ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2014

Performance Analysis of Selfish Nodes in Mobile Ad-hoc Networks

Sheethal Sunny[1], Dr.C.D.Suriyakala[2]

PG Student [Wireless Technology], Dept. of ECE, Toc H Institute of Science and Technology, Kochi, India [1]

Professor & Head, Dept. of ECE, Toc H Institute of Science and Technology, Kochi, India [2]

ABSTRACT: Mobile Ad-hoc Networks is a collection of wireless network that can dynamically form a network to exchange information. Dynamic topology of MANETs may result in network partition. Network partition of mobile nodes in one network may not be able to access data hosted by nodes in other network. Due to the node's non cooperative action, such node's refuses to cooperate fully in sharing its memory space with other nodes. But some of the nodes may decide not to co-operate with other nodes. Such behavior of these nodes considered as selfish may degrade the performance the network or in data accessibility. MANETs performance is mostly dependent on collaboration of all nodes, detection of selfish node is essential task in MANETs in order to improve the performance. This paper highlights novel methods to detect selfish nodes through selfish node detection algorithm that considers partial selfishness and replica allocation techniques to cope with selfish replica allocation. The proposed approach outperforms cooperative replica allocation in terms of improves the data accessibility, reduces the communication cost.

KEYWORDS: Communication Cost, Data accessibility, Mobile Ad-hoc Networks, Selfish replica allocation.

I.INTRODUCTION

MANET (Mobile Ad-hoc Network) can be described as an autonomous collection of wireless network that communicate over low capacity wireless links, without a centralized infrastructure [3]. In mobile computing environments, by utilizing wireless networks, users equipped with portable computers called mobile hosts, can change their locations while retaining network connections [17].Mostly two different approaches for enabling wireless mobile units to communicate with each other:1) Infrastructure: - Mobile networks have been on the cellular concept and depend on good infrastructure, in which mobile devices communicate with access points like base station connected to the stable network infrastructure. 2) Infrastructure less network: In this approach there is no central administration for the entire network. Wireless nodes that can be dynamically form a network to exchange information without any pre-existing stationary network infrastructure. In this network, as all the nodes are having mobility, they move freely. This mobility causes network partitions hence data accessibility in ad hoc networks is lower than the fixed networks. In these networks, dynamic topology leads to network partition and also it degrades the performance of networks. Nodes which are not willing to forward packets and share their memory space are called selfish nodes. Node's non co-operative action, such that the node refuses to cooperate fully in sharing its memory space with other nodes is called selfish replica allocation [5].

The characteristics of selfish nodes:

- Dropping of data packet.
- Intentionally delay the RREQ packet.
- Do not reply or send hello messages.
- Do not take part in routing process.

Many studies were conducted on ad-hoc network and selfish node detection. In this paper we are describing existing methods and suggested solutions. In the proposed work we use the credit risk method to detect selfish node. The suggested work is also taking care of avoiding the false alarm of detecting selfish nodes. Network simulator (ns-2) is used for the comparative study of MANETs different mobility pattern using suitable metric like data accessibility, communication cost, detection time and query delay.

II.RELATED WORKS

From the literature survey highlights MANETs performance, that degrade multi-hop communication requires collaboration among nodes, which forward packets [15] for one another. Most studies of ad hoc networks assume that nodes can be programmed to always perform this forwarding functionality. In commercial deployment of MANETs, some nodes may refuse to forward packets in order to conserve their limited resources (for example, energy), resulting in traffic disruption [17]. Nodes exhibiting such behaviour are termed selfish. Selfishness is usually passive behaviour. Additionally, malicious nodes may intentionally, and without concern about their own resources [10], attempt to disrupt network operations by mounting denial-of-service attacks or by actively degrading the network performance. Selfish and malicious behaviours are usually distinguished based on the node’s intent. Network disruption is a side effect of the behaviour of a selfish node, while disrupting the network is the intent of malicious nodes.

Network partitions can occur frequently, since nodes move freely in a MANETs, causing some data to be often inaccessible to some of the nodes. As a result, data accessibility is often an important performance metric in a MANETs [5].Mostly data are usually replicated at nodes, other than the original owners, to increase data accessibility to cope with frequent network partitions. Data replication can simultaneously improve data accessibility and reduce query delay in MANETs , the nodes have sufficient memory space to hold both all the replicas and the original data. A node may act selfishly by using its limited resource only for its own benefit, since each node in a MANETs has resource constraints, such as battery and storage limitations [14]. A node would like to enjoy the benefits provided by the resources of other nodes, but it may not make its own resource available to help others. Such selfish behaviour can potentially lead to a wide range of problems for a MANETs. To mitigate this problem, survey suggest different approach like watch dog, path rather method [14], TWO ACK scheme and S-TWO ACK schemes [7], reputation scheme ,game theory based scheme and credit payment based scheme [20].

III.SELFISH NODE DETECTION

By analysing the above mention methods with respect to achieved results in survey, we have proposed a selfish node detection method and novel replica allocation techniques to handle the selfish replica allocation appropriately. To solve such problems we examine the impact of selfish nodes in a MANETs from the perspective of replica allocation [5].We term this selfish replica allocation. In particular, we develop a selfish node detection algorithm that considers partial selfishness and novel replica allocation techniques to properly cope with selfish replica allocation. For that every node in MANETs calculates credit risk information of other connected nodes to measure the degree of selfishness. The proposed strategies are based 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. We applied the notion of credit risk from economics to detect selfish nodes. Every node in a MANETs calculates credit risk information on other connected nodes individually to measure the degree of selfishness. We also proposed another method, collaborative watch dog (CWD) for detecting selfish nodes. In collaborative watch dog (CWD) method [15], detection time of selfish nodes has to be reduced based on contact dissemination. In collaborative watch dog (CWD) method, if one node has previously detected as a selfish node using this method, that information can be spread to other nodes when a contact occurs. And finally we propose a set of replica allocation techniques that use the self-centered friendship tree (SCF) and also collaborative watch dog (CWD) method to reduce communication cost, query delay while achieving good data accessibility. The simulations can be carried out using an object oriented network simulating tool called ns-2.

Credit Risk Method

The network is model as a set of N wireless mobile nodes with C collaborative nodes and S selfish nodes (N = C +S). The credit risk for the each node can be described by the following equation:

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

From the equation (1), the credit score (CR) for each node is calculated. Based on the CR score, estimate the “degree of selfishness” for all of its connected nodes. The Selfish node features can be divided into two categories: node specific and query processing-specific features. The Node specific features can be used to represent the number of shared items & shared memory space used for that node. It is used to represent the expected value of a node. If the node N_i requests the data to the node N_k means, the node N_k share the memory space and the data items for that node N_i. So the node N_k is treated as a valuable node. Then the query processing feature is calculated for the node N_i. It is defined as the ratio of

N_i's data request being not served by the expected node N_k. Because the node N_k is selfish node and it does not share its own memory space. This feature is used to measure the expected risk of a node. The probability of the expected risk of the node pick is larger means, the node N_i will be treated as the risky for the node the node neck cannot serve N_i's requests due to selfishness in its memory usage. The value of the crack is the credit risk of node N_i. Each node has its own threshold value δ . α is the system parameter, where 0≤ α ≤1. The formula for finding the credit risk is

$$n\,CR_i^k = \frac{P_i^k}{\alpha * \frac{SS_i^k}{S_i} + (1-\alpha) * \frac{ND_i^k}{n_i}} \text{ , where } 0 \le \alpha \le 1 \qquad (2)$$

In the equation (2) where,

1. SS_i^k is the size of N_k's shared memory space.
2. ND_i^k is the number of N_k's shared data items.
3. P_i^k is the ratio of N_i's data request being not served by the expected node N_k.
4. CR_i^k is the credit risk of node N_i.
5. δ is the threshold value of node N_i.
6. α is the system parameter ,where 0≤α ≤1.

Steps for Detecting the Selfish Node

1. Find the credit risk for each node in the network by using the equation No (2).
2. Based on the credit risk of each node, we can set the threshold value for each ode.
3. If the credit risk value is less than the threshold value, set the node is non selfish node. Otherwise it is selfish node.
4. Find the behavior of the node whether it is partial selfish or fully selfish.
5. For each connected node in Nk , we allocate the number of replica and the total size of the allocated replica.
6. Find the query processing time for each requested node in the network.
7. Determine the expected node responds to the requested node or unexpected node responds to the requested node.

Watch dog Method

In this method, detection time of selfish node have to be reduced based on contact dissemination. If one node has previously detected as a selfish node using its watchdog method, that information can be spread to other nodes when a contact occurs. So that if a node have positive value, if it knows the selfish node. To model this fact, introduce a probability of detection (pd).This probability depends on the effectiveness of the watchdog and the type of contact. The network is modelled as a set of N wireless mobile nodes, with C collaborative nodes and S selfish nodes (N = C +S). It is assumed that the occurrence of contacts between two nodes follows a Poisson distribution λ. In this case, a collaborative node has 2 states: NOINFO [15], when the node has no information about the selfish node, and POSITIVE when the node knows who the selfish node is (it has a positive).All nodes have an initial state of NOINFO and they can change their initial state when a contact occurs. Using a contact rate λ we can model the network using a Continuous Time Markov Chain (CTMC) with states si =(c), where c represents the number of collaborative nodes in the POSITIVE state. At the beginning, all nodes are in NOINFO state [15]. Then, when a contact occurs, c can increase by one.

Assume both nodes are collaborative. 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. We model this with the probability of collaboration (pc). The degree of collaboration is a global parameter of the network to be evaluated. This value is used to reject that either a message with the information about the selfish nodes is lost or that a node temporally does not collaborate.

Building SCF- Tree

The Self Centered Friendship tree based replica allocation techniques are inspired by human friendship management in the real world, where each person makes his/her [5] own friends forming a web and manages friendship by himself/herself. He/she does not have to discuss these with others to maintain the friendship. The main objective of the novel replica allocation techniques is to reduce traffic overhead, while achieving high data accessibility

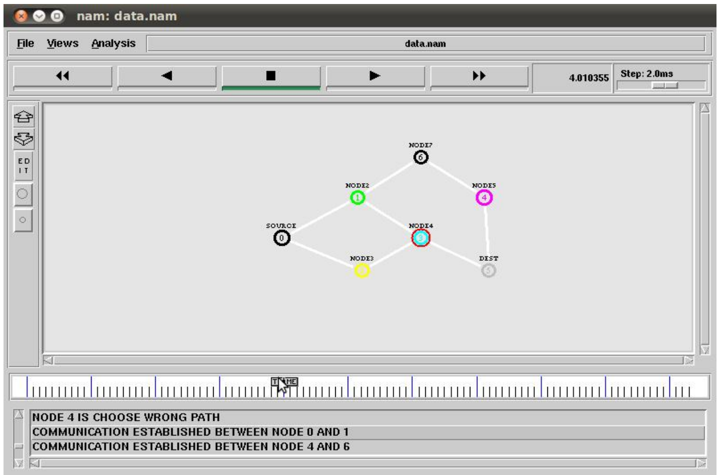
Steps for Building the SCF-tree
1. Consider the network topology.
2. In this network, each node has a parameter depth of the SCF tree.
3. When a particular node builds its own SCF tree, it first appends the nodes that are connected to the appropriate node by one hop to its child nodes.
4. Then the appropriate node checks recursively the child nodes of the appended nodes, until the depth of the tree is equal to the parameter.

SCF Tree Based Replica Allocation
After constructing the SCF-tree, each node allocates replica at its own discretion. At every relocation period, each node determines replica allocation individually without any communication with other nodes. The memory space of each node may be divided into two parts: s area Ms and public area Mp. Each node may use its own memory space Mi freely as Ms and/or Mp. In each node, Ms will be used for data of local interest (i.e., to reduce query delay), while Mp for public data is asked to hold data by other node(s) (i.e., to improve data accessibility). A type-2 node uses Mi for only Ms, whereas a type-3 node uses Mi for Ms and Mp. Consequently, each node allocates replicas in descending order of its own access frequency. This is quite different from existing group based replica allocation techniques (e.g., DCG in [5]) where replicas are allocated based on the access frequency of group members. Each node Ni executes this algorithm at every relocation period after building its own SCF-tree. At first, a node determines the priority for allocating replicas. The priority is based on Breadth First Search (BFS) order of the SCF-tree. After allocating a replica to the last target node, the next node will be the next target in a round-robin manner. The target node will be the expected node in our strategy. Since a node allocates a replica to the target node in its SCF-tree once during a single relocation phase, a node has at most one expected node for each replica. When its own Ms is not full, Ni allocates replica to its Ms first. When its own Ms becomes full, the node requests replica allocation to nodes in its SCF-tree in the order of priority. In our allocation technique, if Ms is full and Mp is not full, a node may use Mp for data items of local interest temporarily. However, public data cannot be held in Ms.

Steps for forming the SCF tree Based Replica Allocation
1. Consider the SCF tree for each node in the network.
2. The SCF tree is based on only partial selfishness node.
3. Make the priority of the node to allocate the replica using Breadth First Search function.
4. If the selfish area of the node Ms is not full then, allocate replica of the data to the selfish are Ms. Otherwise, allocate replica of the data to the target node.
5. If the public area of the node Mp is not full then, allocate replica of the data to the public area Mp.
6. If the node Nk requests for the allocation of Dq then, if the node Nk is in SCF tree TiSCF and Ni does not hold the data Dq.
7. If the public area of the node Mp is not full then, allocate the data Dq to Mp.
8. Otherwise, if the node Ni holds any replica of local interest in public area Mp then replace the replica with Dq;
9. Check the credit risk of the node nCRhi is greater than nCRki then replace the replica requested by the node Nh with Dq;

IV.RESULT ANALYSIS

In our work, the performance evaluation is done using network simulator (ns-2). Here, we created a mobile ad hoc network using AOMDV protocol. Then, we find out selfish node through credit risk method, watch dog method and also proposed novel replica allocation techniques. Finally evaluate it through the communication cost, detection time, query delay and data accessibility.
A mobile ad-hoc network with 7 nodes has been set up. Node 0 is considered as the source and node 5 as the destination. The mobile ad-hoc network is simulated using both AOMDV protocols for varying data rates ranging from 50bps to 150Mbps.



NODE 4 IS CHOOSE WRONG PATH
COMMUNICATION ESTABLISHED BETWEEN NODE 0 AND 1
COMMUNICATION ESTABLISHED BETWEEN NODE 4 AND 6

Fig. 1 NAM window for selfish node detection

In Fig. 1, shows that NAM window for selfish node detection. It consists of 7 nodes and each node represents in different colour. Here node 4 is detected as selfish node through self centered friendship tree (SCF) method and collaborative watch dog (CWD) method.

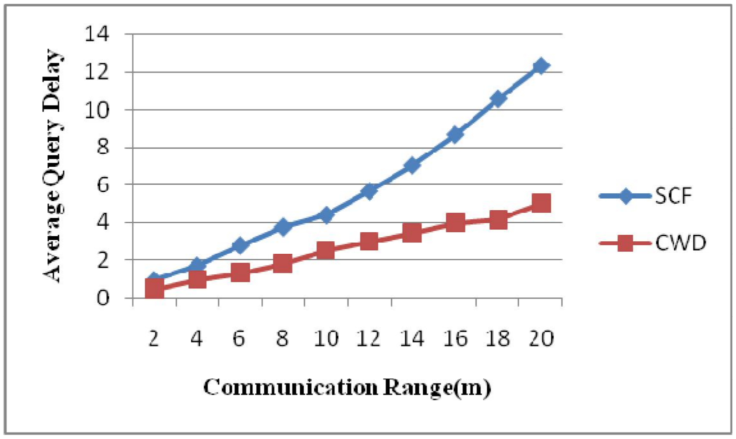


Fig. 2 Communication Range Vs Average Query Delay

In Fig. 2, it shows communication range in terms of average query delay. Average query delay is the total delay of successful data requests to the total number of data requests. This graph shows that, in self centered friendship (SCF) method and collaborative watch dog (CWD) method as the communication range increases then average query also increases. Collaborative watch dog (CWD) method has less average query delay as compared with self centered friendship (SCF) method. So from this we can conclude that collaborative watch dog (CWD) method is better as compared to self centered friendship (SCF) method.

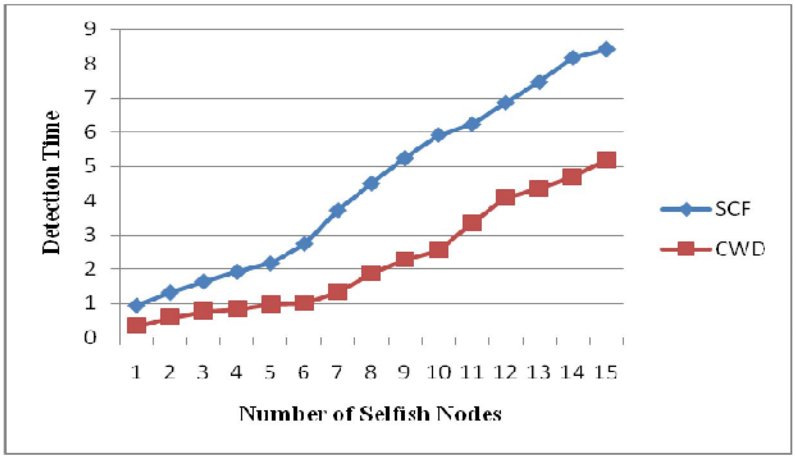


Fig .3 Number of Selfish Node Vs Detection Time

In Fig. 3, it shows that number of selfish node in terms detection time. This graph shows that, in self centered friendship (SCF) method and collaborative watch dog (CWD) method as the number of selfish nodes increases then detection time also increases. Collaborative watch dog (CWD) method has less detection time required for detecting selfish nodes as compared with self centered friendship (SCF) method. So from this we can conclude that collaborative watch dog (CWD) method is better as compared to self centered friendship (SCF) method.

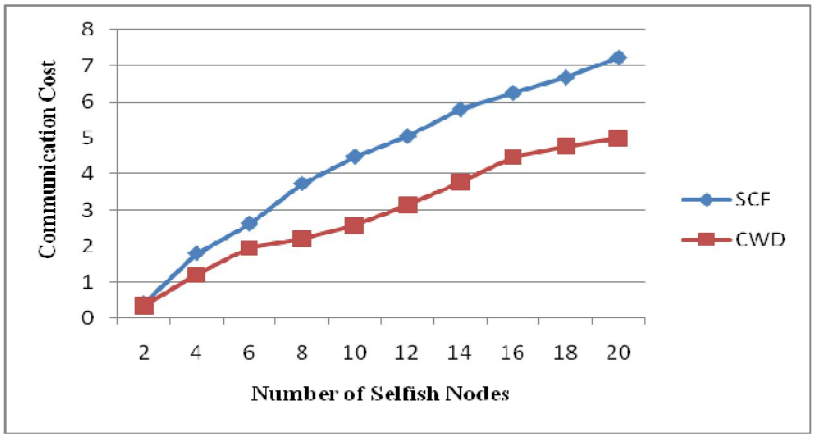


Fig. 4 Number of Selfish Node Vs Communication Cost

In Fig. 4, it shows that number of selfish nodes in terms of communication cost. Communication cost is the total hop count of data transmission for selfish node detection and replica allocation/relocation and their involved information sharing. This graph shows that, in self centered friendship (SCF) method and collaborative watch dog (CWD) method as the number of selfish nodes increases then total hop count of data transmission also increases. Collaborative watch dog (CWD) method has less total hop count of data transmission with more selfish nodes. So from this we can conclude that collaborative watch dog (CWD) method is better as compared to self centered friendship (SCF) method.

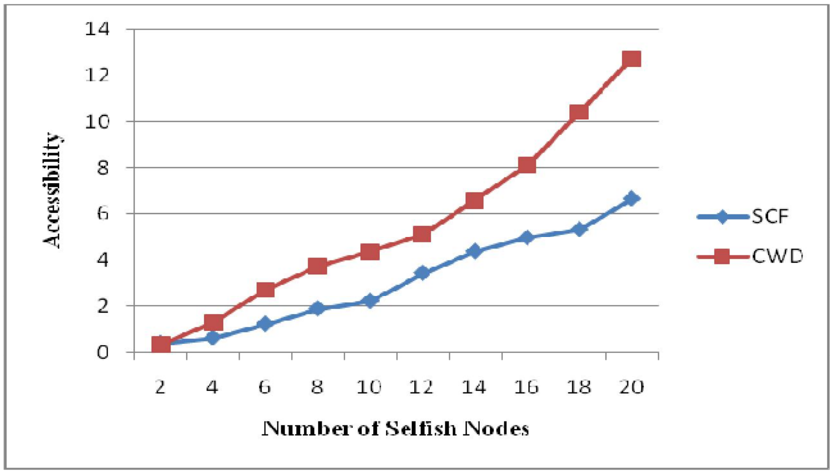


Fig .5 Number of Selfish Nodes Vs Data Accessibility

In Fig. 5, it shows that data accessibility with number of selfish nodes. Data accessibility is the ratio of the number of successful data requests to the total number of data requests. It improves with the wide range of communication, since more selfish node becomes connected. This graph shows that, in self centered friendship (SCF) method and collaborative watch dog (CWD) method as the number of selfish nodes increases then accessibility also increases. Collaborative watch dog (CWD) method has better accessibility as compared with self centered friendship (SCF) method. So from this we can conclude that collaborative watch dog (CWD) method is better compared to self centered friendship (SCF) method.

V.CONCLUSION

This project analysis is the efficiency of a credit risk (CR) method and collaborative watch dog method (CWD), to detect the selfish nodes in MANETs. Here a comparative study is done using self centered friendship (SCF) and collaborative watch dog method (CWD). The analysis is done in terms of query delay, detection time, communication cost and data accessibility. While comparing both methods collaborative watch dog (CWD) method is better than self centered friendship (SCF) method. Network simulator (ns-2) is used for the comparative study of MANETs different mobility pattern using suitable metrics like data accessibility, average query delay, communication cost.

REFERENCES

[1] Byung-Gon Chun, Kamalika Chaudhuri, Hoeteck Wee, Marco Barreno, Christos H. Papadimitriou, and John Kubiatowicz, “Selfish Caching in Distributed Systems: A Game-Theoretic Analysis ,”Appears in Proceedings of the 23rd ACM Symposium on Principles of Distributed Computing, July 2004.

[2] Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi and Pierangela Samarati, “Managing and Sharing Servents Reputations in P2P Systems,” IEEE Transaction on Volume 15, Issue 4, July- August 2003.

[3] Guohong Cao, Liangzhong Yin Chita R. Das,“Cooperative Cache- Based Data Access in Ad Hoc Networks,” Published by the IEEE Computer Society, pp.32-38, February 2004.

[4] Hales, D, “From Selfish Nodes to Cooperative Networks -- Emergent Link-Based Incentives in Peer-to-Peer Networks, in 'Peer-to-Peer Computing,” pp. 151-158, 2004.

[5] Jae-Ho Choi, Kyu-Sun Shim, SangKeun 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.

[6] K. Paul and D. Westhoff, “Context Aware Detection of Selfish Nodes in DSR Based Ad-Hoc Networks,” Proc. IEEE Global Telecomm. Conf., pp. 178-182, 2002.

[7] 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.

[8] 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.

[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] Mei Li Wang-Chien Lee an and Sivasubramaniam ,”Efficient Peer-to-Peer Information Sharing over Mobile Ad Hoc Networks ,”In: Proceedings of the 2nd Workshop on Emerging Applications For Wireless And Mobile Access (MobEA 2004), in conjuction with the World Wide Web Conference (WWW) May 2004.

[11] N. Laoutaris, O. Telelis, V. Zissimopoulos, and I. Stavrakakis, "Distributed Selfish RepLication," IEEE Trans. Parallel and Distributed Systems, vol. 17, no. 12, pp. 1401-1413, Dec. 2006.

[12] N. Laoutaris, G. Smaragdakis, A. Bestavros, I. Matta, and I. Stavrakakis, "Distributed Selfish Caching," IEEE Trans. Parallel and Distributed Systems, vol. 18, no. 10, pp. 1361-1376, Oct. 2007.

[13] S.Bhuvaneshwari, Prof.M.Suguna,"Identifying and handling of false alarm in selfish replica allocation ,"Vol 2 Issue 4 April 2013 ISSN 2278-733X.

[14] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad hoc Networks," Proc. ACM MobiCom, pp. 255-265, 2000.

[15] 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.

[16] 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.

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

[18] T. Hara and S.K. Madria, "Data Replication for Improving Data Accessibility in Ad Hoc Networks," IEEE Trans. Mobile Computing, vol. 5, no. 11, pp. 1515-1532, Nov. 2006.

[19] T. Hara and S.K. Madria, "Consistency Management Strategies for Data Replication in Mobile Ad Hoc Networks," IEEE Trans.Mobile Computing, vol. 8, no.7, pp. 950-967, July 2009.

[20] Y. Yoo and D.P. Agrawal, "Why Does It Pay to be Selfish in a MANET," IEEE Wireless Comm., vol. 13, no. 6, pp. 87-97, Dec. 2006.