figure 3: Traffic Creation

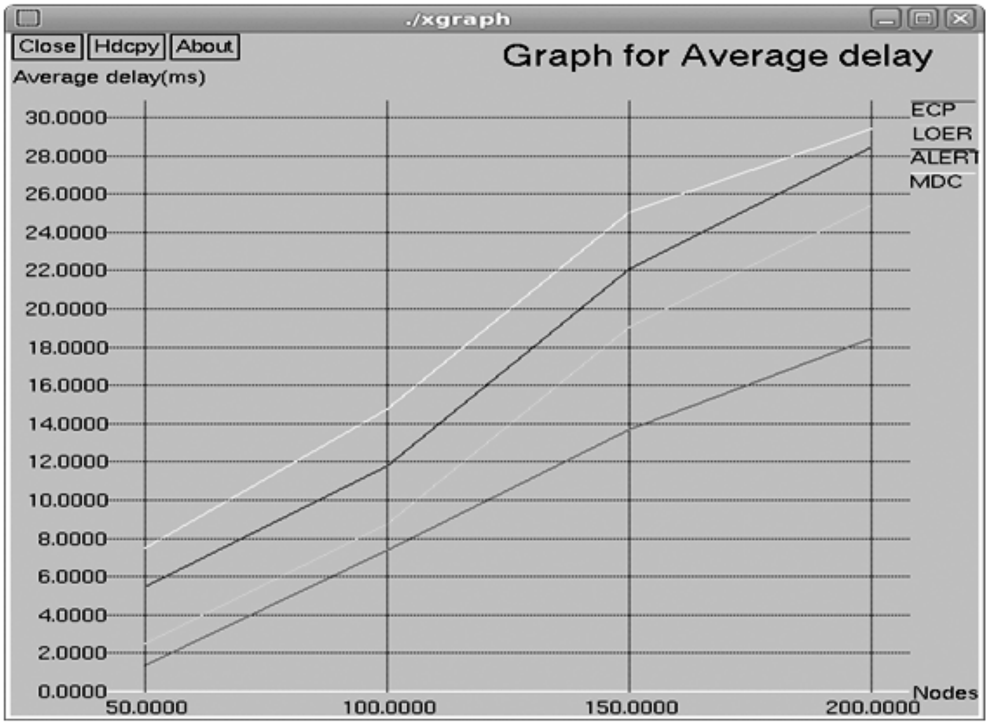


Figure 4: Node Vs Average End – to – End Delay

and MDC schemes because of secure checkpoint protocol. The reduced overhead retards the congestion, collision and improves the efficiency of the network.

Fig [6] shows the results of the packet delivery ratio for varying nodes from 50 to 200. From the results, the ECP scheme has a higher delivery ratio than the ALERT, LOER protocol and LOER schemes. The proposed scheme ECP delivers more authenticated packets as compared to the existing schemes. Fig [7] shows the

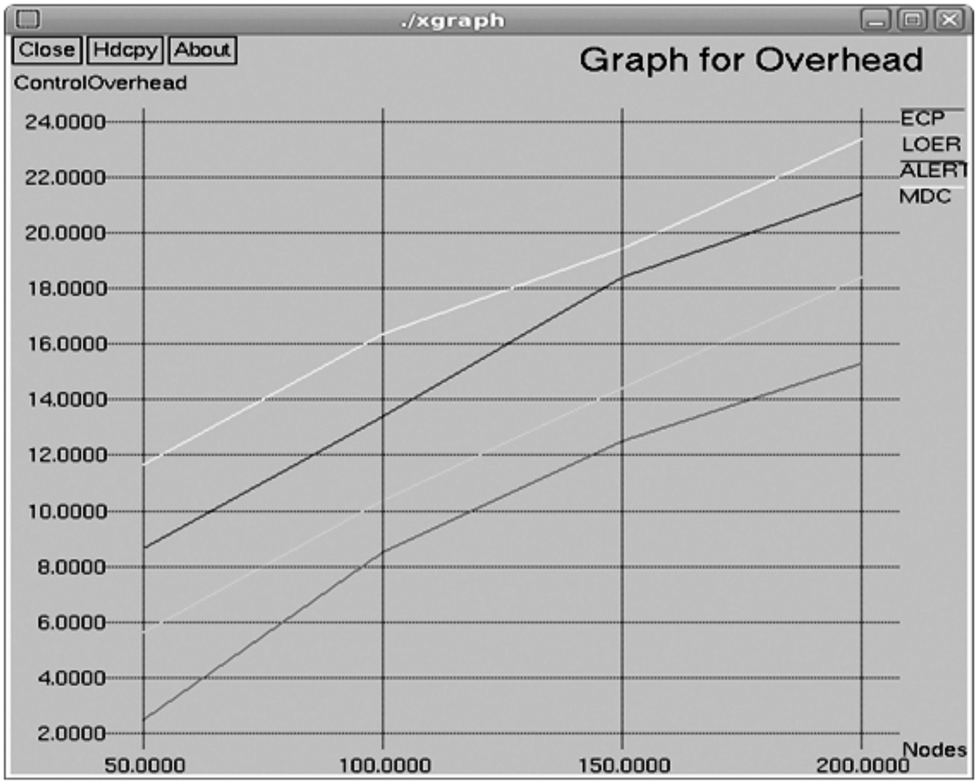


Figure 5: Node Vs Reduced Overhead

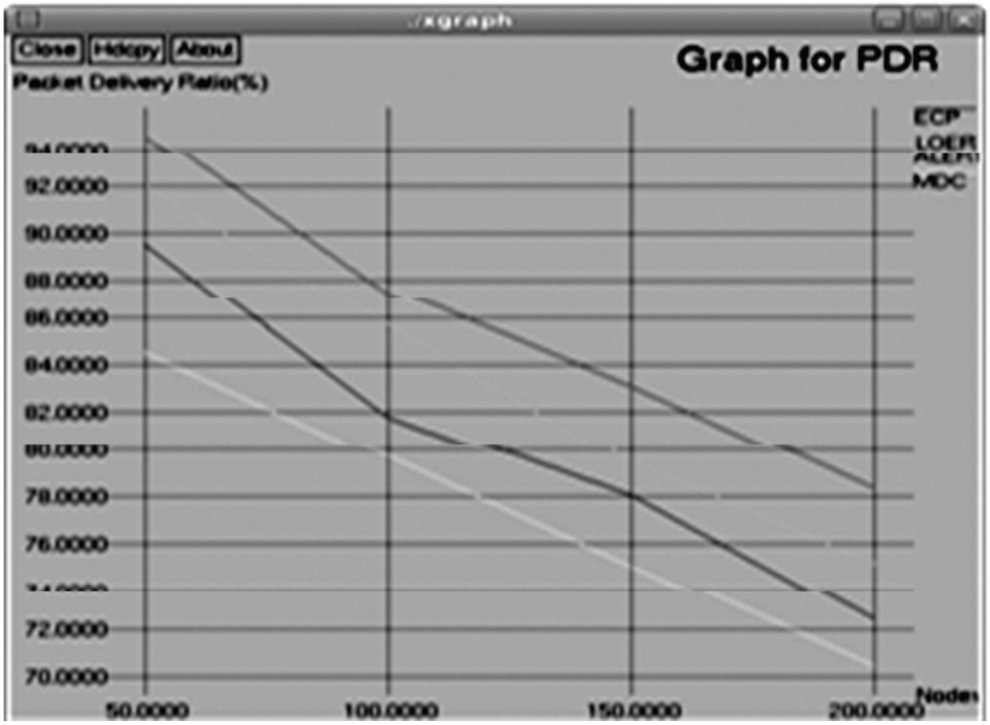


Figure 6: Node Vs Packet Delivery Ratio

results of throughput for varying nodes from 50 to 200. From the results, the ECP scheme attains higher throughput than the LOER, ALERT and LOER schemes because of secure checkpoint protocol. It is because of the optimized signature generation and verification. Improved throughput implies that how many packets and nodes are authenticated to improve the fault tolerance level. This will lead to more security.

Fig [8] shows the results of average node delay for the pause time between 5 to 20 s. From the results, the ECP scheme has minimized the delay than the ALERT, LOER and MDC schemes because of secure checkpoint protocol. The unwanted path are retarded, thus the fluctuations within the path are removed so that the delay in transmission is greatly reduced.

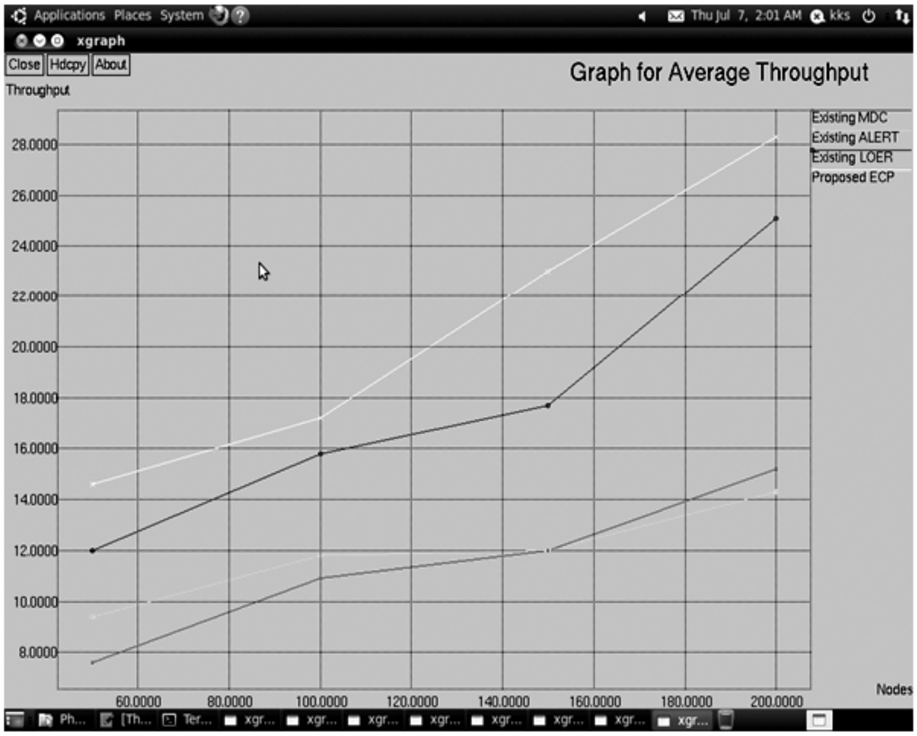


Figure 7: Nodes Vs Throughput

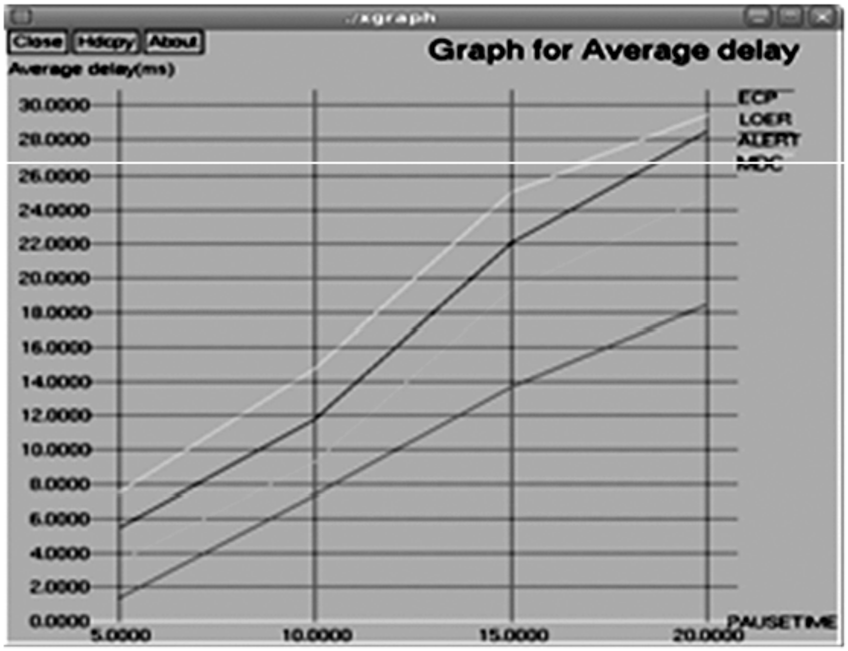


Figure 8: Pause Time Vs Average End – to – End Delay

Fig [9] shows the results of overhead for the pause time between 5 to 20 s. From the results, the ECP scheme has minimized overhead than the ALERT, LOER and MDC schemes because of secure checkpoint protocol. The transmission of unwanted messages is retarded, thus avoiding congestion and collision in between the path thus improving the lifetime of the network. Fig [10] shows the results of the packet delivery ratio for the pause time between 5 to 20 s. From the results, the ECP scheme has improved packet delivery rate than the ALERT, LOER and MDC schemes because of secure checkpoint protocol. The unwanted node communication is retarded by the proposed mechanism in the path, thus reducing the pause time.

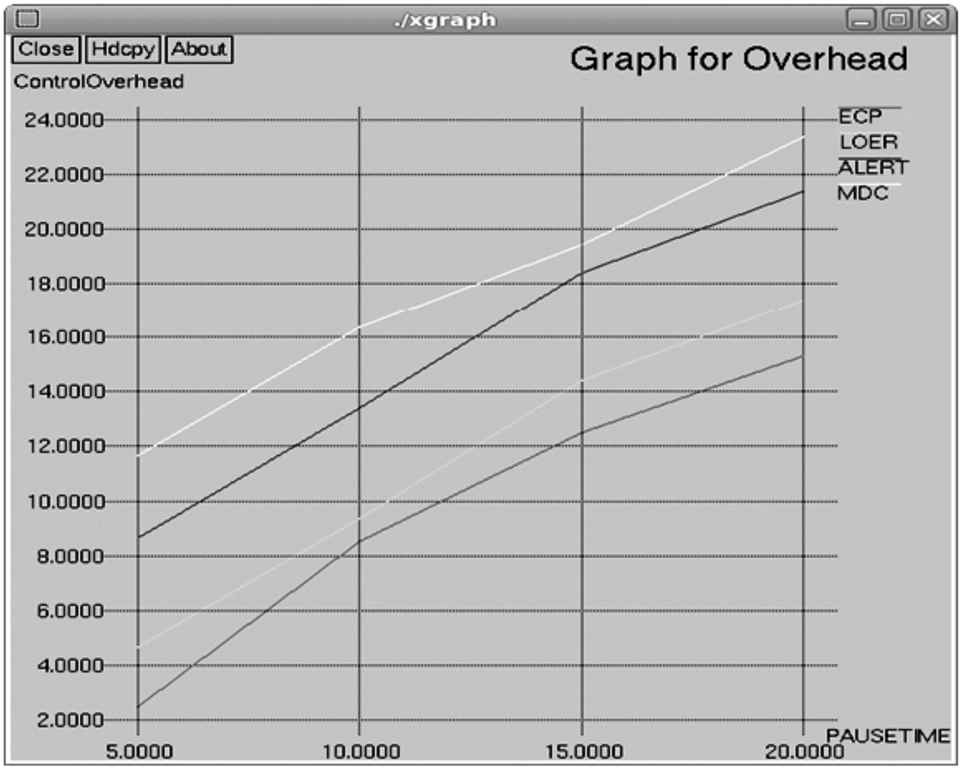


Figure 9: Pause Time Vs Overhead

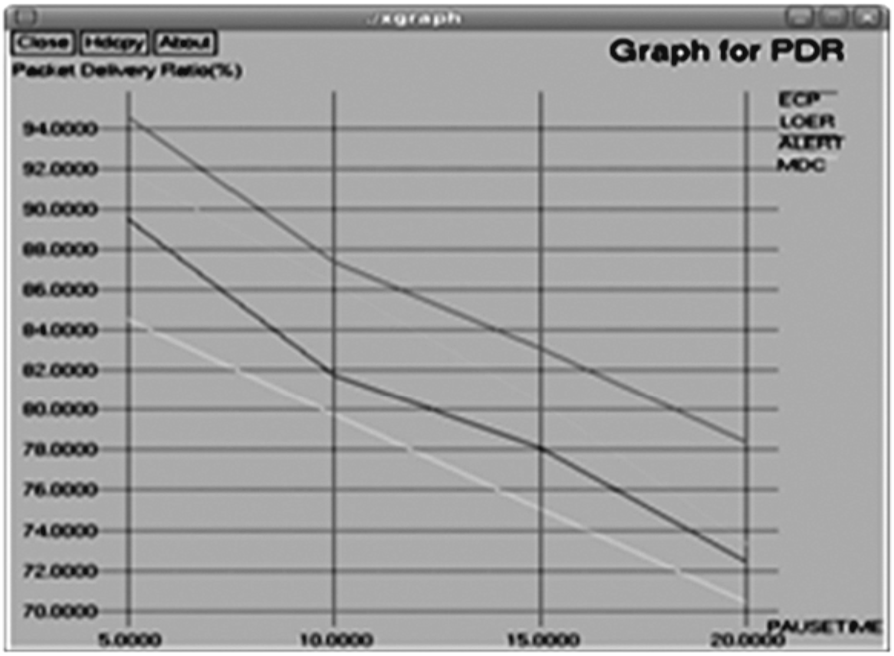


Figure 10: Pause Time Vs Packet Delivery Ratio

Fig [11] shows the results of throughput for the pause time between 5 to 20 s. From the results, the ECP scheme has improved throughput than the ALERT, LOER and MDC schemes because of the secure checkpoint protocol. The probability of network drop is decreased due to the integration of secure authentication scheme. Fig [12] shows the results of average node delay for varying Nodes. From the results, the ECP scheme has minimized the delay than the ALERT, LOER and MDC schemes because of secure checkpoint protocol. Delay of the proposed scheme is decreased because of keeping genuine packets in the path and making pause time between the packets low.

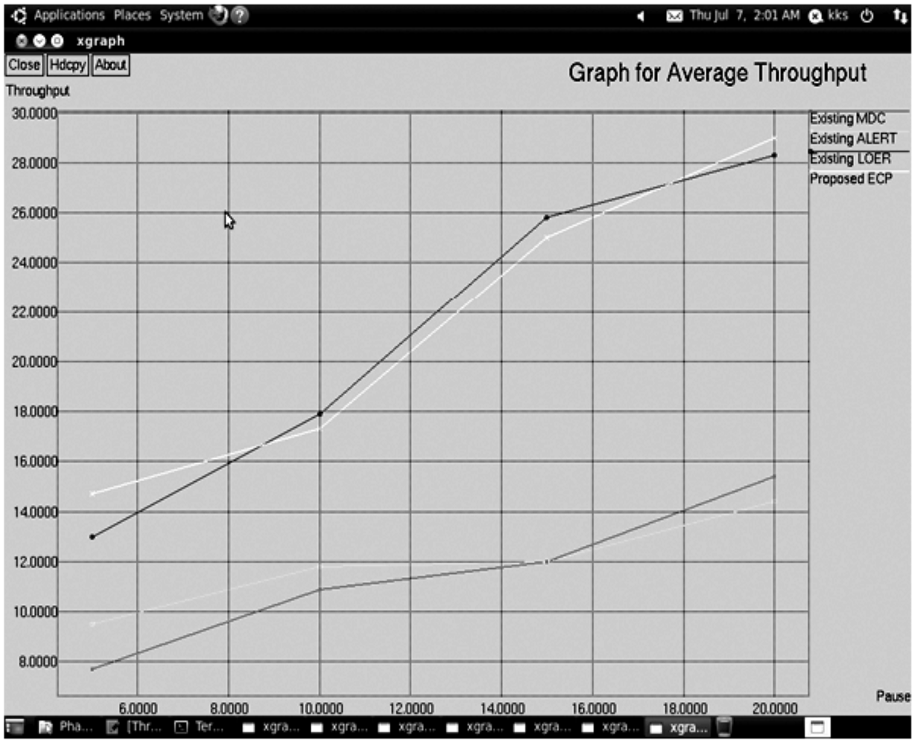


Figure 11: Nodes Vs Throughput

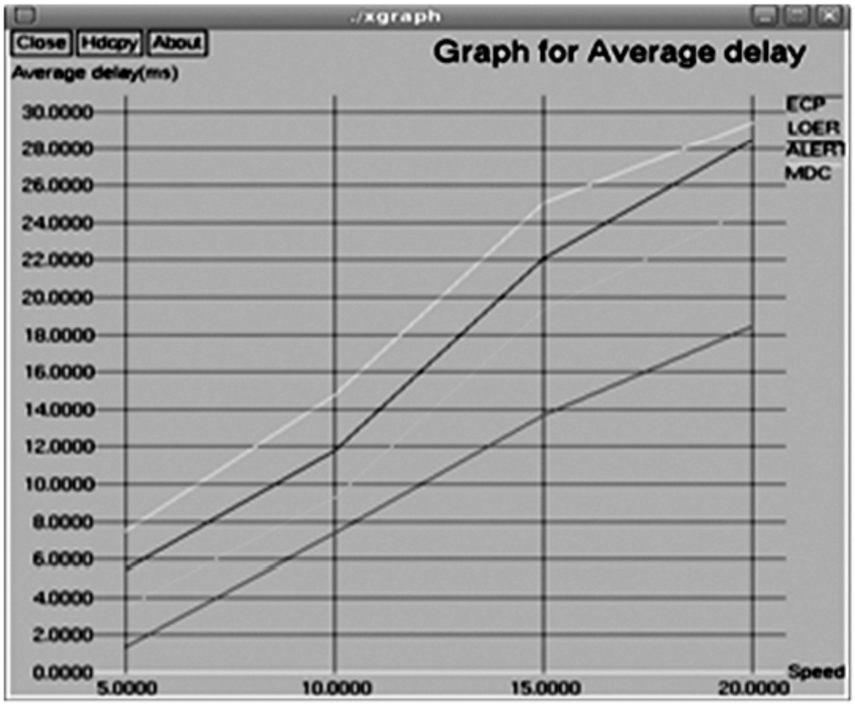


Figure 12: Speed Vs Average End – to – End Delay

Fig [13] shows the results of the packet delivery ratio for varying speed between 5 to 20 s. From the results, the ECP scheme has improved packet delivery ratio than the ALERT, LOER and MDC schemes because of secure checkpoint protocol. The packet delivery ratio of the proposed scheme is improved because of reduced delays and avoidance of malicious packets in the path.

Fig [14] shows the results of reduced overhead for varying speed between 5 to 20 s. From the results, the ECP scheme has minimized overhead than the ALERT, LOER and MDC schemes because of secure

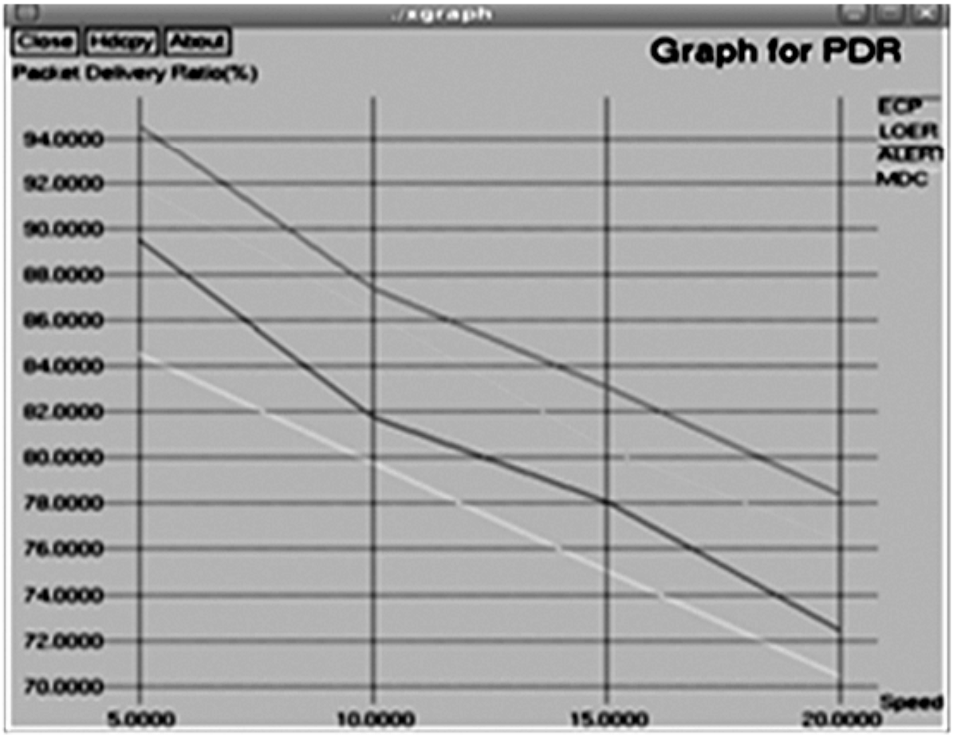


Figure 13: Speed Vs Packet Delivery Ratio

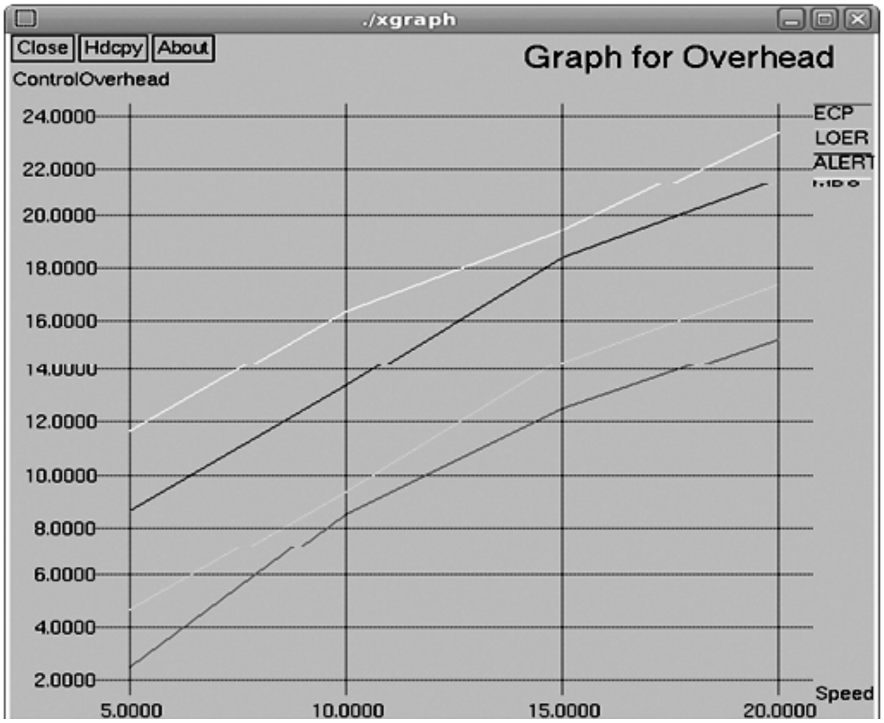


Figure 14: Speed Vs Overhead

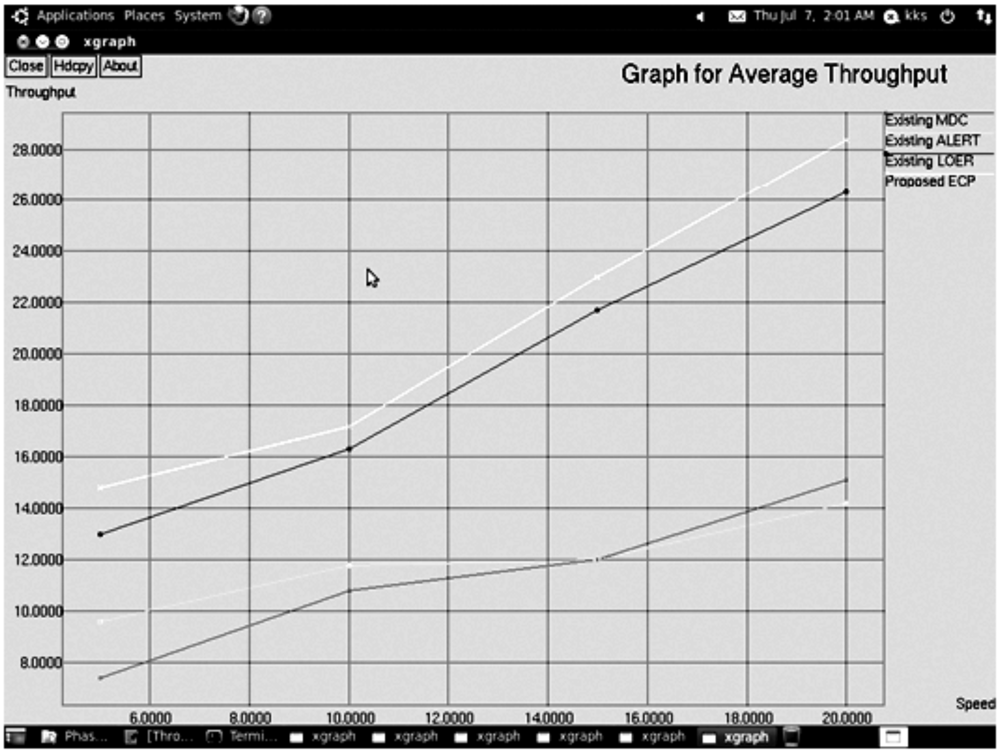


Figure 15: Nodes Vs Throughput

checkpoint protocol. The overhead of the proposed scheme is decreased because of avoiding redundant packets in the path and affects the congestion and collision within the path.

Fig [15] shows the results of the throughput for varying Nodes. From the results, the ECP scheme has improved throughput than the ALERT, LOER and MDC schemes because of secure checkpoint protocol. The throughput of the proposed scheme is improved because of reduced delays and avoidance of malicious packets and minimized overhead in the path.

7. CONCLUSION

Due to the high mobility of mobile agents, the network performance gets degraded. In order to avoid and handle fault tolerance several techniques with or without checkpoint protocol have been proposed. But, there is a shortage for data authentication and fault tolerant rate in these learning. During the occurrence of the attacks, the data is collapsed or damaged. In this paper, we have developed an Efficient Checkpoint Prototype for attaining authentication and fault tolerant for offering integrity and confidentiality among mobile agents.

In the first phase of the scheme, Cross Layer design is achieved. Here, the information transmitted to the source node depends upon the fading of channel determined from destination node. In the second phase, a secure diskless checkpoint scheme is integrated to provide more memory and high stable backup of mobile agents. In this phase, session and rollback recovery system is proposed to achieve failure recovery. By using the extensive simulation results, the proposed scheme ECP achieves the better authentication rate, packet delivery ratio, fault tolerant rate, minimum delay, less overhead and higher network lifetime than the existing schemes while varying the nodes, speed and pause time.

In future, we have intended to employ asymmetric cryptographic scheme to provide high security and data integrity. An effective location based packet forwarding protocol can be implemented.

REFERENCES

[1] P.K Suri and Meenu Satiza, “An Efficient Checkpointing Protocol for Mobile Distributed Systems”, International Journal of Latest Research in Science and Technology, Vol. 1, Issue 2: Page No.109-114.

[2] Jichiang Tsai, Chi-Yi Lin and Sy-Yen Kuo, “Adaptive Communication-Induced Checkpointing Protocols with Domino-Effect Freedom”, Journal of Information Science And Engineering 20, 885-901 (2004).

[3] Jichiang Tsai, “On Properties of RDT Communication-Induced Checkpointing Protocols”, IEEE Transactions on Parallel and Distributed Systems, Vol. 14, No. 8, August 2003, pp. 755-764.

[4] Jichiang Tsai, “An Efficient Index-Based Checkpointing Protocol with Constant-Size Control Information on Messages”, IEEE Transactions On Dependable And Secure Computing, Vol. 2, No. 4, October-December 2005, pp.287-296.

[5] Jichiang Tsai et.al, “Adaptive Communication-Induced Checkpointing Protocols with Domino-Effect Freedom”, Journal of Science and Engineering, 20, 2004, pp. 885-901.

[6] D V Subba Rao, M M Naidu & V Sai Krishna, “Efficient Diskless Checkpointing and Log Based Recovery Schemes”, International Journal of Computer Applications (0975 – 8887), Volume 5– No.12, August 2010, pp. 29-36.

[7] Mehdi Lotfi, Seyed Ahmad Motamedi and Mojtaba Bandarabadi, “Lightweight Blocking Coordinated Checkpointing for Cluster Computer Systems”, South eastern Symposium on System Theory, 2009, pp. 144-147.

[8] Doug Hakkarinen, Student Member, IEEE, and Zizhong Chen,” Multilevel Diskless Checkpointing”, IEEE Transactions on Computers, Vol. 62, No. 4, April 2013, pp. 772-783.

[9] Ge-Ming Chiu, Member, IEEE Computer Society, and Jane-Ferng Chiu, “A New Diskless Checkpointing Approach for Multiple Processor Failures”, IEEE Transactions on Dependable and Secure Computing, Vol. 8, No. 4, July/August 2011, pp. 481-493.

[10] Yi Luo, D. Manivannan, “HOPE: A Hybrid Optimistic checkpointing and selective Pessimistic mEssage logging protocol for large scale distributed systems”, Future Generation Computer Systems, Elsevier, 2012, 28, pp. 1217-1235.

[11] Parveen Kumar and Poonam Gahlan, “A Low-Overhead Minimum Process Coordinated Checkpointing Algorithm for Mobile Distributed System”, International Journal of Computer Applications (0975 – 8887), Volume 3 – No.1, June 2010, pp. 17-21.

[12] Ge-Ming Chiu, Member, IEEE Computer Society, and Jane-Ferng Chiu, “A New Diskless Checkpointing Approach for Multiple Processor Failures”, IEEE Transactions on Dependable And Secure Computing, Vol. 8, No. 4, July/August 2011, pp.481-493.

[13] Men Chaoguang, Cao Liujuan, “Low-Overhead Non-Blocking Checkpointing Scheme for Mobile Computing Systems”, Tsinghua Science and Technology, Volume 12, Number S1, July 2007, pp. 110-115.

[14] Cheng-Min Lin And Chyi-Ren Dow, “Efficient Checkpoint-based Failure Recovery Techniques in Mobile Computing Systems”, Journal Of Information Science And Engineering 17, 549-573 (2001).

[15] Anjin Xiong, Guoqiong Liao, Guoqiang Di & Jiali Xia, “ A Review of Checkpointing Strategies for Mobile Networks”, Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013), 1579-1582.

[16] Salman Khan, Nikolas Ioannou, Polychronis Xekalakis &Marcelo Cintra, “Increasing the Energy Efficiency of TLS Systems Using Intermediate Checkpointing”, IEEE Conferences, 2011, pp. 1-10.

[17] Ruchi Tuli and Parveen Kumar, “ Asynchronous Checkpointing and Optimistic Message Logging for Mobile Ad Hoc Networks”, International Journal of Advanced Computer Science and Applications, Vol. 2, No. 10, 2011, pp. 70-76.

[18] Amina Guermouche z_, Thomas Ropars _, Elisabeth Brunet _, Marc Sniry, Franck Cappello, “Uncoordinated Checkpointing Without Domino Effect for Send-Deterministic MPI Applications”, IEEE International Parallel & Distributed Processing Symposium, 2011, pp. 989-1000.

[19] Rajwinder Singh and Mayank Dave, “Antecedence Graph Approach to Checkpointing for Fault Tolerance in Mobile Agent Systems”, IEEE Transactions On Computers, Vol. 62, No. 2, February 2013, 247-258.

[20] Sangho Yi1, Junyoung Heo1, Yookun Cho1, and Jiman Hong, “Adaptive Mobile Checkpointing Facility for Wireless Sensor Networks”, Springer, 2006, pp. 701-709.