ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

A Novel Replica Allocation Approach Depth-First Iterative-Deepening for Handling Selfishness in MANETs

D. Francis Xavier Christopher[1], C. Amirtham[2]

Director, School of Computer Studies, RVS College of Arts and Science, Coimbatore, India[1]

Research Scholar, Department of Computer Science, RVS college of Arts and Science, Coimbatore, India[2]

Abstract:: In mobile unexpected networks (MANET), nodes move freely and therefore the distribution of access requests changes dynamically. Duplicate allocation in such a dynamic atmosphere may be a important challenge. The communication price has become an outstanding issue influencing the performance of duplicate allocation within the MANET environment. In reality, however, some nodes might egotistically decide solely to work part, or not in the least, with alternative nodes. These selfish nodes may then scale back the knowledge accessibility within the network. In this paper, the impact of selfish nodes in a very mobile unexpected network from the angle of duplicate allocation is examined. This can be termed as selfish duplicate allocation. Especially, a selfish node detection formula is developed during this research work that considers partial selfishness and novel replica allocation technique to properly deal with selfish replica allocation. Depth-First Iterative-Deepening formula is used during this in this work for higher performance.

Keywords: MANET, Replica Allocation, Mobility, Selfish node detection, Depth-First Iterative-Deepening

I. INTRODUCTION

A mobile ad-hoc network (MANET) is that the assortment of mobile nodes that are equipped by many 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 do not 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.

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.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

- 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.
-

A selfish node 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 de-lay. 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 de-creased. So to increase data accessibility, a node should not hold the replica that is also held by many other nodes. How-ever, 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 re-source only for its own benefit i.e., it acts selfishly [7].

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 [16].

II. REVIEW OF LITERATURE

i) Effective Replica Allocation in MANET

There are three techniques planned within the effective replica allocation. That are,

- Static Access Frequency
- Dynamic Access Frequency and Neighbourhood
- Dynamic connectivity based Grouping,

These techniques create the subsequent assumptions:

- Each information item and every mobile host is appointed as distinctive symbol
- Every mobile host has finite memory space to store replicas
- There are not any update transactions

The access frequency of every information item, a particular is the range of times a specific mobile host accesses that information item during a unit interval, is understood and doesn't modification.

the choice of that information things square measure to be replicated on that mobile hosts relies on information items' access frequencies and such a call is taken throughout a particular amount of your time, known as the relocation amount [7], [16].

ii) Static Access Frequency

In this SAF method [7] [16], the nodes replica is based on the access frequencies of the data items. The following fig.2.1 shows the Static Access Frequency method. In SAF method, the mobile nodes with the same access frequencies to data items allocate the same replica. 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 the nodes.

iii) The DAFN method (Dynamic Access Frequency and Neighbourhood)

To overcome the problem of replica duplication in the SAF method, a new method of replica allocation called DAFN method [7] was developed. It eliminates the replica duplication among neighbouring mobile hosts.
The algorithm of DAFN [16] method is as follows:

- Each mobile host broadcasts its host id and access frequency information at relocation period.
- Each mobile node allocates the replica according to SAF method.
- If two mobile nodes have the same data item then the node having replica changes it to another replica which having high access frequency.

.

iv) Dynamic Connectivity based Grouping (DCG)

The DCG method shares replicas in each larger group of mobile hosts than the DAFN. The DAFN method shares replicas among neighbouring nodes. But the DCG method [13] [7], shares replica data items in many groups of mobile nodes than DAFN. The Dynamic Connectivity grouping method creates groups of mobile

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

nodes that are bi-connected components in an ad hoc network.

v) Dynamic connectivity grouping with detection (DCG+)

The technique combines DCG with detection method. Initially, groups of nodes are created according to the DCG methodology. Subsequently, in each group, selfish nodes are detected based on our detection method. For the detection, each node in a group sends its nCR scores to the coordinator with the lowest suffix of node identifier in the group [14]. The coordinator excludes selfish node(s) from the group for replica allocation. As a result, only non-selfish nodes form a group again. The replica allocation is only performed within the final group without any selfish nodes.

After replica allocation, the coordinator shares the information of replica allocation with group members for the subsequent selfishness detection. In particular, selfish nodes are determined to be selfish only when all other nodes in the group agree with the node's selfishness.

The DCG method shares replica of data items in many groups of mobile nodes than the DAFN method that shares replicas among neighbouring nodes. The DCG method creates groups of mobile nodes that are bi-connected components in an ad hoc network. In spite of grouping mobile nodes as a bi-connected component, the group is not divided even if one mobile node is disconnected from the network.

The algorithm of DCG method is as follows:

- Each mobile node broadcast its host id and information about its access frequency with data items to other nodes.
- By using the broadcasting information every node identifies the bi-connected component nodes.
- In each group, an access frequency of the group to each data item is calculated by adding all the access frequencies of mobile node in that group.
- According to the access frequencies of the group, replicas of data items are allocated until memory of all mobile nodes in the group becomes full.
- After allocating replicas of all data items, if the mobile nodes have any free space then replicas are allocated according to their access frequencies until the memory space is full. It causes high traffic due to exchange of information but it provides high data accessibility and stability over nodes.

In MANETs some of the nodes do not take part in forwarding packets to other nodes to conserve their resources such as energy, bandwidth and power. The nodes which act selfishly to conserve their resources are called selfish nodes.

III. PROPOSED STRATEGY

Proposed strategy consists of three parts:

i) Detecting selfish nodes
ii) Constructing SCF-tree
iii) Allocating replica using DFID

At a specific period, or relocation period [6], each node executes the following procedures: Each node detects the selfish nodes based on credit risk. Each node makes its own topology graph and constructs its own SCF-tree by excluding selfish nodes. Based on SCF-tree, each node allocates replica in a fully distributed manner. The CR score is updated accordingly during the query processing phase. The credit risk is used to effectively measure the "degree of selfishness."

A node wants to know if another node is believable, in the sense that a replica can be paid back, or served upon re-quest to share a memory space in a MANET. To measure degree of selfishness, the novel tree that represents relationships among nodes in a MANET, for replica allocation, termed the SCF-tree. The key strength of the SCF-tree-based replica allocation techniques is that it can minimize the communication cost, while achieving high data accessibility. This is because each node detects selfishness and makes replica allocation at its own discretion, without forming any group or engaging in lengthy negotiations.

i) Detecting Selfish Node

The credit risk is represented by the subsequent equation:

$$\text{Credit Risk} = \frac{ExpectedRisk}{Expectedvalue}$$

Each node calculates a CR score for each of the nodes to which it is connected. Each node shall estimate the "degree of selfishness" for all of its connected nodes based on the score. It describes selfish features that may lead to the

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

selfish replica allocation problem to determine both expected value and expected risk.

$$CR_i^k = \frac{P_i^k}{\alpha * SS_i^k + (1-\alpha)|*ND_i^k}, where 0 \le \alpha \le 1 \quad [7]$$

Where, CR= Credit Risk

α -the system parameter

SS_i^k - Size of N_k 's shared memory space

ND_i^k -The number of N_k 's shared data items

The system parameter is used to adjust the relative importance of and Node updates at every query processing and looks it up for the connected node at every relocation period. In addition, each node also has its own threshold of If the measured CR_i^k exceeds to node N_k will be detected as a selfish node by N_i .

The value of P_i^k (as well as SS_i^k and ND_i^k) is updated at every query processing of some item that N_i allocates to other node(s) during the replica allocation phase. The effect of parameters SS_i^k and ND_i^k on CR_i^k can be weighted by taking into consideration the size of memory space at node N_i , S_i , and the total number of data items accessed by N_i , n_i. The rationale is that CR_i^k may be strongly affected by S_i , n_i and if CR_i^k is not normalized. By normalizing, we obtain (3), where nCR_i^k stands for the normalized CR_i^k

$$nCR_i^k = \frac{P_i^k}{\alpha * \frac{SS_i^k}{S_i} + (1-\alpha) * \frac{ND_i^k}{n_i}}, Where 0 \le \alpha \le 1$$

ii) Constructing SCF tree

The SCF-tree based replica allocation techniques are extol by human friendship management in the real world, where each one makes his/her own friends create a web and negotiate, friendship by himself/herself. He/she does not have to address these with others to maintain the friendship. The main objective of the novel replica allocation technique is to reduce traffic overhead, while reach high data accessibility. The novel replica allocation techniques can allocate replica without discussion with other nodes, as in a human friendship management, traffic overhead will decrease. .

iii) Allocating Replica using DFID

After constructing the SCF-tree, a node allocates replica at every relocation period. Every node asks non-selfish nodes within its SCF-tree to hold replica when it can-not hold replica in its local memory space. The SCF-tree based replica allocation is execute in a fully spread out manner, each node determines replica allocation individual basis without any communication with other nodes. Since every node has its possess SCF-tree, it can perform replica allocation at its carefulness.

a) Depth-First Iterative-Deepening

A search algorithm which suffers neither the draw-backs of breadth-first nor depth-first search on trees is depth-first iterative-deepening (DFID). The algorithm first, perform a depth-first search to depth one. Then, dumping the nodes generated in the first search, start over and do a depth-first search to level two. Next, start over again and do a depth-first search to depth three, etc., enduring this process until a goal state is reached. DFID expands all nodes at a given depth before expanding any nodes at a greater depth; it is definite to find a shortest-length solution. Also, since at any given time it is performing a depth-first search, and never searches deeper than depth d, the space it uses is O (d).

The spontaneous reason is that almost all the work is done at the deepest level of the search. Regrettably, DFID suffers the same drawback as depth-first search on arbitrary graphs, namely that it must explore all possible paths to a given depth [13].

The nodes at depth d are generated once during the final iteration of the search. The nodes at depth d - 1 are generated twice, once during the final iteration at depth d, and once during the intermediate iteration at depth d - 1. Similarly, the nodes at depth d - 2 are generated three times, during iterations at depths d, d - 1, and d - 2, etc. Thus the total number of nodes generated in a depth-first iterative deepening search to depth d is

$$b^d + 2b^{d-1} 3b^{d-2} + \cdots + db.$$

Factoring out b^d gives

$$b^d(1 + 2b^{-1} + 3b^{-2} + \cdots + db^{1-d}).$$

Letting x = llb yields

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

$$b^d(1+2x^1+3x^2+\cdots+dx^{d-1}).$$

This is less than the infinite series

$$b^d(1+2x^1+3x^2+4x^3+\cdots),$$

This converges to

$$b^d(1-x)^{-2}\ for\ abs(x)<1.$$

Since (1 - x)-~, or (1 - l/b), is a constant that is independent of d, if b > 1 then the running time of depth-first iterative-deepening is O(bd). To see that this is optimal, we present a simple challenger argument. The number of nodes at depth d is bd. Imagine that there exists an algorithm that examines less than bd nodes. Then, there must be present at least one node at depth d which is not examined by this algorithm. Since there is no additional information and challenger could place the only solution at this node and hence the proposed algorithm would fail. Since DFID at any point is engaged in a depth-first search, it requires storing a stack of nodes which represents the branch of the tree it is expanding. Since it finds a solution of most favourable length, the maximum depth of this stack is d, and hence the maximum amount of space is O (d) [13].

IV. RESULTS AND DISCUSSION

This research work is simulated in java. The performance of the proposed approach is evaluated using certain parametric standards.

i. Overall Selfishness Alarm
ii. Communication cost
iii. Average Query Delay
iv. Data Accessibility
v. Varying size of memory space.

i) Overall Selfishness Alarm

Fig.1 presents the overall selfishness alarm with varying relocation period and the size of memory space, respectively. As expected, the DCG+ technique significantly reduces the selfishness alarms in the entire cases. This can be explained as follows: The small number of selfish nodes becomes expected nodes in DCG+ than in DCG, since our detection method augmented in DCG+ detects selfish nodes effectively and the detected selfish nodes are removed from replica allocation groups. Accordingly, more expected nodes serve queries in DCG+ than in DCG.

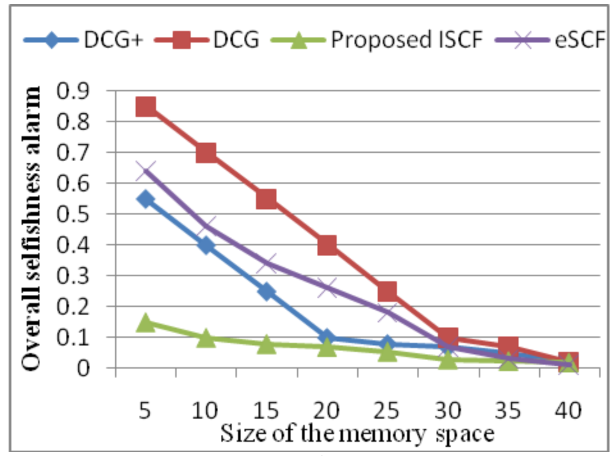

Fig1: overall selfishness alarm

As expected, the overall selfishness alarm of DCG (selfishness only) and DCG+ (selfishness only) is less than that of DCG. We see that, on average, about 62 and 56 per-cent of the overall selfishness alarm with DCG and DCG+ are caused by node selfishness, not disconnections, in Fig1. Clearly, Fig. 1 shows that the detection method can reduce the overall selfishness alarm effectively.

ii) Communication cost

As shown in Fig. 2, communication cost increases as local memory size at first, but it decreases from a assured memory size in every technique, except Static Access Frequency. When the memory size is larger than a assured memory size, each node holds replicas of many data items and thus replica relocation rarely occurs.

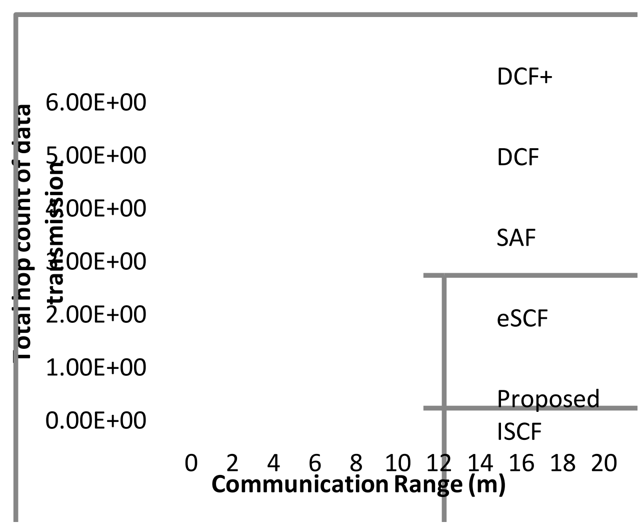

Fig2: Communication cost

Equally, when the communication range is larger than a certain point, the number of hops in the middle of

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 9, September 2013

connected nodes decreases. Then, the communication cost caused by replica repositioning decreases. Compare to existing method ISCF reduce the communication cost.

iii) Average query delay

Fig.3 shows average query delay for various parameters. As expected, the proposed ISCF technique shows the best performance in terms of query delay, since most winning queries are served by local memory space. Since most winning queries are served by group members in these techniques, the long distance between group members affects query delay unconstructively.

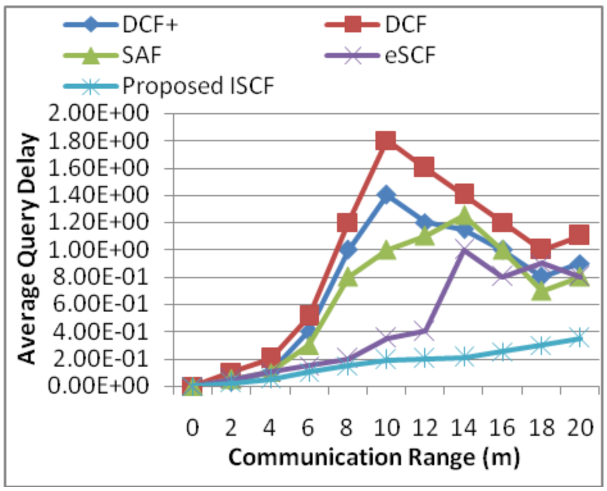

Fig 3 : Average query delays

The above fig.3 shows that the average query delay of all techniques degrades as the communication range increases, but it improves from a certain point, since when the communication range is larger than 9, the number of hops between connected nodes decreases.

iv) Data Accessibility

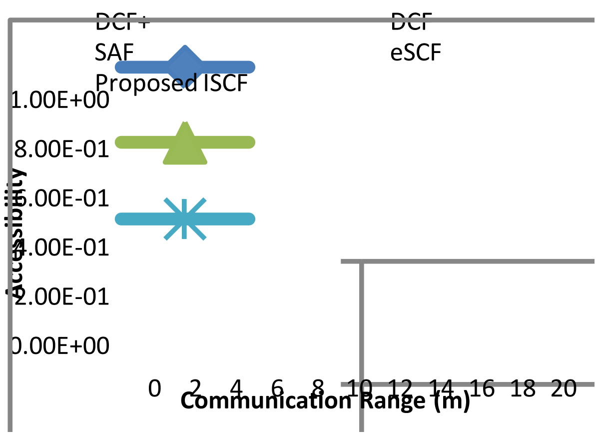

Fig 4: Data Accessibility

Fig.4 shows that the data accessibility improves with the wide range of communication, since additional nodes become connected.

v) Varying size of memory space

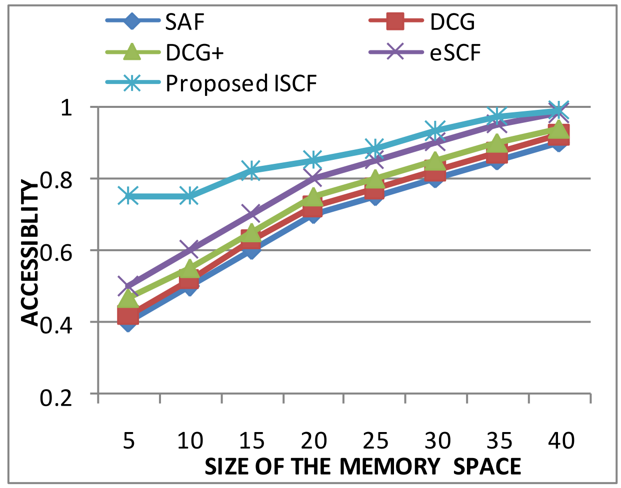

Fig5: Varying size of memory space.

The performance of DFID techniques improves faster than do others, since the ISCF technique fully utilize the memory space of nodes. Fig.5 shows the robustness of ISCF techniques with respect to varying percentage of selfish nodes. The profit of DCG technique is significantly hampered by selfish nodes, whereas the Static Access Frequency technique is insensitive at all.

IV. Comparative Study

Table 1: Performance Comparisons of Existing and Proposed method

Parameters	SAF	DCG	DCG+	eSCF	ISCF
Overall selfishness alarm	0.9	0.85	0.7	0.6	0.15
Varying size of memory space	0.38	0.4	0.45	0.5	0.7
Accessibility	0.8	0.9	1.00	0.9	1.00
Average query delay	8.00	1.10	9.00	8.00	3.50
Communication cost	7.50	1.60	1.90	7.50	4.50

Table1 shows the performance comparison of Existing and Proposed method. ISCF is the proposed method. Compared the SAF, DCG, DCG+, eSCF with ISCF, the ISCF gives best performance of accessibility, and reduce the average query delay.

V. CONCLUSION

This research work was motivated by the fact that a selfish replica allocation could lead to overall poor data accessibility in a MANET. An efficient selfish node detection method and novel replica allocation technique is proposed to handle the selfish replica allocation suitably. The proposed strategies are inspired by the real-world observations in economics in terms of credit risk and in human friendship management in terms of choosing one's friends completely at one's own discretion. The concept of credit risk from economics to detect selfish nodes has been considered. Every node in a MANET calculates credit risk information on other connected nodes individually to measure the degree of selfishness. Since conventional replica allocation techniques failed to consider selfish nodes, a novel replica allocation technique has been presented in this paper. Extensive simulation shows that the proposed strategies outperform existing delegates' cooperative replica allocation techniques in terms of data accessibility, communication cost, average query delay, varying size of memory space and overall selfishness alarm.

REFERENCES

[1] Abhishek Gupta , Amit Saxena " Detection and Prevention of Selfish Node in MANET using Innovative Brain Mapping Function: Theoretical Model" International Journal of Computer Applications (0975 – 8887) Volume 57– No.12, November 2011

[2] Baumeister and M.R. Leary, "The Need to Belong: Desire for Interpersonal Attachments as a Fundamental Human Motivation," Psychological Bull., vol. 117, no. 3, pp. 497-529, 1995.

[3] 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

[4] G. Tamilarasi, Devi Selvam, "Allocation of Replicas by Solving Selfishness in MANET", IJERT, Vol. 2 Issue 1, January- 2013.

[5] Hugo Miranda Lu´ıs Rodrigues "Friends and Foes: Preventing Selfishness in Open Mobile Ad Hoc Networks" Proceedings of the 23 rd International Conference on Distributed Computing Systems Workshops (ICDCSW'03)0-7695-1921-0/03 $17.00 © 2003 IEEE International Journal Of Computer Science And Applications Vol. 6, No.2, Apr 2013 ISSN: 0974-1011

[6] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Net work Routing Protocols," Proc. ACM MobiCom, pp. 85-97, 1998.

[7] 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, 2012.

[8] Jim Solomon Raja.D1, Immanuel John Raja.J2" A Survey on Selfishness Handling In Mobile Ad Hoc Network" International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 11, November 2012)

[9] JiunLong Huang, MingSyan Chen and WenChih Peng, "Exploring Group Mobility for Replica Data Allocation in a Mobile Environment", CIKM'03, ACM 1581137230, 2003

[10] 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

[11] K.Indumathi, 2N.Jayalakshmi, 3S.Kartiga " Selfish node detection using replica allocation techniques and SCF -tree in MANET" International journal for advance research in engineering and technology, Vol. 1, Issue II, Mar. 2013 ISSN 2320-6802

[12] K.P.Shanmuga Priya and V.Seethalakshm, "Replica Allocation In Mobile Adhoc Network For Improving Data Accessibility Using SCF-Tree" IJMER, Vol.3, Issue.2, pp-915-919, 2013.

[13] Richard E. Korf, " Depth-First Iterative-Deepening : An Optimal Admissible Tree Search" , Artificial Intelligcnce 27 (1985) 97-109 0004-3702/85/$3.30 @ 1985, Elsevier Science Publishers B.V. (North-Holland).

[14] S.Bhuvaneshwari1, Prof.M.Suguna "Identifying and handling of false alarm in selfish replica allocation" International Journal of Computer Science and Management Research Vol 2 Issue 4 April 2013 ISSN 2278-733X

[15] 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, ISSN: 2319-5967 ISO 9001:2008 Certified

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

[17] Zheng Jing, Lu Xieng, Yang Kan, Wang Yije, "A Dynamic Adaptive Replica Allocation Algorithm in Mobile Ad Hoc Networks" Computing and Informatics, Vol. 23, Pp. 133–156, 2004.

BIOGRAPHIES

D. FRANCIS XAVIER CHRISTOPHER received his B.Sc., in 1996, M.Sc., in 1998 from Bharathiar University, Coimbatore. He obtained his M.Phil, in the area of Networking from Bharathiar University, Coimbatore in 2002. He submitted PhD thesis in 2013. At present he is working as a Director, School of Computer Studies, Coimbatore. His research interest lies in the area of Software Engineering.

C. Amirtham received her BCA, in 2009, M.Sc., in 2011 from Bharathiar University, Coimbatore. B.Ed in 2012 from Tamilnadu teacher's' education University,Chennai. At present she is pursuing her M.Phil in the area of Networking in RVS College of Arts and Science, Coimbatore.