International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 02, February-2015

Performance Analysis of Techniques to Improve Data Accessibility in MANETs

Gayathry S S
Dept. of ECE
TIST
Cochin, India

Dr. S Perumal Sankar
Professor, Dept. of ECE
TIST
Cochin, India

***Abstract*--- Wireless networks can be infrastructure based or infrastructureless. An infrastructureless network having large number of mobile nodes, which can fulfill a wireless interconnection, can be called as a MANET. The MANET nodes should be aware of the data stored in them. No MANET communication should be a failure due to the problem of inaccessibility of data packets. In the past, many people have worked on this problem, and developed the idea of replicating data items to minimize this data inaccessibility. The process can be called as replica allocation. This paper provides a comparison between different techniques used to increase data accessibility in MANETs. Attacker nodes like selfish nodes will generate a situation of imperfect replica allocation, commonly referred to as selfish replica allocation. The problem of selfish replica allocation will not allow the network nodes to allocate replica effectively. A selfish network shows much less packet delivery ratio, in comparison with a non-selfish network. Among the replica allocation methods considered, the SCF-tree based replica allocation method eliminates the overall selfishness in the network; and excels in providing maximum data accessibility compared to the other methods.**

Keywords--- MANET; Data accessibility; Replica allocation; SAF; DAFN; DCG; SCF-Tree; Selfish nodes ; Credit Risk

I. INTRODUCTION

Mobile AdHoc networks (MANETs) or simply adhoc networks, is a type of network that comprise of nodes that can move freely and dynamically self-organize into temporary and arbitrary network topology without any fixed infrastructure support. The nodes in a MANET are connected wirelessly. These nodes can act as either source or destination node. The mobile nodes that are in radio range of each other can directly communicate, whereas others need the help of intermediate nodes to route their packets. Whether the destination can be reached in a single hop or multiple hops, routing can be done within the network by using various protocols.

In mobile ad hoc networks (MANETs), all the nodes are free to move. Wireless links existing between them complete their communication. To reach the destination, the source will take multiple hops into consideration if the target node is within the range. But if these nodes move far away, they cannot be reached by the source. Thus, the path gets broken. And, the network will be divided. ie, since mobile nodes move freely, network partition may occur.

A single network will get divided into two different parts. Then, there occurs a problem such that the nodes in one partition cannot access data held by nodes in other partitions. Or, the nodes in one partition cannot make use of the data stored in the nodes of the next partition. Thus, data availability (i.e., the number of successful data accesses over the total number of data accesses) in MANETs is lower than that in conventional wired networks.

A wonderful solution to this problem is data replication[6]. By replicating data at mobile nodes which are not the original owners of the data, data inaccessibility can be reduced. Because, this provides multiple replicas of data items in the network and the probability of finding one copy of the data is higher. Also, data replication can reduce the query delay since mobile nodes can obtain the data from some nearby nodes. However, most mobile nodes only have limited storage space, bandwidth, and power, and hence it is impossible for one node to collect and hold all the data considering these constraints. Data replication has been widely used to improve data availability in distributed systems, and we will apply this technique to MANETs.

The organization of this paper is as follows. Section II describes the process of replica allocation in MANETs. The following section III gives out the various types of replica allocation, the procedure behind them, their merits and demerits. The SCF-tree based method solves the selfish replica allocation problem by handling the selfish nodes effectively. The method used for this purpose is also discussed here. Simulation results show that a selfish network reduces the overall throughput of the network. Finally, the paper ends with the performance comparison of these 4 methods.

II. REPLICA ALLOCATION IN MANETS

Random motion of mobile nodes may lead to disconnection of networks. This phenomenon can be called as network partitioning. During this situation, the data availability in the network gets reduced. Then, measures have to be taken to tackle this problem. One such measure is the replica allocation. By implementing this property, more copies of same data will be available so that, the necessary data will not be lost even after partitioning of the network.

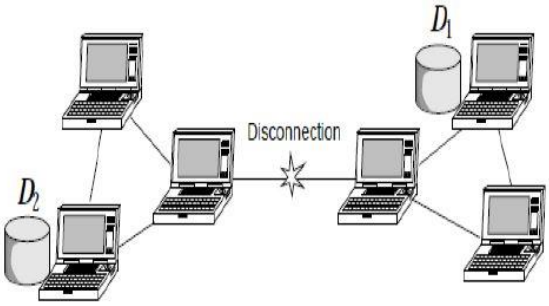

Fig 1: Network partitioning[7]

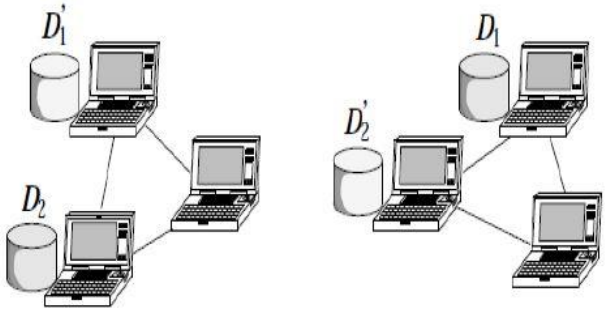

Fig 2: Effective data replication[7]

The figure 1 and 2 explains how a mobile network is being divided and how copies of data are generated and stored elsewhere[2].In Fig 1, if the radio link between two mobile hosts at the central part is disconnected, the mobile hosts in the left-hand side and those in the right-hand side cannot access data items D1 and D2 respectively. In ad hoc networks, it is necessary to prevent deterioration of data accessibility at the point of network division. Then, replication has to be done at mobile hosts which are not the original owners of the data. In Fig 1, if the replicas of data items D1 and D2 are allocated at one of the mobile hosts in the opposite networks as shown in Fig 2, every mobile host can access both data items after the network division. Hence, in order to improve data accessibility, replication can be a solution.

III. REPLICA ALLOCATION METHODS

Replica allocation can be done in many ways. The primary methods of allocating replicas are based on the access frequencies of network nodes to each data items. This section describes certain methods to allocate replicas in MANETs.

A. SAF method

SAF(Static Access Frequency)method allocates replicas of data items according to the access frequencies. A mobile node can access data items held by other connected mobile hosts, and it is more possible to share different kinds of replica among them.

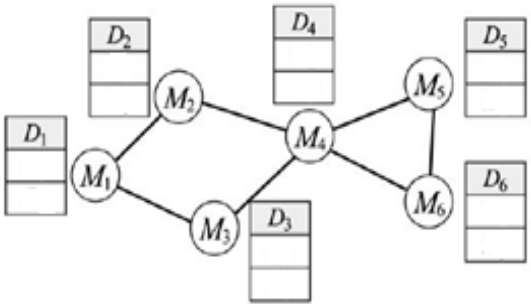

Fig 3 : Network having 6 nodes having data stored within them

Consider a network having a large number of mobile hosts. Each node stores its own original data. Also, each of these nodes has its own access frequencies to each data item stored in the other nodes, as shown in Table 1. If the mobile nodes try to access data item to which it has low frequency, it cannot access the data. So, the node will access data to which it has high access frequency. In other words, the MN (Mobile Node) accesses data in descending order of access frequencies. Here, a network having 6 nodes is shown. And, each node has its own data items. The following table shows the access frequencies of these nodes to data in the other nodes.

Table 1 shows the table of access frequencies. Now, each node tries to access data from other nodes. First, consider that M1 has to allocate replica. It has D1 inbuilt. Then, it checks the table of access frequencies. From the table, it finds that M1 has maximum access frequency to D1. It is already stored. D5 is the second highest, and D2 is the third. So, M1 allocates the replicas of D5 and D2. Similarly, other nodes also allocate data replicas according to the table of access frequencies. The output of SAF method of replica allocation is as shown below. The memory space is filled with the data items being accessed.

Table 1 : Table of Access Frequencies

Data	Mobile host					
	M_1	M_2	M_3	M_4	M_5	M_6
D_1	0.65	0.25	0.17	0.22	0.31	0.24
D_2	0.44	0.62	0.41	0.40	0.42	0.46
D_3	0.35	0.44	0.50	0.25	0.45	0.37
D_4	0.31	0.15	0.10	0.60	0.09	0.10
D_5	0.51	0.41	0.43	0.38	0.71	0.20
D_6	0.08	0.07	0.05	0.15	0.20	0.62
D_7	0.38	0.32	0.37	0.33	0.40	0.32
D_8	0.22	0.33	0.21	0.23	0.24	0.17
D_9	0.18	0.16	0.19	0.17	0.24	0.21
D_{10}	0.09	0.08	0.06	0.11	0.12	0.09

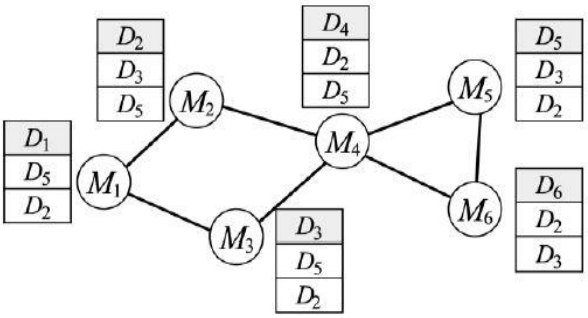

Fig 4: Relocated replicas based on SAF method

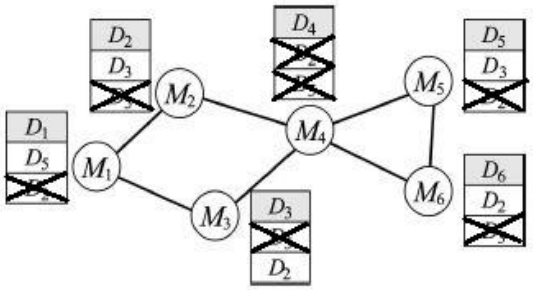

Fig 5: Cancelling duplicated replica in SAF[6]

The major disadvantage of SAF method is that, if the mobile nodes allocate replicas based only on the access frequencies, it will continuously create replicas. ie, replica duplication. So that, same replicas will be allocated repeatedly which will further generates control overhead.

B. DAFN method

The DAFN *(Dynamic Access Frequency and Neighborhood)* method is developed to solve the problem of replica duplication among neighboring MNs. ie, the formation of same replicas continuously. In DAFN method, each mobile host first broadcasts both its id and access frequency at relocation period. At first, this method allocates replica in the same way as in SAF method. Then, if there is replica duplication, the MN with lower access frequency changes the replica to another one. The elimination of replica duplication is repeated through a Breadth First Search from the MN. It is made possible by considering connected nodes. DAFN method checks whether replica is duplicated among the connected nodes. This method is used to minimize the demerits of SAF method. So, it begins with the allocated replicas of SAF, and the problem has been worked out.

Table 2 : Table of Access Frequencies

Data	Mobile host					
	M_1	M_2	M_3	M_4	M_5	M_6
D_1	0.65	0.25	0.17	0.22	0.31	0.24
D_2	0.44	0.62	0.41	0.40	0.42	0.46
D_3	0.35	0.44	0.50	0.25	0.45	0.37
D_4	0.31	0.15	0.10	0.60	0.09	0.10
D_5	0.51	0.41	0.43	0.38	0.71	0.20
D_6	0.08	0.07	0.05	0.15	0.20	0.62
D_7	0.38	0.32	0.37	0.33	0.40	0.32
D_8	0.22	0.33	0.21	0.23	0.24	0.17
D_9	0.18	0.16	0.19	0.17	0.24	0.21
D_{10}	0.09	0.08	0.06	0.11	0.12	0.09

Table 2 shows the same table of access frequencies used in SAF method. If replica duplication occurs, it replaces the replicated data by the data item to which the node is having next highest access frequency. Here, in order to avoid replica duplication, the repeatedly stored replicas are replaced by allocating the next highly accessible replicas. Then, the output will be as shown above in Fig 6.Data accessibility is expected to be higher than SAF method. But DAFN does not completely eliminate replica duplication. This is the major disadvantage of DAFN method.

C. DCG method

Differing from the DAFN method that shares replicas among neighboring nodes, the DCG *(Dynamic Connectivity and Grouping)* method shares replica of data items in several groups of mobile nodes. The DCG method creates groups of mobile nodes that are bi-connected in an ad hoc network. The group is not being divided even if one mobile node is disconnected from the network.

In DCG method, each mobile node broadcast its host id and its access frequency along with data items to other nodes. By using the broadcasting information every node identifies its bi-connected nodes. The access frequency of each group is calculated by adding the access frequencies of all the mobile nodes in that group. According to the overall access frequency of the group, replicas of data items are allocated till the memory of all mobile nodes in the group becomes full.

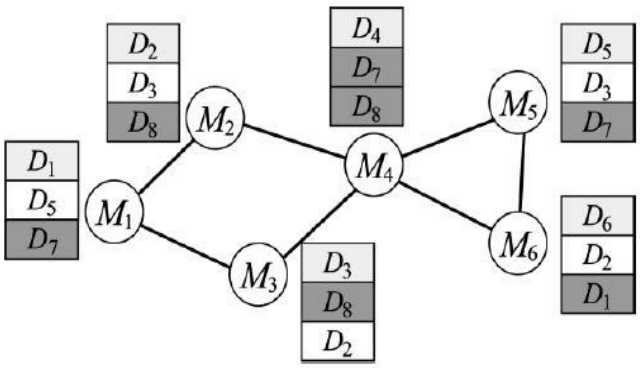

Fig 6: Relocated replicas based on DAFN method

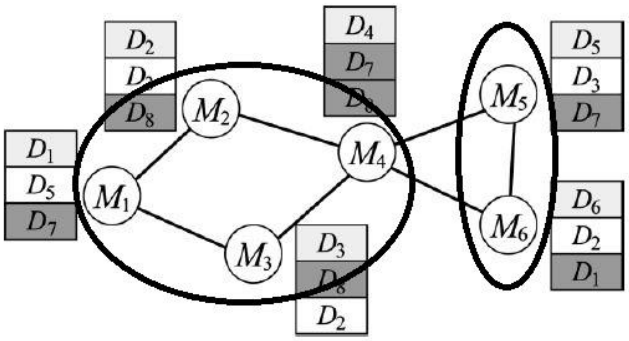

Fig 7: Formation of Groups[2]

The DCG method of replica allocation works as follows. First, it classifies the whole network into possible groups[2]. Then, allocate replicas until all the memory space within the same group become filled. It tries to avoid replica duplication within the group. But only at one circumstance, the case changes. It is when vacant memory space is identified. If there is vacant memory space, allocate replica even if it is a duplicate. This method provides high data stability but takes the longest time. The disadvantage is that, there is a chance of replica duplication again. In Fig 7, the network grouping is done effectively, and Fig 8 gives the output of replication in groups.

D. SCF-tree based replica allocation method

Replication can simultaneously improve data accessibility and reduce query delay; if the mobile nodes in a MANET together have sufficient memory space to hold both the original data and the replicas. However, if there is only limited memory space and many of the nodes hold the same replica, the overall data accessibility would be reduced. Thus, a node should not hold the same replica that is already held by other nodes. The issue on memory space can be easily solved like this. But, another important character of mobile nodes, called as selfishness will not allow the other nodes to follow this procedure. It can be called as *Selfish Replica Allocation*. It refers to a node's non-cooperative action, such that the node refuses to cooperate fully in sharing its memory space with other nodes. To solve this problem, we have to first detect and eliminate the selfish nodes and then perform replica allocation.

The SCF-Tree based replica allocation can be completed in 3 steps such as;

- Identifying selfish nodes
- Building SCF-Tree
- Allocating replica

1) Identifying Selfish Nodes

SCF-tree based replication requires the formation of an SCF-tree. It is formed without selfish nodes. So there is a need to eliminate selfish nodes. This section describes the effect of selfish nodes and the methods to eliminate them.

Table 3 : Table of access frequencies

Data	Mobile host						Group	
	M_1	M_2	M_3	M_4	M_5	M_6	G_1	G_2
D_1	0.65	0.25	0.17	0.22	0.31	0.24	1.29	0.55
D_2	0.44	0.62	0.41	0.40	0.42	0.46	1.87	0.88
D_3	0.35	0.44	0.50	0.25	0.45	0.37	1.54	0.82
D_4	0.31	0.15	0.10	0.60	0.09	0.10	1.16	0.19
D_5	0.51	0.41	0.43	0.38	0.71	0.20	1.73	0.91
D_6	0.08	0.07	0.05	0.15	0.20	0.62	0.35	0.82
D_7	0.38	0.32	0.37	0.33	0.40	0.32	1.40	0.72
D_8	0.22	0.33	0.21	0.23	0.24	0.17	0.99	0.41
D_9	0.18	0.16	0.19	0.17	0.24	0.21	0.70	0.45
D_{10}	0.09	0.08	0.06	0.11	0.12	0.09	0.34	0.21

During replica allocation, each node stores a copy of data present in some other nodes. But, some nodes will not allow other nodes to use their resources to store other's data. Such nodes can be called as selfish nodes. A node behaves as selfish only because of lack of energy and memory power. Some selfish nodes may not forward packets to conserve battery. Another category of selfish nodes may not receive packets to conserve their storage space. The second category of selfish nodes is responsible for adhoc networks not to allocate replica effectively. The presence of selfish nodes in a network will reduce the average throughput of the network. So there is a need to identify and eliminate the selfish nodes in every network.

The process of identification of selfish nodes can be done by using *Credit Risk* method. Credit Risk is defined as the measured risk of loss due to a node's non-cooperation in the network. If risk is more, the node is probably non-cooperative. Low risk means, the node can be cooperative. To estimate this, a threshold value is to be set, and calculate the credit risk by using an equation

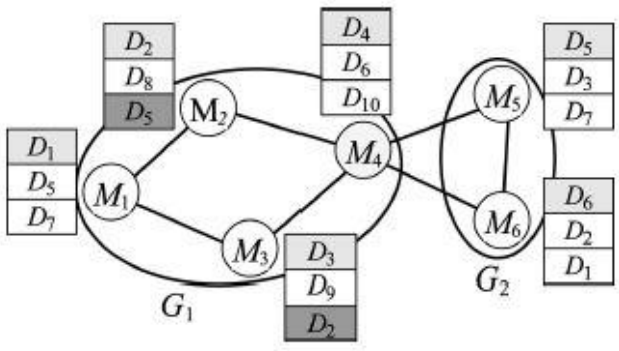

Fig 8: Relocated replicas based on DCG method

$$Credit\ Risk = \frac{Expected\ risk}{Expected\ value} \qquad (1)$$

If Credit risk > threshold → selfish node
If Credit risk < threshold → non-selfish node
The idea behind this concept is that, before sending the packets, the source node should identify whether the destination node is believable or not.

2) Building SCF-tree

SCF (Self-Centered Friendship) tree is formed with non-selfish nodes only. The nodes identified by the above method are eliminated, and the tree is formulated accordingly. The term, Self-Centered Friendship indicates that a person himself takes decision in selecting his friends. Applying this in MANETs means, a node maintains connection with other nodes in its own decision.

A node has to be taken as the source. Before building SCF-tree, each node makes its own topology graph having the network nodes and links as its elements. Then, the source node eliminates the selfish nodes using the above method, and redraws the graph. So, the topology graph will become a graph of non-selfish nodes only. Now, consider a node that wish to allocate replica, as the root node. Its SCF-tree connects the corresponding child nodes. Child nodes are the nodes which can be reached with a one-hop distance. The children of this generation will then be appended to them again. Suppose that if there is no child for a node taken as a parent, then it has to be considered as the child of the nearest parent node. According to the SCF-tree, a root node may have multiple routes to some nodes. Such nodes can be considered more stable. If such nodes exist, replicas will be allocated to them at first.

3) Allocating replica effectively

Every node has its own SCF-tree. The nodes within SCF-tree should maintain their priority based on *Breadth First Search* approach. According to this priority path, target node will be determined. Also, the memory space of each node is divided into Selfish memory (Ms) and public memory (Mp). Ms is assigned for storing its own data whereas Mp is for others. The replicas allocated by the other nodes are being stored in this part of memory space.

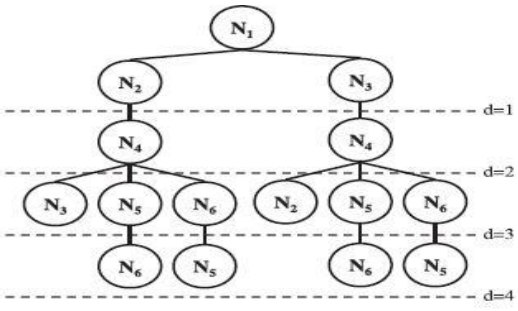

Fig 10 : SCF-tree of root node N1[2]

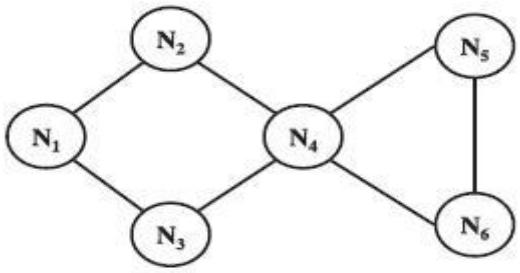

Fig 9 : A sample topology[8]

According to the SCF-tree shown in Fig 13, node N1 is the root node, and N1 is assumed to allocate its data replicas. First, it checks whether its own memory is having free space or not. If its Ms is not full, it will allocate replica to itself. Otherwise, the request will be forwarded to the first target node, N2. N2 will not donate its Ms for anyone else. The root node checks whether the Mp of the target is free or not. If its Mp is not full, N1 will allocate to N2. If it is full, it forwards the request to the next target node, N3. This target node is the root's expected node. If the target node provides its memory space to allocate replica, we say that the expected node have served the query. Any other node let the root node to allocate replica means, an unexpected node have served the query. In accordance with this expected node serving or not serving the query, the selfish features of the network nodes will be updated.

Suppose we are following an SCF-tree before eliminating selfish nodes, while the request is being forwarded, this process takes place. If the target node is non-selfish; it allocates its Mp for the source to store its replica. If it is selfish; the request will be forwarded to the next target node. In nutshell, each node asks other non-selfish nodes in SCF-tree to hold replica when the node cannot hold the replica in its own local memory space.

IV. PERFORMANCE EVALUATION

The above mentioned problem of improving data accessibility by allocating replicas has been performed in a simulation environment, and evaluated the performance at various constraints. The following section explains the analysis.

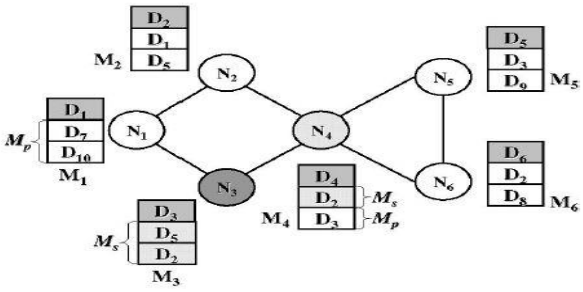

Fig 11 : Relocated replicas based on SCF-tree

A. *Simulation Environment*

The simulation model is similar to that employed in [7]. In the network taken into consideration, the number of nodes is set to 20. Each node has its selfish and public memory space for storing and relocating data packets. The network lies around a 1000*1000 topography. First we use the highly efficient *Credit Risk* method to filter out the selfish nodes among the network. Then, replicas will be allocated according to the access frequencies of each data item. The term *Data Accessibility* can be defined as the ratio of number of successful data requests to the total number of data requests sent. The four replica allocation methods are simulated in 150 seconds of simulation time, and evaluated their performance. The major performance metric considered was the access frequency itself.

B. *Simulation Results*

In accordance with the above mentioned parameters, the problem has been simulated. As a result, selfishness among an adhoc network has been identified. Along with that, the excellence of the above described replication methods have been compared.

1) *Detecting selfish nodes*

We completed the process of detection of selfish nodes among a MANET by Credit Risk method. The Credit Risk value of each node has been calculated first. The threshold was set to 0.7 with effect from a number of iterations performed. Finally the value of Credit Risk was compared with the already set threshold, and those having less credit risk were filtered as non-selfish nodes. Naturally, the other group of nodes was classified as selfish.

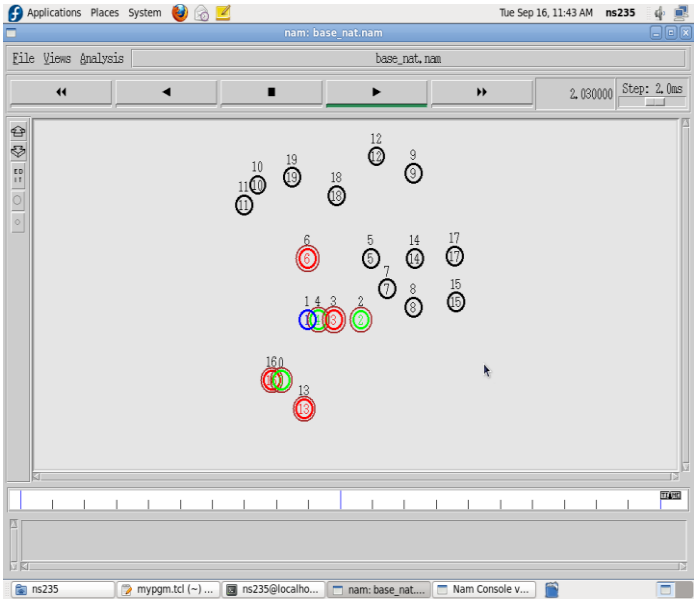

Fig 12 : NAM window showing selfish and non-selfish nodes

Table 4 : Simulation Parameters

No. of nodes	20
No. of data items	20
Routing Protocol	AODV
Topography	1000 * 1000
Simulation time	150 s
Packet size	1000
Channel Type	Wireless channel
Propagation Model	TwoRayGround
MAC type	MAC 802_15_4
Antenna model	OmniAntenna
Interface queue	Queue/DropTail/PriQueue
LLType	Link Layer

Fig 12 shows the output of step 1 of our work. A network of 20 nodes is being taken. We used some colours to distinguish the selfish and non-selfish nodes. Any one node has to be considered as source. Then find its neighboring nodes in accordance with their distance from the source node. Only the neighbors have to be considered for Credit Risk value calculation. The threshold of credit risk scores is set as 0.7. The credit risk value of each node is calculated and compared with this threshold value. Then, selfish and non-selfish nodes are identified. In the below simulation output, the node indicated with blue colour is the source node. Then, neighbors are given brown colour. Finally the selfish nodes are highlighted with red colouration and the remaining non-selfish nodes with green.

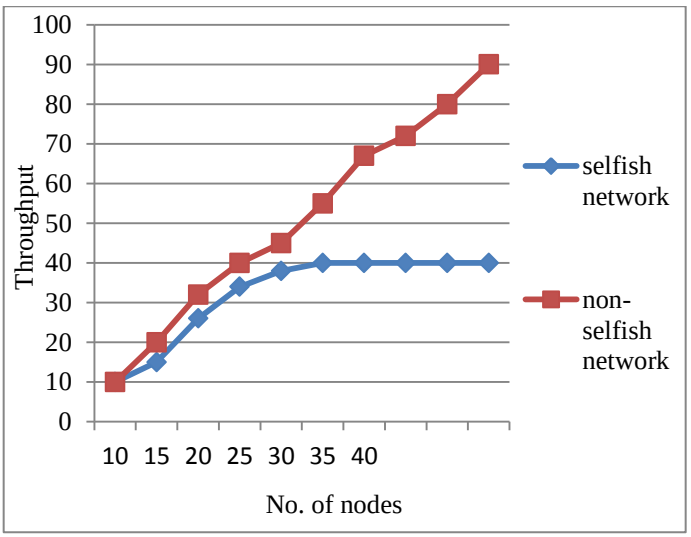

Fig 13 : Throughput variation between a selfish and non-selfish networks

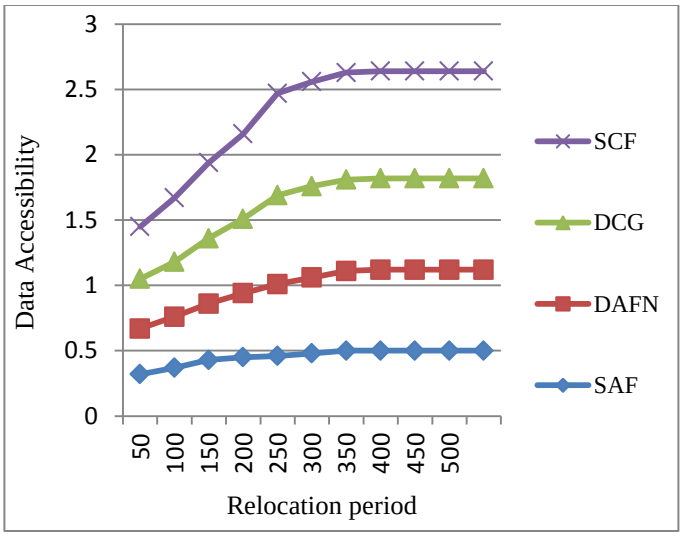

Fig 14 : Data accessibility in varying relocation period

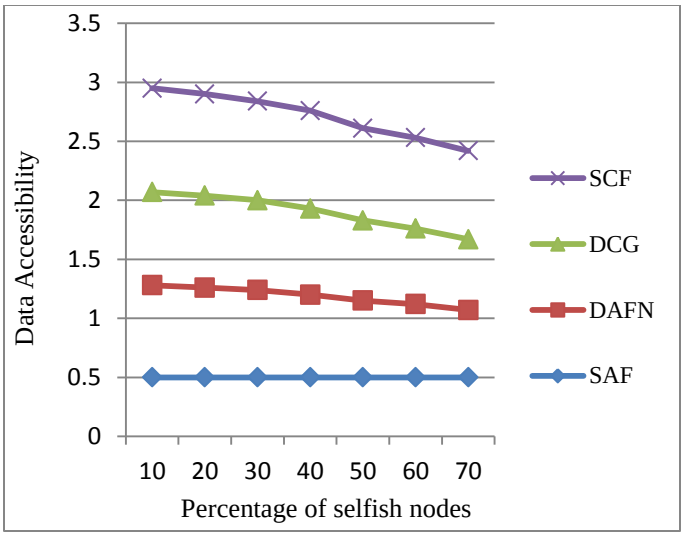

Fig 15 : Data accessibility in varying percentage of selfish nodes

The selfish nodes reduce the overall performance within a MANET. Fig 13 shows the graph for throughput variation between a selfish network and a non-selfish network. From the above description in section III, we know that selfish nodes are a kind of attacker nodes in MANETs. They create difficulties to the whole network. No node is always selfish; but when an ordinary node becomes selfish due to the lack of battery power or storage space, it reduces the overall network performance. So the number of packets delivered successfully at the destination will be low. Simulation result shows that the throughput obtained in a selfish network is much less than a non-selfish network. ie, more number of packets will be lost during transmission; if the network is affected by a group of selfish nodes.

2) Comparing replica allocation methods

We evaluated the variations in data accessibility with varying relocation periods and varying percentage of selfish nodes. The graphs shown in Fig 14 and 15, indicates this. During each relocation period, the data accessibility was measured according to the access frequencies. Here, we considered slightly variable values of access frequencies. The simulation result highlights the criterion behind these replication methods that data accessibility is often kept constant in maximum possible relocation periods. The converse is proved while calculating the data accessibility along with varying percentage of selfish nodes. As the number of selfish nodes in a network increases, the overall data accessibility of the network gets reduced. An excellent result to be noted is that, even though the data accessibility is being reduced, the SCF-tree based method is providing the highest data accessibility. The simulation results for the two above discussed conditions are pointed through Fig 14 and Fig 15. Finally, we came to a conclusion that the SCF-tree based replica allocation method outperforms SAF, DAFN as well as DCG methods.

The demerits of SAF method is washed out by the DAFN method. Similarly, the major drawbacks found in DAFN method are rectified by DCG method. But still the problem remained as unsolved; and finally the SCF method was able to solve this problem of replica duplication. It is because the SCF-tree based replica allocation method can identify and handle selfish nodes effectively and efficiently.

V. CONCLUSION

A network affected with the problem of selfishness will definitely reduce its lifetime. On one hand, we are replicating data packets to solve the problem of network division. At the same time, this replication is badly affected by a group of selfish nodes. As a result, the replication process changes to selfish replica allocation. This paper points out two things. From the simulation results, it is obtained that the overall throughput of a selfish network is always less than that of a non-selfish network. Also, various replica allocation techniques are discussed, and the performance of all those techniques are compared according to varying access frequencies. The concentration of selfish nodes also affects the network performance in some other way. The above analysis leads to a conclusion that the well-defined SCF based replica allocation method is showing maximum data accessibility in solving the problem of selfish replica allocation in mobile adhoc networks. We can extend this work to identify the nodes that are misinterpreted as selfish nodes; and to eliminate the actual selfishness in the network.

VI. ACKNOWLEDGEMENT

We would like to use this opportunity to express our gratitude to Mr. George M. Jacob, Lab Instructor, Toc H Institute of Science and Technology for his sincere support and guidance throughout this work.

REFERENCES

[1] S J K Jagadeesh Kumar, R. Saraswathi & R. Raja, "Improving the Performance of Mobile Ad Hoc Network using a Combined Credit Risk and Collaborative Watchdog Method", Global Journals Inc. (US)., Volume 13, Issue 6, Version 1.0, Year 2013

[2] Jae Ho Choi, Kyu Sun Shim, Sang Keun Lee, and Kun Lung Wu, "Handling Selfishness in Replica Allocation over a Mobile Ad Hoc Network", IEEE Transactions on mobile computing, Vol 11, no. 2,pp.278-291,February 2012

[3] Enrique Hern´andez-Orallo, Manuel D. Serrat, Juan-Carlos Cano, Carlos T. Calafate, Pietro Manzoni, "CoCoWa: A Collaborative Contact-based Watchdog for Detecting Selfish Nodes",IEEE Transactions on Mobile Computing, DOI 10.1109/TMC.2014.2343627, 2014

[4] L Anderegg and S Eidenbenz, "Ad Hoc-VCG: A Truthful and Cost-Efficient Routing Protocol for Mobile Ad Hoc Networks with Selfish Agents", Proc. ACM MobiCom, pp.245-259, 2003

[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] Takahiro Hara, Sanjay K. Madria, "Data Replication for Improving Data Accessibility in Ad Hoc Networks", IEEE transactions on mobile computing, Vol. 5, no. 11, November 2006

[7] T Hara, "Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility," Proc. IEEE INFOCOM, pp. 1568-1576, 2001

[8] 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

[9] L Yin and G Cao, "Balancing the Tradeoffs between Data Accessibility and Query Delay in Ad Hoc Networks", Proc. IEEE Int'l Symp. Reliable Distributed Systems, pp. 289-298, 2004

[10] Josephin Jeneba Y, Prabakaran T, "Detection of Selfish Node in Manet using a Collaborative Watchdog", international journal of engineering sciences & research Technology, ISSN: 2277-9655, May 2013

[11] Sheethal Sunny1, Dr.C.D.Suriyakala, "Performance Analysis of Selfish Nodes in Mobile Ad-hoc Networks", ISSN (Print) : 2320 – 3765, ISSN (Online): 2278 – 8875, Vol. 3, Issue 5, May 2014

[12] 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

[13] Yanchao Zhang, Wenjing Lou, Yuguang Fang, "SIP: A Secure Incentive Protocol against Selfishness in Mobile Ad Hoc Networks", IEEE Communications Society, 2004

[14] Kejun Liu, Jing Deng, Pramod K. Varshney, Kashyap Balakrishnan, "An Acknowledgment-Based Approach for the Detection of Routing Misbehavior in manets", IEEE transactions on mobile computing, Vol. 6, no. 5, May 2007

[15] K.Sridevi, S.Kannan, S.Karthik, "A Survey on Selfish Node Detection Using Several Techniques in MANET", International Journal of Inventive Engineering and Sciences (IJIES) ISSN: 2319–9598, Volume-I, Issue-I, December 2012

[16] J.Vijithanand, K.Sreerama Murthy, "A Survey on Finding Selfish Nodes in Mobile Ad Hoc Networks", J.Vijithanand et al,/ (IJCSIT) International Journal of Computer Science and Information Technologies, ISSN:0975-9646,Vol. 3 (6), 2012,5454-5461, 2012

[17] S.-Y. Wu and Y.-T. Chang, "A User-Centered Approach to Active Replica Management in Mobile Environments," IEEE Trans.Mobile Computing, Vol. 5, no. 11, pp. 1606-161, Nov. 2006.

[18] Sagar D. Padiya, Rakesh Pandit, Sachin Patel, "A System for MANET to Detect Selfish Nodes Using NS2", International Journal of Engineering Science and Innovative Technology (IJESIT), Volume 1, Issue 2, November 2012

[19] Mr. Ritesh Dhanare, Dr. Sunita Varma, "Replica Allocation in MANETs for Eliminating Selfish Node", International Journal of Scientific & Engineering Research, ISSN 2229-5518,Volume 4, Issue 8, August 2013

[20] Y-Yoo and D P Agrawal,"Why does it pay to be selfish in a manet?", IEEE Wireless Communications, December 2006

[21] N.Muthumalathi, Dr.M.Mohamed Raseen, "Fully Selfish Node Detection, Deletion And Secure Replica Allocation Over Manet", International Conference on Current Trends in Engineering and T echnology, 2013

[22] Gayathry S S, R N Gaur, "Handling Selfishness in MANETs – A Survey", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE), ISSN (print) : 2320–3765, ISSN (online) : 2278 – 8875, vol. 3, issue 11, pp 13193-13200, November 2014