

CLUSTERING AND RECOVERY MECHANISM USING CHECK POINTING AND HIERARCHICAL APPROACH FOR MOBILE AD-HOC NETWORKS

Vikas Kumar¹, Prof. A. K. Solanki², Raghav Mehra³

¹ Research Scholar, Computer Science and Engineering, Bhagwant University, Ajmer, Rajasthan, India

² Professors, Computer Science and Engineering, BIET Jhansi, Uttar Pradesh, India

³ Associate Professor, Computer Science and Engineering, Bhagwant University, Ajmer, Rajasthan, India

ABSTRACT

In ad hoc network we used the clustering technique for check pointing and recovery mechanism for fault tolerance which is helpful in increasing the lifetime and consistency of the network. In network we place checkpoints to keep network free from failures time to time. If the failure occur the rollback mechanism from checkpoints will occur which preserve the global consistency. In this paper we focus on the selection of cluster head which is used in the check pointing and recovery mechanism to remove faults and start work again quickly. The study and work on algorithms for check pointing for ad hoc network are going on from years. In this paper, the work is shown on clustering, recovery algorithm and security for the ad hoc networks. In this for cluster head selection we used the genetic algorithm so that we get optimized cluster head. Security issue is also very important. In this paper we also focus on the performance and security of the network by using cluster trust table for finding malicious node in the network. The simulation analysis shows that proposed algorithm increases the lifetime with better system performance by using checkpoints and clustering technique.

Keyword :- Check pointing, Cluster, Recovery, Fault tolerance, Consistency etc.

1. INTRODUCTION

A Mobile Ad hoc network (MANET) is a group of mobile nodes with self organizing protocol that create a temporary network without any centralized management or infrastructure. These networks are dynamic in nature and have high security threats. For removing some of the pitfalls in the network we implement the algorithm which we discuss in the next section. Our main aims in employing check pointing for high performance computation and include the following functions:

1.1 Clustering

Clustered networks reduce the communication between the nodes by transmitting data within the formed clusters and reducing the number of transmissions to the base station node.

1.2 Dynamic Changes in Resources Utilization

In the national-wide mobile ad hoc network environment such as the computational grids, computation resources may join or leave the environment at any moment. Since applications running in such large-scale environment are

usually of highly importance, process migration is necessary for adaptability of the applications to the constant changes in resource utilization.

1.3 Load balancing

When load-imbalance occurs in a distributed system, processes on overloaded computers can be migrated to under-loaded computers for load balancing. In a non-dedicated environment, computers tend to be heterogeneous and privately owned. This means the privately owned machines may only be used for collaborative processing on an available basis. Competition for computing resources does not lead to guaranteed high performance. Stealing of computing cycles is more efficient method to ace parallel processing rather than competing for computing cycles in a non-dedicated parallel and distributed environment. Previous work [24] shows that process migration is a promising solution to the cycle "stealing" concept. Recent research shows process migration is also efficient for utilizing idle machines for collaborative processing.

1.4 Locality Accesses to Data

Process migration allows a process to move closer to sources of data or to acquire a specific device. In many cases, moving data to the process is not cost-effective when the data size is very large. Moving the process to the sources of data could be a better alternative in such circumstances. For example, in a large enterprise or even the internet, data is highly distributed.

1.5 Access More Powerful Resources

Processes can be migrated to more powerful node to seek faster computation or higher quality communication services. Mechanisms of checkpoint process in heterogeneous environment could also be beneficial to the following applications. Fault-tolerant computing is a common application of check pointing. Check pointing enables the execution of the code to be resumed from a previously saved process state (i.e., execution state, memory state, and external state) rather than its beginning; thus, process can be restarted at other node in a distributed environment, fault-tolerance can be achieved by either resuming the process at the same node after the fault is recovered or resuming at a new node via network. The ability to checkpoint and restart process among heterogeneous node can significantly expand resource utilizations. Other applications of check pointing which are not for performance are fault-resilience and system administration. Check pointing is applied for fault-resilient when the current host of a process has partial failure and then the process is check pointed and migrated to somewhere else.

1.6 Node Failure

The node failure occurs when processor get fail to execute in the distributed environment. Node failure can be stop if we save the state in stable storage. When failure occur, start sending message from the valid state. Node failure divided into 2 categories:

- Hard Failures-1

Hard failures consider as permanently failure or complete loss. This type of failure is non-voluntary in nature and processes stops any further actions forever such as falls, breaks, lost or stolen.

- Soft Failures-2

Soft failures do not permanently damages the system. These two distinct types of failure can be handled by using check pointing. Hard checkpoints are more reliable than soft checkpoints. Soft checkpoints are cheap in cost than hard checkpoints.

1.7 Security

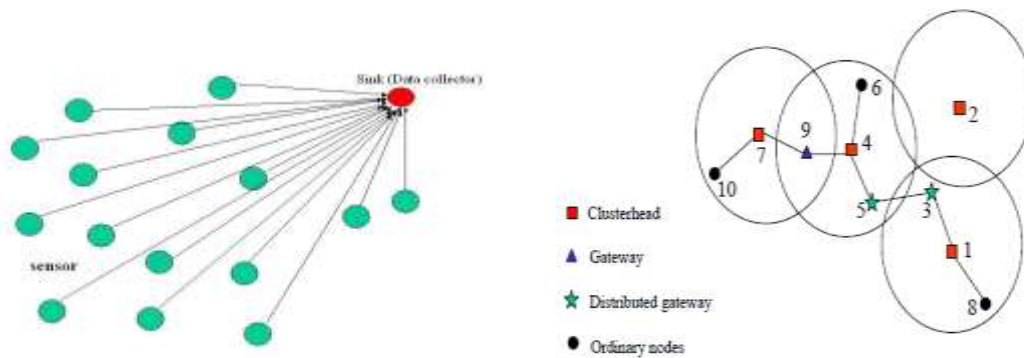
In MANET all the nodes are self configured and created network without any infrastructure. Thus the security in the MANET is weak because any node in this environment can access common radio link can participate in the network. That's why we need secure communication and route to relay the packet in the network. To secure the network and before creating the route the node should be capable to identify any node in the network before sending any data packet. The node should provide any identity and important to any other node. The identity of node should be authenticated and receiver node should not question on the identity of the node. It is important to provide security architecture to secure the MANET network.

2. METHODOLOGY FOR PROPOSED ALGORITHM

In this proposed system, mobile ad hoc network distinguishes itself from traditional wireless networks by its dynamic changing topology and the need of multi hop communication, a mobile host (MH) is free to move around and may communicate anytime with each other. When the node wants to communicate with another node and node lie within its coverage, they communicate directly in the one-hop method with each other. Route consisting of several relaying hosts is needed to forward messages from the source to the destination in a multi hop fashion. The source node sends the request to its entire neighbor to construct the route. The neighboring node check the message whether it is destination node or not. If the node is not destination nodes then it passes the packet to its entire neighbor's node until it reaches to its destination; because of this packet flooding occur in the network. To reduce packet flooding and minimize the routing of data, the cluster based method or hierarchal routing method is used in the proposed algorithm. Clustering an ad hoc network means partitioning its nodes into clusters CLs, each one with a cluster head (CH) and possibly some ordinary nodes. In the clusters, cluster head act as local coordinators. Each cluster is represented by the ID of its cluster head. For example, in cluster based distributed mobile computing systems and there are four clusters CL1, CL2, CL3 and CL4. In MANET, if one node wants to communicate with any other node in same or different cluster only through its own cluster head. In cluster technique, two type of message passing occur in the network-1) inter-cluster messages and 2) intra-cluster message. The main objective of clustering is to minimize energy consumption efficiently by multi hop communication within a cluster and also to perform data aggregation in order to decrease the number of messages transmitted. Since Normal nodes only communicate with their cluster head, which in turn, aggregates the collected information. When a cluster head fails, re-election of cluster head is performed within the cluster. For an algorithm to work feasible in an ad hoc environment, it must follow these properties:-

- Each normal node must have at least one cluster head as a neighbor
- No two cluster heads can be neighbors
- In a cluster, any two nodes are at most two hops away, since the cluster head is directly linked to every node in a cluster.

Most hierarchical clustering architectures are based on cluster head concept. In the clusters, cluster head act as local coordinators and it resembles a base station in cellular systems. In clustering architecture, the system contains two types of nodes one is cluster head represent from CH and other normal node or non cluster head represent from O and n which denotes the total number of nodes in the network.



There are five well known clustering algorithms to elect a cluster head. The first scheme is highest connectivity, in which the node with the highest degree is always selected as the cluster head by the adjacent nodes in the same cluster. The major disadvantage of this algorithm is the frequent cluster head change problem. The second scheme is the lowest-id clustering algorithm, in which each node is assigned a unique id. The node with the lowest ID is always selected as the cluster head. The third scheme is the least cluster change clustering (LCC) algorithm, which restricts cluster head changes under two conditions. The first condition is when two cluster heads come within transmission range of one another and the second one when a node becomes disconnected from any other cluster. The fourth one is distributed mobility adaptive clustering (DMAC), in which nodes are grouped using a weight-based criterion. The choice of clusters is based on a generic weight associated with each node: the larger the weight, the better the node fits the role of a cluster head. For the selection of cluster we use genetic algorithm.

Following type of communication will take place in the network

- CM sends the message data to their corresponding CH.
- Control information from BS to CM and CH.
- Control information from CM and CH to BS.
- Control information from CH to CH.

The clustering architectures provide many benefits. Clustered networks reduce the communication between the nodes by transmitting data within the formed clusters and reducing the number of transmissions to the base station node.

In the proposed system, check pointing approach and recovery of the messages that are sent but never reached is considered. The nodes of the system are divided among clusters. Nodes that need to communicate with their counterparts of another cluster send the message to their own cluster head which then sends the message to the cluster head of receiving node's cluster. The cluster head of the receiving cluster then sends the message to appropriate node.

In the proposed system, a time window of 8 seconds is considered. This time window is taken by each cluster head. Each time window or checkpoint is further subdivided into four time windows of 2 seconds each. In the first time window, the message from all the nodes of a particular cluster which want to communicate to their counterparts in another clusters, are recorded. At the end of this time window, a composite message is created for all the clusters where message need to be sent. Meanwhile, a control message is sent to all the receiving clusters, so as to initiate a connection between the receiving and the sending cluster. Now while sending a control message two possibilities can arise:

1. the control message is delivered to the receiving cluster
2. the control message is not delivered to the receiving cluster

If the control message is delivered to the receiving cluster, then the receiving cluster updates its synchronization number and waits for the application message to be sent from the sending cluster. On the other hand, if control message is not delivered then there is no such path established between the receiving and the sending cluster, neither the synchronization number is updated.

In the next time window, the sending cluster sends the messages destined for the receiving clusters. Again, there are two possibilities:

1. the application message is delivered to the receiving cluster
2. the application message is not delivered to the receiving cluster.

If the application message is delivered, then the receiving cluster checks for the synchronization number of the application message and its own synchronization number. If the Synchronization number of application message $>$ synchronization number of receiving cluster then receiving cluster updates its synchronization number

1. Synchronization number of application message = synchronization number of receiving cluster then receiving cluster has latest synchronization number. So no update is required.
2. Synchronization number of application message $<$ synchronization number of receiving cluster then received application message is out of date, discard this message.

Second case is that if the application message is not delivered. Now, if the control message is received, it means that application message is lost in the way, then the receiving cluster sends a message to the sending cluster to resend application message, thus relaxing the recovery mechanism.

The final recovery mechanism involves following log files:

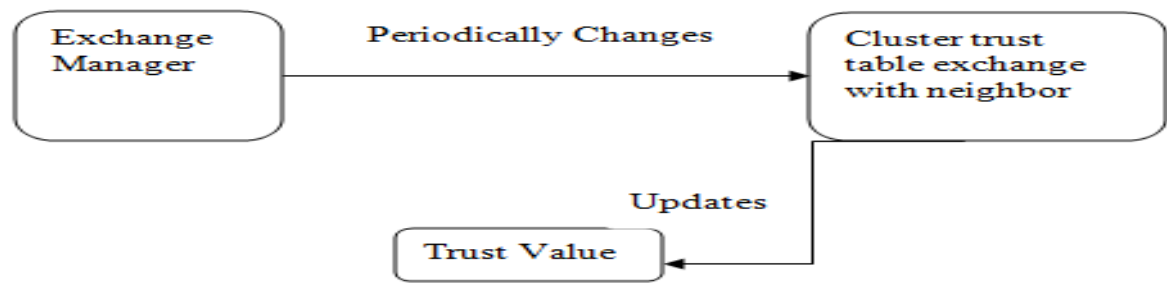
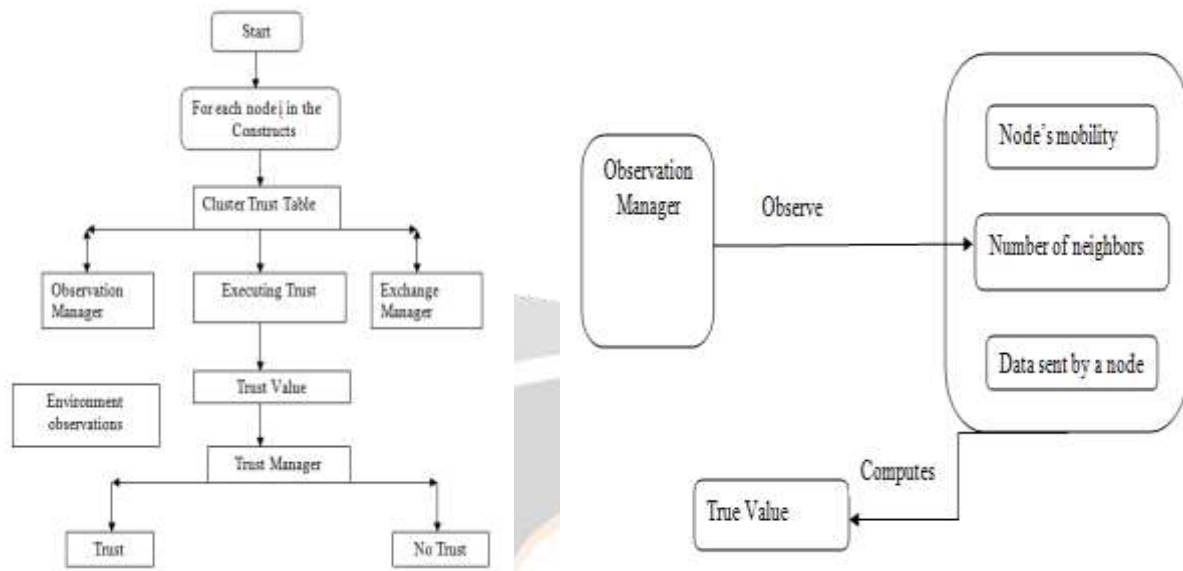
- a) Cluster Log File: This log file is maintained by each cluster which contains the information of its current synchronization number, clusters to which message is sent by this cluster.
- b) Process Log: This contains three values: process id of process to which message is sent, its synchronization number and the message itself

These logs are used by the recovery mechanism to find the messages that are lost in the communication and those need recovery.

Efficient recovery mechanism with check pointing approach is based on communication that occurs between two processes in different clusters, as a result of which dependencies are generated between checkpoints taken in different clusters.

The malicious node attacks on the network are very common and cannot be limited. We can handle the attacks if we detect and avoid the malicious node from the network and sent the message to communicate to other node from the reliable path. Malicious nodes are different from normal nodes which behave abnormally to give some unwanted effect and disturb other nodes in the network. They have higher mobility then normal nodes. In this paper we focus on the denial of service attack which busy the network with lots of unwanted messages and perform its dubious actions. If we avoid the communication via malicious node then network performance get increased. The main goal is to enhance the performance of the secure routing protocols for MANETs by detecting and avoiding malicious nodes. The proposed algorithm finds the malicious node in the network. A malicious path is a path that may contain one or more malicious nodes. The proposed algorithm named Cluster Trust Scheme is used to detect the malicious node in the Mobile Ad-Hoc networks and then avoiding the communication through them. Nodes send data through the reliable path to increase the performance of the network. The Cluster Trust Scheme is used to calculate the trust value for the nodes which help in finding the malicious nodes. It calculates the trust value of a node by directly observing the behavior of the node and then passing this value with other observations from other nodes in the network. The behavior of the neighboring nodes is used as an indication to distinguish between normal and malicious nodes. Every node in the network knows the behavior of its neighbor's node through trust value of nodes. It observes node's mobility, number of neighbors each node has, number of packets generated and forwarded by the

neighboring nodes, and the past activity of the node. Those parameters are then used to determine which nodes are misbehaving in the network. Then, the observer node builds a table called Cluster Trust Table (CTT).



2.1 Cluster head selection

Initial Phase

In initial phase fitness function is be used to select cluster head and cluster member. Choose M initial cluster heads z_1, z_2, \dots, z_M from the n nodes $\{x_1, x_2, \dots, x_n\}$. Then cluster head will be selected for each cluster member.

Step 1: Calculate fitness function

Fitness(x) is a function that returns the fitness value of the node. The fitness of a node is defined as:

$$\text{Fitness}(x) = \text{Nbrs}(x) + \text{ID}(x)/n+1$$

Where,

Nbrs(x) is a function that returns the neighbors of a node x. For instance, in figure 2, Degree of Nodes 1, 2, 3 and 4 are 2, 0, 2 and 3

ID(x) is a function that returns the id of a node x. Each node in ad hoc network has a unique id. The ID number of nodes 1, 2, 3, and 4 are 1, 2, 3 and 4.

Each mobile node has a unique number, so no nodes have same fitness function even the two nodes have same number of neighbors.

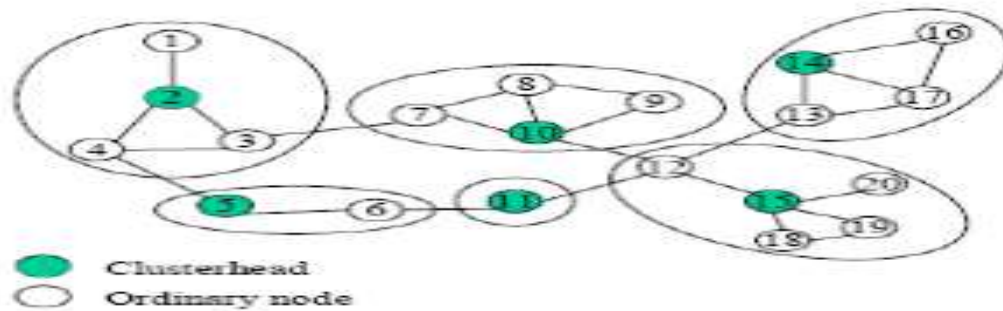


Fig-1: System Model

Step 2: After calculating this built in $F(t)$ function. All nodes sort in ascending order. For each network, network administrator decides number of cluster heads. Selected nodes send their information to the base station for the selection of cluster head.

Step 3: Selection of Cluster Head

Base station perform crossover to select the cluster head and sent the message in the network of selected cluster nodes.

2.2 Recovery Algorithm

For each Process p_k and $1 < i < n, i \neq k$ in any cluster A

if $SL_x^{ik} > RL_x^{ki}$

P_{ini} records SN of each receiving process $(R_x^{ki} + 1)$ from SL_x^{ik} in $Lost-mes-p_i^k$
 //message with $SN (R_x^{ki} + 1)$ into SL_x^{ik} are the lost messages sent from p_i to p_k

P_{ini} forms the total order of all lost messages sent by every $p_i, i \neq k$ to p_k using $Lost-mes-p_i^k$ and then message log $mesg_k$ for p_k is sent.

2.3 Security Algorithm

Cluster Trust Table (CTT)

Each cluster in the network constructs a cluster trust table that keeps track of the information observed about each connected cluster node and all the cluster nodes in order to calculate the trust value for a node. Every cluster node sends information to their cluster head. Cluster head send the information of nodes to other nodes connected to cluster head node. Every node calculates the trust value for its neighbors as shown in the following formula:

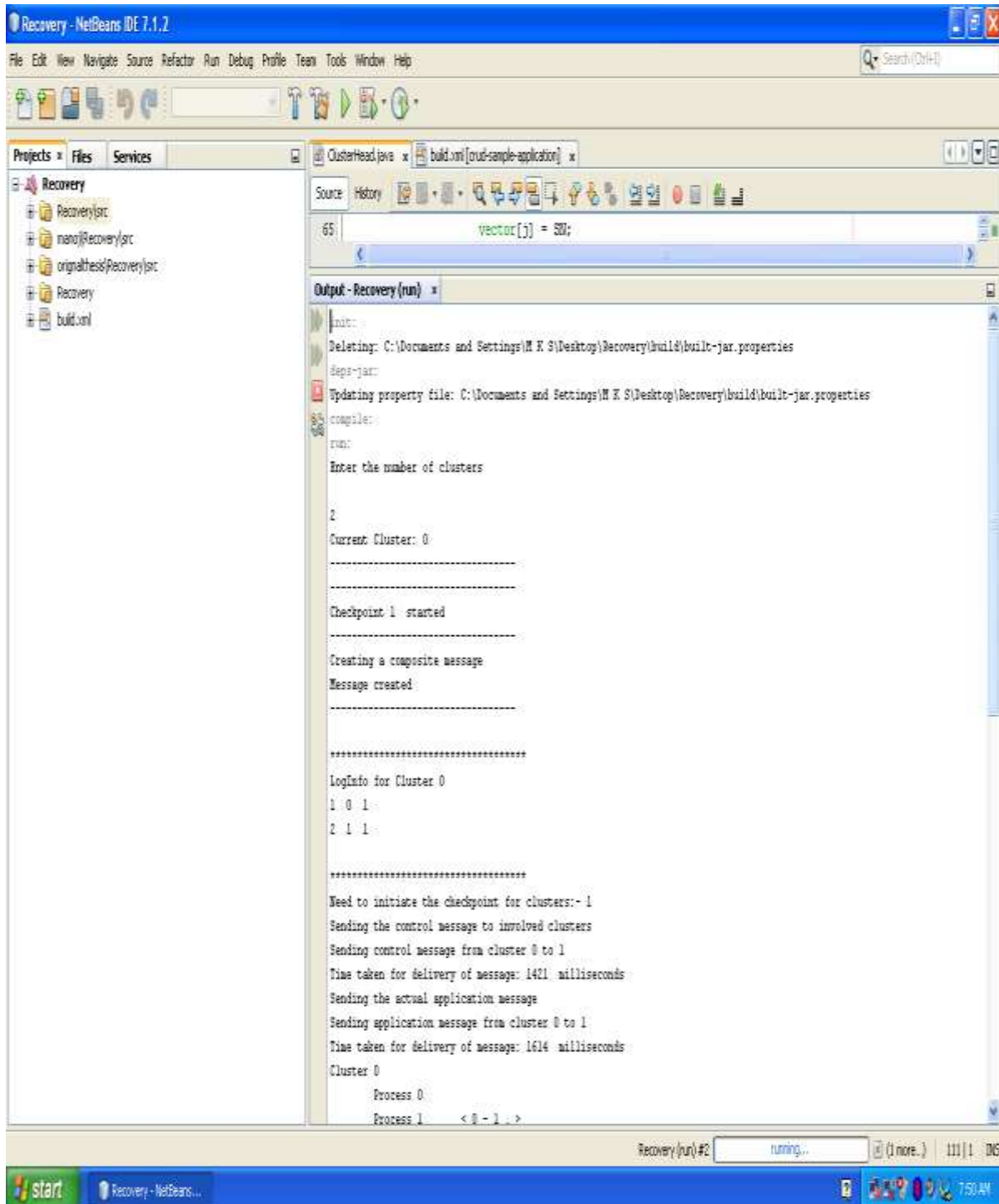
$$\text{Trust_Value} = \text{mobility} + \text{nbrs} + \text{data} + \text{old_trustvalue}$$

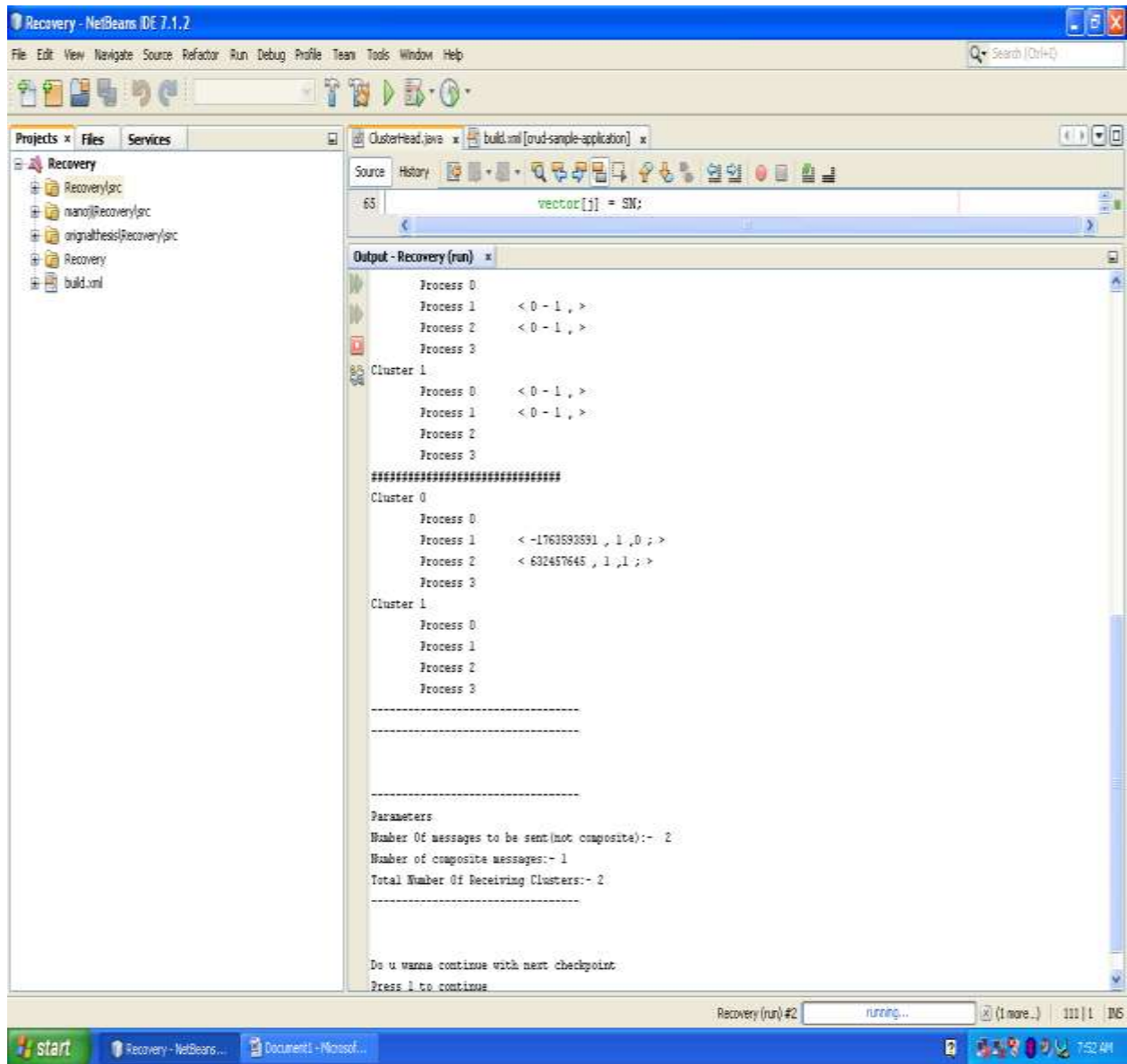
The sum of these constants equals 1. And, nbrs is the number of neighbors each node has, data is the message in the form of packets forwarded and generated by nodes and the old_trusvalue is the past value given by other nodes decide whether the node is malicious or not by checking the activity of a node in the past. Mobility is measured based on the number of changes in the one-hop neighborhood of a node. The Trust Scheme allows scattering the trust values for all nodes throughout the network. And thus, all nodes participate in deciding which nodes considered as malicious not only one node has the decision. A node sends its own table only to one hop neighbors. Every node receive new version table from its cluster head to compute the new trust values.

3. SIMULATION RESULTS

To evaluate the implementation of algorithm, following parameters have been taken into consideration: Bandwidth utilization, Number of clusters, Number of messages to be sent individually, Number of messages sent as a composite message, number of checkpoints taken, number of messages to be recovered since this thesis is an attempt to develop a recovery system which may succeed in reducing the number of messages required to be recovered.

RESULT 1



RESULT 2

Check pointing protocols require the processes to take periodic checkpoints with varying degrees of coordination. At one end of the spectrum, coordinated check pointing requires the processes to coordinate their checkpoints to form global consistent system states. Coordinated check pointing generally simplifies recovery and garbage collection, and yields good performance in practice. At the other end of the spectrum, uncoordinated check pointing does not require the processes to coordinate their checkpoints, but it suffers from potential domino effect, complicates recovery, and still requires coordination to perform output commit or garbage collection. Between these two ends are communication-induced checks pointing schemes that depend on the communication patterns of the applications to trigger checkpoints. These schemes do not suffer from the domino effect and do not require coordination. Recent studies, however, have shown that the nondeterministic nature of these protocols complicates garbage collection and degrades performance.

4. CONCLUSIONS



The main features of the algorithm are to select the cluster head using genetic algorithm and security. Each cluster maintains its cluster trust table to detect the malicious node by using with the help of trust value and also perform checkpointing and recovery method in MANET network to increase the performance of the network. The proposed technique can further be extended to save energy of nodes by less communication between nodes to nodes by using fuzzy technique and also provide strong security mechanisms. Recovery process can be roll-forward.

5. REFERENCES

- [1]. Chandy , K.M. and Lamport, L. “ Distributed snapshots: determining global status of distributed system”. ACM transactions on Computer Systems, 3(1),pp. 63-75, February 1985
- [2]. Jalote P. “Fault Tolerance in Distributed Systems”. 1st. edition of Englewood Cliffs, USA: Prentice Hall, 1994.
- [3]. Randell, B, “Fault tolerance in decentralized systems”, In proceedings of the 14th international symposium on Autonomous Decentralized systems (ISA DS'99), pp. 174-179, March 1999.
- [4]. Russell, D.L. “State Restoration in systems of communicating processes”. IEEE transactions on software Engineering, 6(2), pp. 183-194, March 1980.
- [5]. Strom, R. and Yemini, S.,”Optimistic recovery in distributed systems”, ACM transactions on Computer Systems, 3(3), pp. 204-226, August 1985.
- [6]. Elnozahy, E.N., Alvisi, L., Wang, Y.-M. and Johnson, D.B. “A Survey of Rollback-recovery protocols in message passing systems”, ACM computing surveys ,34(3),pp. 375-408,September 2002.
- [7]. Borg. A., Blau, W., Graetsch, W.,Herrmann, F. and Oberle,W, “Fault Tolerance under UNIX”, ACM transactions on Computer Systems, 7(1), pp.1-24, January 1989.
- [8]. Zambonelli, F.,”On the effectiveness of distributed checkpoint algorithms for domino free recovery”, In proceedings of the 17th international symposium on high performance Distributed Computing, pp. 124-131, Chicago, USA, July 28-31,1998.
- [9]. Bhargava, B. and Shu-Renn, L. ,”Independent Checkpointing and Concurrent rollback for recovery in distributed Systems-an optimistic approach”,n proceedings of The 17th Symposium on Reliable Distributed Systems, pp. 3-12. Columbus, USA, October 1988.
- [10]. Tamir ,Y and Sequin, C.H.,”Error Recovery in Multicomputers Using Global checkpoint”,In proceedings of the 13th international conference on parallel proceedings, pp. 32-41, Bellaire, USA, August 1984.
- [11]. Elnozahy, E.N., Johnson , D.B. and Zwaenepoel, W. “The performance of consistent checkpointing”, In proceedings of the 11th symposium on reliable Distributed Systems, pp. 39-47. Houston, USA, October 1992.
- [12]. Cristian, F. and Jahanian, F. “A timestamp-based Checkpointing Protocol for long-lived distributed computations”, In proceedings of the 10th symposium on reliable distributed systems, pp. 12-20, October 1991.
- [13]. Tong, Z., Kain, R.Y. and Tsai, W.T. “Rollback recovery in distributed systems using loosely synchronized clocks”, IEEE transactions on Parallel and Distributed systems, 3(2), pp. 246-251, March 1992.
- [14]. Koo, R. and Toueg, S. “Checkpointing and Rollback recovery for distributed systems”, IEEE Transactions on Software engineering, 13(1), pp. 23-31, January 1987.
- [15]. Netzer, R.H.B. and Jian, X. “Necessary and Sufficient conditions for consistent global Snapshots”, IEEE Transactions on Parllel and Distributed systems, 6(2), pp. 165-169, February 1995.
- [16]. Helary, J.-M. , Mostefaoui, A. and Raynal, M. “Virtual precedence in asynchronous systems: concepts and applications”, In proceedings of the 11th workshop on distributed algorithms (WDAG'97) pp. 170-194, September 1997.
- [17]. Alvisi, L., Elnozahy, E., Rao, S., Husain, S.A. and de Mel, A. “An analysis of communication induced checkpointing”, In proceedings of the 29th Annual International Symposium on Fault-Tolerant Computing, pp. 242-249, June 1999.
- [18]. Elnozahy, E.N., Alvisi, L., Wang, Y.-M. and Johnson, D.B. “A survey of Rollback- recovery protocols in message passing systems”, ACM computing surveys, 34(3), pp. 375-408, September 2002.
- [19]. Helary, J.-M. , Mostefaoui, A. Netzer, R.H.B. and Raynal, M. “Preventing useless checkpoints in distributed computations”, In proceedings of the 11th symposium on reliable distributed systems, pp. 183-190, October, 1997.

- [20]. Baldoni , R., Quaglia, F. and Ciciani, B. AVP- “Accordant checkpointing protocol preventing useless checkpoints”, In proceedings of the 17th IEEE symposium on reliable distributed systems, pp. 61-67,October 1998.
- [21]. Wang, Y.-M. “Consistent global checkpoints that contain a given set of local checkpoints”, IEEE transactions on Computers, 46(4), pp. 456-468, April 1997.

BIOGRAPHIES

	<p>Vikas Kumar is a research scholar of Dept. of Computer Science in Bhagwant University, Ajmer, Rajasthan. He has published a lot of National and International (Review and Research) paper in Computer Science field.</p>
	<p>Dr. A. K. Solanki working as a Professor in BIET Janshi (U.P). He Have passed M.Tech (CSE) and P.h.d. He has published several no. of books on field of Computer Science and he has published several number of national and International paper in Computer Science.</p>