Allocation of Replicas by Solving Selfishness in MANET

G. Tamilarasi[#], Devi Selvam[*]
[#]*PG Student (II ME-CSE),*
[*]*Asst.professor Department of CSE,*
[#] *Sri Shakthi institute of engineering and Technology - Coimbatore, India,*
[*]*Sri Shakthi institute of engineering and Technology - Coimbatore, India,*

Abstract

In MANET, most of the Replica allocation techniques are assuming that all mobile nodes cooperate fully in the network functionalities. Some nodes decide not to cooperate partially or fully. Network performance and data accessibility are affected by these selfish nodes. This paper aims to represent a Replica server which monitors and maintains the status of all mobile nodes in the network. If it finds any selfish node in the shortest path between source and destination, the replica server sends signal to that selfish node and requests it to share the loads of the heavily loaded node. The conducted simulations demonstrate the proposed approach based on proxy method which outperforms in terms accessibility of data items, cost of communication and average query delay and also it improves the network performance of MANET.

Keywords — *MANET, Selfish Nodes, Replica Server, Data accessibility, Communication cost, Query delay.*

1. Introduction

A mobile ad-hoc network (MANET) is the collection of mobile nodes that are equipped by several wireless mobile devices which are using for communication. The transmission of the data of a particular mobile node is received by all nodes within its transmission range. It is because of broadcast nature of wireless communication and help of directional antennas. Other mobile hosts located between the two wireless hosts can forward their messages, which are out of their transmission ranges in the ad hoc networks. It will effectively improve the performance of the MANET.

Each host needs to be equipped with the capability of an autonomous system due to the mobility of wireless hosts, or a routing function without any statically established infrastructure or centralized administration. Without notifying other nodes the mobile nodes can move and turned on or off. Mobility and autonomy introduces a dynamic topology of the networks and it is because of is transient nature of the end host and intermediate hosts on a communication path.

Mobile Ad hoc Networks don’t rely on extraneous fixed infrastructure and can be installed without base station and dedicated routers. This makes the nodes as ideal candidate nodes for rescue and emergency operations. The nodes in these networks have limitations in battery power and bandwidth, and each node needs the assistance from other nodes to forward their packets. The conventional protocols like WRP, DSDV, AODV and DSR are assuming that all the nodes in MANET are cooperative fully and IT always does so truthfully.

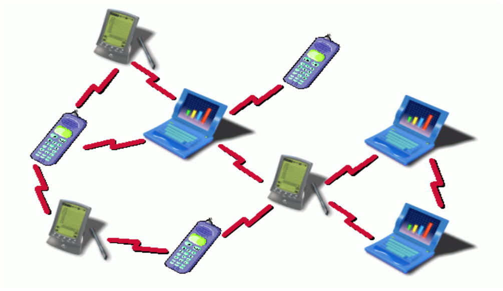


Fig 1.2. Mobile adhoc network

However the experience has shown that as the time passes there is a tendency in the nodes in an ad hoc network to become selfish. The selfish nodes are reluctant to spend their resources such as battery power, CPU memory and CPU time for others but they are not malicious nodes. Especially the problem is critical when with the passage of time the nodes have little residual power and for their own purpose they want to conserve it. Thus in MANET environment there is a strong chance to a node to become selfish. The characteristics of selfish nodes as follows in the following process ere explained below:

- Routing process Delay: The nodes are dropping routed packets or it will modify the Route Request and Reply packets by changing its TTL value to smallest value.

- Reply or sending hello message to neighbor nodes: A selfish node may not care about hello messages coming from other mobile nodes, so other nodes may not be able to detect its presence when they need it.
- Delaying the RREQ packet Intentionally: A selfish node may not pass the RREQ packet up to the upper maximum limit time. It will certainly avoid itself from routing paths.
- Delaying Data packet Intentionally: A selfish nodes may participate in routing messages but may not relay data packets.

In general, if the mobile nodes in a MANET together have sufficient memory space to hold both all the replicas and the node's own original data, replication can simultaneously improve data accessibility and query response time i.e., reduce query delay. For example, the query delay can be substantially decreased, if the query accesses a data item that has a locally stored replica.

Since most nodes in a MANET have only limited memory space, there is often a trade-off between data accessibility and query delay, For example, a node will hold a part of data items locally which are accessed frequently in order to reduce its own query delay. However, if many of the nodes hold the same replica locally and there is only limited memory space, then some data items may be missing or changed. Thus, the overall data accessibility will be decreased. So to increase data accessibility, a node should not hold the replica that is also held by many other nodes.

However, this will increase its own query delay. Since each node in a MANET has resource constraints, such as battery power and memory limitations, a node may using its limited resource only for its own benefit i.e., it acts selfishly. A node may not make its own resource available to help others, but it would like to enjoy the benefits provided by the resources of other nodes.

Such selfish behaviour in MANET can potentially lead to a wide range of problems. Existing research mostly focus on network issues on selfish behaviours in a MANET. For example, to conserve their own battery power, selfish nodes may not transmit data to other nodes. Although network issues are important in a MANET, the ultimate goal of using a MANET is to provide data services to users but replica allocation is also cruial.

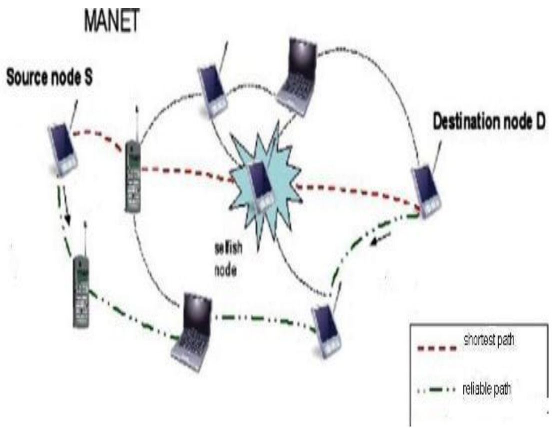


Fig 1.2 Selfishness in MANET

The technical contributions of the existing paper can be summarized as follows:

- Selfish replica allocation problem Recognization: They view a selfish node in a MANET from the perspective of data replication, and recognize that selfish replica allocation can lead to degraded data accessibility in a MANET.
- Selfish nodes detection effectively: They devise a selfish node detection method that can measure the degree of selfishness.
- Allocating replica: They propose a set of replica allocation techniques that use the self-centered friendship tree to reduce communication cost, while achieving good data accessibility.
- Simulation results: The simulation results verify the efficiency of existing proposed strategy.

2. Preliminaries

2.1. Model of the System

In the paper, assumption is made that each node has limited local storage area and acts as a service provider and a service consumer of several data items. Each node holds replicas of data items, and maintains the replicas in local storage area. The replicas can be relocated periodically. There are m nodes, N1, N2,... til m number of nodes and no central server determines the allocation of replica. Any node can freely attach and organizes an open MANET. The system is model a MANET in an undirected graph G= (IN, IL) that consists of a finite set of nodes, IN, and a finite set of communication links, IL, where each element is a tuple (N_j,N_k) of nodes in the network.

To focus on the selfish replica allocation problem, do not consider selfishness in data forwarding. The following assumptions are made,

- Each node in a MANET has a unique identifier. All nodes that are placed in a network, are denoted by N= {N_1,N_2, . . .,N_m},where m is number of nodes in a network.
- All data are having equal size, and each data item is held by a particular node as its original node. Each data item has a separate unique id, and the set of all data is denoted by D ={D_1,D_2, . . .,D_n}, where n is number of data items.
- Each node N_i ($1\leq i \leq m$) has limited memory space for replica and original data items. The size of the memory space is Si. Each node can hold only C, where $1 < C < n$, replica in its memory space.
- Data items are not updated. This assumption is for the sake of simplicity, i.e., not having to address issues about data consistency or currency.
- Each node N_i ($1\leq i \leq m$) has its own access frequency to data item D_j € D ($1\leq j \leq n$), AF_i^j. The access frequency is stable.
- Each node can moves freely within the maximum velocity.

A node Ni checks its own memory space first, when it makes an access request to a data item (i.e., issuing a query). The request is successful when a node Ni holds the original or replica of the data item in its own memory. If it does not hold the original or replica, the request will be broadcast in the network.

The request is also successful when a node Ni receives any reply from at least one node which holds the original or replica of the targeted data item that is connected to Ni within a single or multiple hops. Otherwise, it means that the request fails. When Ni receives a request to access the data item, it will either

1) Sending its original or replica if it holds the target data item (the data may go through multiple hops before reaching the requester) i.e., it serves the request. Or

2) If Ni does not hold the target data item, it will forward the request to its neighbour nodes.

4.2. Behavior of the nodes:

The work considers only binary behavioural states for selfish nodes from the network routing perspective: selfish or not (i.e., forwarding data or not). It is necessary to further consider the partial selfish behaviour to handle the selfish replica allocation. Therefore, the node is classified into define three types of behavioural states for nodes from the viewpoint of selfish replica allocation.

- *Non- selfish node:* The nodes hold replicas allocated by other nodes within the limits of their memory space.
- *Fully selfish node*: The nodes do not hold replicas allocated by other nodes, but allocate replicas to other nodes for their accessibility.
- *Partially selfish node:* Their memory space may be divided logically into two parts: selfish and public area. These nodes use their memory space partially for allocated replicas by other nodes for improving their data accessibility.

The identification of the partially selfish nodes is a tedious work, because they are not always behaving selfishly. In some situation, partially selfish node may also be considered as nonselfish nodes, because the node shares part of its memory space. In the existing paper, however, they have considered partial selfish nodes as selfish nodes, because the node also leads to the selfish replica allocation problem. Also note that selfish and nonselfish nodes perform they behave differently in using their memory space and they use same procedure when they receive a data access request.

3. Existing System

In the Existing strategy consists of three parts: 1) detecting selfish nodes, 2) building the SCF-tree, and 3) replica allocation. At a specific period of relocation, each node executes the following procedures:

- Based on credit risk scores each node will detect the selfish nodes.
- By excluding selfish nodes, each node makes its own (partial) topology graph and builds its own SCF-tree.
- In a fully distributed manner each node allocates replica based on SCF-tree.

During the query processing phase the credit risk score is updated accordingly. They borrow the notion of credit risk from economics measure the “degree of selfishness” effectively. In economics the CR is define as a Credit risk which is the calculated risk of loss due to a nonpayment of a loan by a debtor. The credit risk of an applicant is examined prior to approving the loan by a bank. The measured credit risk of the applicant indicates if he/she is creditworthy.

They take a similar approach. A node wants to know if another node is believable means that served upon request to share a memory space a replica can be paid back in a MANET. With the measured degree of selfishness, they propose a novel tree that represents relationships among nodes for replica allocation in a MANET, termed as the SCF-tree. The SCF-tree modeled by considering human friendship management. The SCF-tree-based replica allocation techniques are it can minimize the communication cost and can achieve high data accessibility simultaneously. The reason is that without forming any group or engaging in lengthy negotiations each node can detect selfish nodes and makes replica allocation at its own discretion.

- Detecting Selfish Node: The notion of credit risk can be described by the following equation:
  Credit Risk =expected risk / expected value
  In the existing strategy, each node calculates a CR score for each of the nodes to which it is connected. Each node shall estimate the selfishness degree for all of its connected nodes based on the CR score. They first describe selfish features that may lead to the selfish replica allocation problem to determine both expected value and expected risk.
- Building SCF-Tree: It was build based on human friendship management in the real world, where each person makes their own friends forming a web and manages friendship by their self. They do not have to discuss these with others to maintain the friendship. The decision is solely at their discretion. The main goal of the replica allocation techniques are reducing traffic overhead, achieving data accessibility to maximum level. If this replica allocation technique can allocate replica without considering with other nodes, as in a human friendship management, it will decrease the traffic overhead.
- Allocating Replica: A node allocates replica at every relocation period, after building the SCF-tree. Within its SCF-tree each node asks nonselfish nodes to hold replica when it cannot hold replica in its local memory space. Each node determines replica allocation individually without any communication with other nodes, since the SCF-tree based replica allocation is performed in a fully distributed manner.

The main drawback of the existing system is that still there is a problem of selfishness; they may reduce the network performance.

4. Proposed System

In the existing system there is still having a problem of selfish nodes which creates problem in accessing data and slow down the network performance. And also they are considering partial selfish nodes as selfish nodes which may not create problem sometimes so there may be an inconvenience. Also there is no server or control to monitor the replica allocation of nodes. The major disadvantage is that if any node become selfish to protect their resources there is no way to identify that selfish node. To overcome these disadvantages the following technical contribution of the paper is used.

- Designing replica server
- Monitoring nodes
- Identifying the selfishness
- Rectifying the selfishness

4.1. Designing a Replica Server

In MANET, all the nodes are handling data and they are having the dynamic counter value. The counter value is dynamic. So the size of the counter is changing dynamically. Each node will have their own counter. So the main functionality of the mobile nodes is that,

- Transmitting data
- Updating the counter value

The disadvantages of the existing strategy are solved by using the proxy replica server. The server will keep on monitoring the nodes which are allocated to that particular server and it will check whether the node is transmitting data or not.

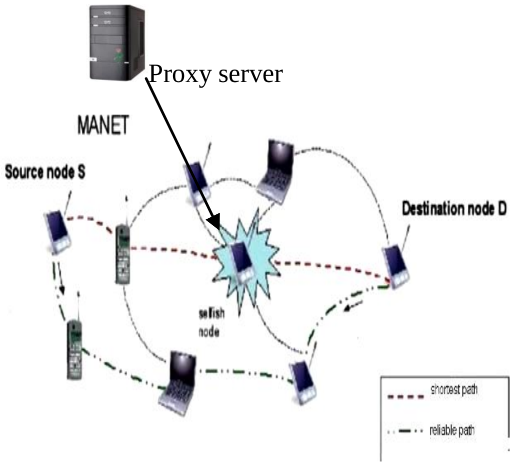


Fig 4.1 Proposed systems

If the server finds that any node is not transmitting the data or in the idle state the server will check the counter value. The counter is overflowed means the server identifies that the node cannot transmit the data.

It will maintain the previous or past value of the counter of the each node. If the counter value remains same means the server can know that the node behaves selfishly. After that the server will refresh or clear the particular mobile node's counter value AND helps to the nodes to transmit the data. So the main functionalities of the replica server,

- Allocating dynamic counter for each node
- Monitoring the status of counter value
- It maintains the status of the each node
- Finding selfishness of the nodes
- Refreshing the selfish node counter value

The above mentioned functionalities are only carrying by the server. Only functionality of the node is that it will update the counter value after sending the data. So the nodes will not get any functional overhead. The server will refresh the counter only it finds that there is no data transmission. So the network performance will not be affected by the functions of the replica server.

4.2. Monitoring Nodes

The monitoring is the process of supervising the counter value and data transmission of the nodes in the mobile ad hoc network. The intrusion detection system (IDS) for mobile ad hoc networks (MANET) consists in monitoring the nodes' behavior, in order to detect the activity of nodes which behaving maliciously. The replica server will overhear the data transmission of the network. And also it will check the counter value of the each node in the network. The previous status of each node i.e., the counter value of the nodes is maintained in the server. The table is called as status table. By using the status table the replica server can easily monitor and compare it with the previous values. Monitoring can be done by several ways. Here we refer the following two ways,

- mobile agent
- watch dog

4.2.1. Mobile Agent. Intrusion Detection System (IDS) based on Mobile Agents. The approach uses a set of Mobile Agent (MA) that can move from one node to another node within a network. This as a whole reduces network bandwidth consumption by moving the computation for data analysis to the location of the intrusion. The Mobile Agent maintains the following table to perform the computation and comparison with threshold value

SERVER NODE ID	DESTINATION NODE ID	HOP COUNT	THRESOLDPDR

The structure contains the Server node ID, destination node id that will be initiated by the source node i.e., server node. The HOP count field in the table denotes number of HOP between the source node and destination node. THRESOLDPdr signifies the number of packet drop to be considered for any node in the forward path. The forward path is generated by the AODV routing protocol. Besides, it has been established that the proposed method also decreases the computation overhead in each node in the network.

4.1.2. Watch dog. Watchdogs are used to detect selfish nodes in computer networks these are initiated by Replica server. A way to reduce the detection time and to improve the accuracy of watchdogs is the collaborative approach. A collaborative watchdog based on contact dissemination of the detected selfish nodes. Although some of the aforementioned papers introduced some degree of collaboration on their watchdog schemes, the diffusion was very costly (usually based on sending periodic messages). If one server node has previously detected a selfish node using its watchdog it can spread this information to other nodes. Formally, we have a network of *N* wireless mobile nodes, with *C* collaborative nodes and *S* selfish nodes. Initially, the collaborative nodes have no information about the selfish nodes. A collaborative node can have a positive when a contact occurs in the following way:

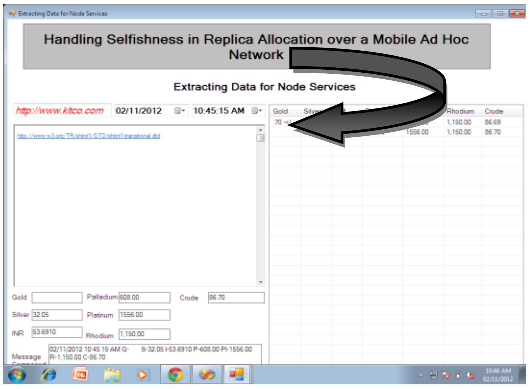
- Contact with Selfishness: one of the nodes is the selfish node. Then, the collaborative node *can* detect it using its watchdog and have a positive about this selfish node. Nevertheless, a contact does not always imply detection.
- Contact Collaboration: If two nodes are collaborative that a node has a positive if it knows the selfish node. Then, if one of them has one or more positives, it *can* transmit this information to the other node; so, from that moment, both nodes have these positives. As in the selfish contact case, a contact does not always imply collaboration.

The watch dog will collect the information and returns to the server. The information will contain counter value and address of the selfish node. The server will update the status table by using that information.

4.3. Identifying Selfishness

The counter value is monitoring by the replica server and also status of data transmission in the network. The node can update the counter value after transmitting the data otherwise the counter value will be the same. Fig 4.2 shows the simulation about identification of selfishness in MANET. So the identification of selfish nodes in the MANET will be in the following ways:

- If any node is not participating in data transmission, that can be identified by the mobile agent or watch dog means the server can identify that there is selfishness occurred in that node.
- If the counter value is same as in the status table means the server can identify that the node is behaving selfishly.
- If the counter size is exceeded or it is full, in this case also the server can identify the selfishness of the node.



Extracting Data for Node Services

Handling Selfishness in Replica Allocation over a Mobile Ad Hoc Network

Extracting Data for Node Services

http://www.kitco.com 02/11/2012 10:45:15 AM

http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd

Gold	Silver	INR	Palladium	Platinum	Rhodium	Crude
.70 -</	[illegible]	[illegible]	[illegible]	[illegible]	1,150.00	86.69
[illegible]	[illegible]	[illegible]	[illegible]	1556.00	1,150.00	86.70

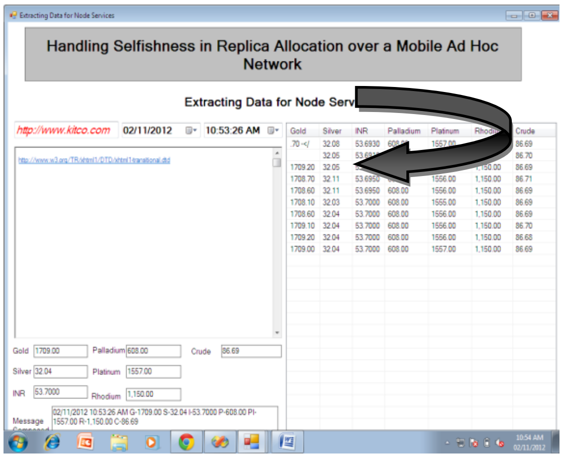
Gold:
Palladium: 608.00
Crude: 86.70
Silver: 32.05
Platinum: 1556.00
INR: 53.6910
Rhodium: 1,150.00
Message Composed: 02/11/2012 10:45:15 AM G- S-32.05 I-53.6910 P-608.00 Pl-1556.00 R-1,150.00 C-86.70

Fig 4.2 Identifying Selfishness in MANET

4.4. Rectifying Selfishness

The server finds selfish node by using mobile agent or watch dog. After finding the selfish node the replica server will decide the rectification of selfishness. Fig 4.3 shows the simulation about rectification of selfishness in MANET. For rectification,

- The server will send the signal to that particular selfish node in order to allow the nodes in the shortest path.
- It will refresh counter i.e., it clear the counter value so that the selfishness can be removed.



Extracting Data for Node Services

Handling Selfishness in Replica Allocation over a Mobile Ad Hoc Network

Extracting Data for Node Serv

http://www.kitco.com 02/11/2012 10:53:26 AM

http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd

Gold	Silver	INR	Palladium	Platinum	Rhodi	Crude
.70 -</	32.08	53.6930	608.[illegible]	1557.00	[illegible]	86.69
	32.05	53.69[illegible]	[illegible]	[illegible]	[illegible]	86.70
1709.20	32.05	[illegible]	[illegible]	[illegible]	1,150.00	86.69
1708.70	32.11	53.6950	[illegible]	1556.00	1,150.00	86.71
1708.60	32.11	53.6950	608.00	1556.00	1,150.00	86.69
1708.10	32.03	53.7000	608.00	1555.00	1,150.00	86.69
1708.60	32.04	53.7000	608.00	1556.00	1,150.00	86.69
1709.10	32.04	53.7000	608.00	1556.00	1,150.00	86.70
1709.20	32.04	53.7000	608.00	1556.00	1,150.00	86.68
1709.00	32.04	53.7000	608.00	1557.00	1,150.00	86.69

Gold: 1709.00
Palladium: 608.00
Crude: 86.69
Silver: 32.04
Platinum: 1557.00
INR: 53.7000
Rhodium: 1,150.00
Message Composed: 02/11/2012 10:53:26 AM G-1709.00 S-32.04 I-53.7000 P-608.00 Pl-1557.00 R-1,150.00 C-86.69

Fig 4.3 Rectifying Selfishness in MANET

So the selfishness can be removed fully. And the performance of the network is improved.

5. Conclusion

In contrast to the MANET viewpoint, this paper has addressed the problem of selfish nodes from the replica allocation perspective. This paper terms this problem selfish replica allocation. Our work was motivated by the fact that a selfish replica allocation could lead to overall poor data accessibility in a MANET. We have proposed a selfish node detection method and method to solve selfishness to handle the selfish replica allocation appropriately. By using Proxy replica server the selfishness of MANET nodes can be removed. This proposed system is capable of handling selfishness in small size network. Based on the server's capacity the selfishness can be handled by the server. We are currently working on the impact of different mobility patterns and improving the scalability of proposed system. We plan to identify and handle false alarms in selfish replica allocation. False alarm is the problem that the nodes are not transmitted to the destination not because of selfishness. The failure will occur due to the network failure.

References

[1] Jae-Ho Choi, Kyu-Sun Shim, SangKeun Lee, and Kun-Lung Wu, Fellow, IEEE, "Handling Selfishness in Replica Allocation over a Mobile Ad Hoc Network" *IEEE Transactions On Mobile Computing*, Vol. 11, No. 2, February 2012.

[2] K. Balakrishnan, J. Deng, and P.K. Varshney, "TWOACK: Preventing Selfishness in Mobile Ad Hoc Networks," Proc. *IEEE Wireless Comm. and Networking*, pp. 2137-2142, 2005.

[3] Shailender Gupta, C. K. Nagpal and Charu Singla, "IMPACT OF SELFISH NODE CONCENTRATION IN MANETS" *International Journal of Wireless & Mobile Networks (IJWMN)* Vol. 3, No. 2, April 2011.

[4] Yang Zhang, Student Member, IEEE, Liangzhong Yin, Jing Zhao, and Guohong Cao, Fellow, IEEE, "Balancing the Tradeoffs between Query Delay and Data Availability in MANETs", *IEEE Transactions On Parallel And Distributed Systems*.

[5] Enrique Hern´andez-Orallo, Manuel D. Serrat, Juan-Carlos Cano, Carlos T. Calafate, and Pietro Manzoni, "Improving Selfish Node Detection in MANETs Using a Collaborative Watchdog", *IEEE Communications Letters*, VOL. 16, NO. 5, MAY 2012.

[6] Debdutta Barman Roy and Rituparna Chaki, "MADSN: Mobile Agent Based Detection of Selfish Node in MANET", *International Journal of Wireless & Mobile Networks (IJWMN)* Vol. 3, No. 4, August 2011.

[7] S. Usha, *Member, IACSIT* and S. Radha, "Multi Hop Acknowledgement Scheme based Selfish Node Multi Hop Acknowledgement Scheme based Selfish Node", *International Journal of Computer and Electrical Engineering*, Vol. 3, No. 4, August 2011.

[8] Rajeev Kumar, and Prashant Kumar, "Replica Allocation Technique Based on Clusters for MANETs" *International Conference on Emerging Trends in Computer and Electronics Engineering (ICETCEE'2012)* March 24-25, 2012 Dubai.

[9] Hongxun Liu, José G. Delgado-Frias, And Sirisha Medidi, "Using A Two-Timer Scheme To Detect Selfish Nodes In Mobile Ad-Hoc Networks", *proceedings of the sixth IASTED International conference communications, Internet and Information Technology* July 2007.

[10] Zaiba Ishrat, "Security issues, challenges & solution in MANET", *IJCST* Vol. 2, Iss ue 4, Oct . - Dec. 2011 ISSN : 0976-8491 (Online) | ISSN : 2229-4333(Print).

[11] Dipali Koshti, Supriya Kamoji, "Comparative study of Techniques used for Detection of Selfish Nodes in Mobile Ad hoc Networks*", International Journal of Soft Computing and Engineering (IJSCE)* ISSN: 2231-2307, Volume-1, Issue-4, September 2011.