Detection of False Alarm in Handling of Selfish Nodes in Mobile Ad Hoc Networks

Karthik M[1] Jyothish K John[2] Leenu Rebecca Mathew[3] Tibin Thomas[4]
[1,3]M. Tech. [2,4]Assistant Professor
[1, 2, 3, 4]Department of Computer Science and Engineering
[1, 2, 3, 4]Federal Institute of Science and Technology (FISAT), Angamaly, India

Abstract--- An mobile Ad Hoc network is a collection of mobile nodes. They does not have any existing infrastructure and they does not have any centralized administrator. So the MANET is self-creating, self-organizing and self-administrative wireless network. In MANET each node act as router. In practice some of the nodes may act as the selfish nodes. These nodes use the network and its services but they do not cooperate with other nodes. Such selfish nodes do not consume any energy such as CPU power, battery and bandwidth for retransmitting the data of other nodes. They will preserve the resources for their own use. Several data replication techniques have been proposed to minimize performance degradation. Most of them assume that all mobile nodes collaborate fully in terms of sharing their memory space. In reality, however, some nodes may selfishly decide only to cooperate partially, or not at all, with other nodes. These selfish nodes could then reduce the overall data accessibility in the network

I. INTRODUCTION

An mobile Ad Hoc network is a collection of mobile nodes. They does not have any existing infrastructure and they does not have any centralized administrator. So the MANET is self-creating, self-organizing and self-administrative wireless network. In MANET each node act as router. These nodes in the network are responsible discovering the path to the particular node and forward the data to that node. Since the nodes in the network have the capability of moving so the infrastructure of network will change rapidly. Each node in the MANET will forward the data to other node but some nodes will not forward the data packet to other nodes they are called the selfish nodes. The selfish node [1] in the network will cause the different problems in network.

Each node in the network has its own resources such as bandwidth, energy, local CPU time and transmission power etc. The selfish nodes will reduce the cooperation among the nodes in the network. While the selfish nodes [11] trying to preserve their resources such as battery life or bandwidth, they will get the benefits from the networks. The selfish node will forward its own data packet but it will refuse to forward the data packet of the other nodes. The selfish node can perform the actions such as do not participate in the routing process, turn off the power when there is no communication with other nodes, do not send hello or no reply messages, does not unicast or broadcast the Route Error packet, selectively drop the data packets etc. The selfish nodes in the network will reduce the packet delivery ratio by dropping the data packets. The selfish node detection is important to increase the performance of the ad hoc networks in practical applications.

In ad hoc networks, since mobile hosts move freely, disconnections occur frequently, and this causes frequent network partition. If a network is partitioned into two networks due to the migrations of mobile hosts, mobile hosts in one of the partitions cannot access data items held by mobile hosts in the other. Thus, data accessibility in ad hoc networks is lower than that in conventional fixed networks. In ad hoc networks, it is very important to prevent the deterioration of data accessibility at the point of network partition. A possible and promising solution is the replication of data items at mobile hosts which are not the owners of the original data. Since mobile hosts generally have poor resources, it is usually impossible for them to have replicas of all data items in the network .here we focus on the selfish node detection in the case of replication allocation [2] [3]

There are several nodes in the MANET and they will participate in the data transmission and data forwarding between the source and destination nodes .We can define the behavioural states of nodes [6] as three types in MANET from the viewpoint of certain constraints at memory space.

Type-1 node: The nodes are non-selfish nodes .These nodes can hold replicas allocated by other nodes within the limits of their memory space

Type-2 node: The nodes are fully-selfish nodes .These are the nodes which do not hold replicas allocated by other nodes, but it will allocate replicas to other nodes for their own accessibility.

Type-3 node: The nodes are partially- selfish nodes. These partially selfish nodes would use their own memory space partially for allocated replicas by other nodes. Their memory space may be separated logically into two parts: one is selfish area and another one is public area. These partially selfish nodes allocate replicas to other nodes for their accessibility.

The detection of the type-3 nodes is always complex, since they are not always selfish. In some situation, a type-3 node may be considered as non selfish , since the node shares part of its memory space. But in our paper, we have considered it as selfish node only, since these nodes also leads to the selfish replica allocation problem. Note that selfish and non-selfish nodes perform the same procedure when they receive a data access request [13].

II. PROPOSED STRATEGY

This chapter focuses on the mobile ad hoc network having selfish nodes and the way of replica allocation in the system helps to solve the data accessibility problem and about the simulation of system. When a node Ni makes an access request to a data item typically sending a query, the particular node will checks its own memory space first. If

the requested item is present in its own local memory space then the request got successful. If it does not hold the original or replica, the request will be broadcasted to other nearby nodes which are connected to the node Ni. The request is also successful when Ni receives any reply from at least one node connected to Ni with one hop or multiple hops of nodes, which holds the original or replica of the targeted data item. Otherwise, the request fails.

A. *Detecting Selfish Nodes*

In our strategy, each node calculates a CR score [4] for all the nodes to which it is connected as its neighbourhood. Each node at the network shall estimate the "degree of selfishness" for all of its connected nodes based on the CR score. We describe selfish features that may lead to the selfish replica allocation problem to determine both expected value and expected risk. Credit risk will be calculated as the ratio of expected risk to the expected value.[12]

Selfish features are classified into two groups: query processing-specific and node-specific feature. In the query processing-specific feature, we develop the ratio of selfishness alarm which is the ratio of Node N1's data request being not served by the expected node Nk due to Nk's selfishness in its memory space.

The query processing-specific feature can represent the expected risk of a node [1]. To effectively identify the expected node, Node N1 should know the (expected) status of other nodes' memory space. The SCF-tree-based replica allocation techniques support this assumption. Node-specific features can be explained by considering the following case: A selfish node may share part of its own memory space, or a small number of data items, like partially selfish node. In this occasion, the size of shared memory space or the number of shared data items can be used to represent the degree of selfishness. The features can be used to represent the expected value of a node. Table 1 shows the credit risk values for the different nodes [14].

N_i	N_k					
	N_1	N_2	N_3	N_4	N_5	N_6
N_1	·	0.30	0.85	0.80	0.45	0.22
N_2	0.40	·	0.80	0.90	0.30	0.50
N_3	0.25	0.35	·	0.75	0.65	0.75
N_4	0.45	0.44	0.51	·	0.23	0.37
N_5	0.30	0.60	0.85	0.40	·	0.21
N_6	0.40	0.50	0.90	0.52	0.30	·

Table 1: Credit Risk Table

B. *Building SCF-TREE*

The main objective of our novel replica allocation techniques is to attain high data accessibility while reduce in traffic overhead. High Data accessibility is the prominent concern in all networks. If the replica allocation techniques allocate replica of the specified data item without any other node's concern, the traffic overhead will decrease. Since the SCF-tree [5] consists of only non-selfish nodes, we need to measure the degree of selfishness to apply in real-world friendship management to replica allocation in a MANET.

We use the value of credit risk for building the tree. Before constructing or updating the SCF tree, node Ni eliminates selfish nodes from the base group INi .The key strength of the SCF-tree-based replica allocation techniques is that it can minimize the communication cost, even though achieving high data accessibility. The high data accessibility is possible because each node detects selfishness and makes replica allocation at its own pleasure, without forming any group in the network.

Each node has a parameter d, the depth of SCF tree. When N1 builds its own SCF-tree, N1 first appends the nodes that are connected to N1 by one hop to N1's child nodes. Then, N1 checks recursively the child nodes of the combined nodes, until the depth of the tree is equal to d. we assume that all nodes are non-selfish nodes for simplicity [1]. Figure 2 shows the sample topology of the manet and figure 3 shows the SCF tree based on the node N1

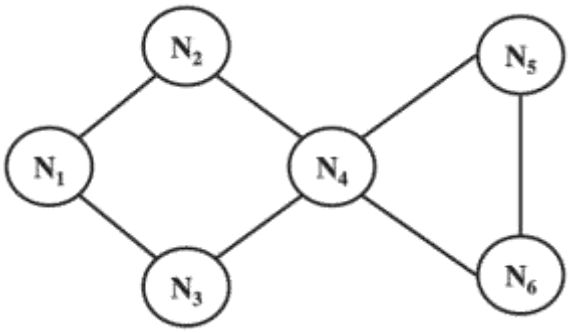

Fig. 2: sample topology

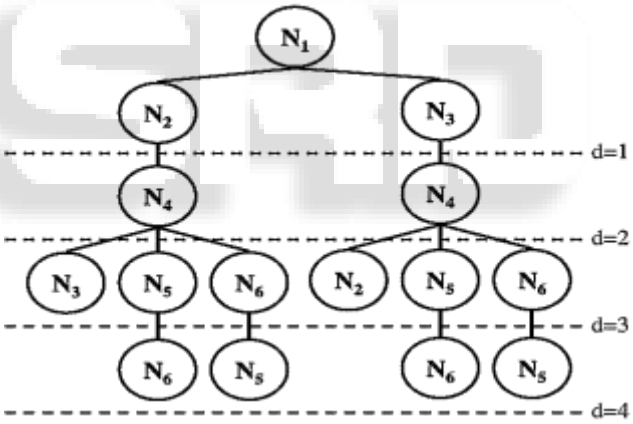

Fig. 3: SCF tree for node N1

As by the figure the SCF-tree may have multiple routes for some nodes from the root node and that confer high stability. At every specific relocation period, each node in the network updates its own SCF-tree based on the network topology of that moment.

C. *Replica Allocation*

Each node allocates replica [7] at its discretion based on Credit Risk Table and the SCF tree for the corresponding. When each node receives a request for replica allocation from Nk during a specific relocation period, the specific node solely determines whether to accept or reject the request. If the request is accepted by other node, the specific node will maintains its Mp based on the nCRki given by credit risk Table. If the highest nCRhi among the nodes which allocated replica to Ni, is greater than nCRki, Ni replaces replica allocated by node with replica requested by Nk. Each node Ni executes this algorithm at every

relocation period after building its own SCF-tree

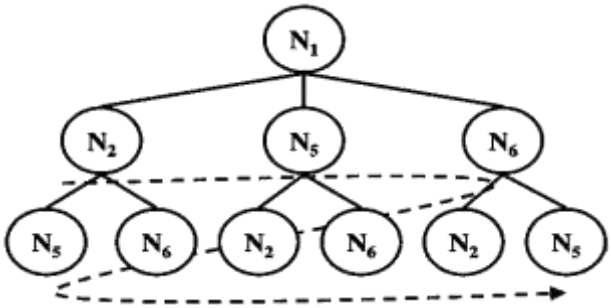

Fig. 4: SCF based replica allocation

The node determines the priority for allocating replica items. The allocating replica priority is based on Breadth First Search (BFS) order of the SCF tree [1]. The dotted arrow in Fig. 4 represents the priority for allocating replica. For example, in Fig.4, N1 selects N2 as the first target of the allocation in the tree. After allocating a replica to the last target node (i.e., N5 in Fig), the first node of the tree, N2 will be the next target in a round robin manner.

D. False Alarm Detection

The false alarm will be differentiated from the overall selfishness alarm. If any alarm generated means we should verify the reason of the alarm. We should calculate the degree of selfishness again and to confirm the behaviour of selfish nodes at the network. If the number of selfish nodes exceeds the threshold value means it will get confirm as overall selfishness alarm else the alarm has been raised because of the network disconnections. We should diagnose the network disconnections by use of false detection algorithm. If it became true we should neglect the alarm with of less concern. The detection of this false alarm leads to better performance in the overall network.

Node Nk sends the request to access the data item to node Ni. Node Nk moves out of range of Ni thereby fails to receive the request. Actually, the reply from expected node Nk holding a target data item may not reach Ni due to disconnection, not its selfishness. In the current solution, however, we assume that Ni cannot tell Nks selfishness from network disconnection, since their impacts are identical to Ni,i.e., data inaccessibility. The identification and handling of a false alarm is one of interesting, promising future works .Data inaccessibility [10] due to link failure can be identified by employing communication overhearing process over the period. Each node in the network should update the neighbors list periodically. Each node should overhear and ensure the connection with the requested node by overhearing the communication of requested node with any other nodes in the network. If requestor does not get any signal related to requested node, requestor can conclude that the link failure has been occurred. Using this feature false alarm can be detected

III. RESULTS

We evaluate our strategy using four performance metrics:

A. Overall selfishness alarm:

This is the ratio of the overall selfishness alarm of all nodes to all queries that should be served by the expected node in the entire system.

B. Communication cost:

This is the total hop count of data transmission for selfish node detection and replica allocation/relocation, and their involved information sharing.

C. Average query delay:

This is the number of hops from a requester node to the nearest node with the requested data item. it is the total delay of successful requests divided by the total number of successful requests

D. Data accessibility:

This is the ratio of the number of successful data requests to the total number of data requests

In addition to the basic SCF based replication, alternative replica allocation techniques can be developed based on the SCF-tree structure. Thus, we propose a set of replica allocation techniques, as follows

1) SCF-tree based replica allocation with degree of selfishness (SCF-DS):

This technique takes into account the degree of selfishness in allocating replicas. That is, less selfish nodes should be visited first at the same SCF-tree level.

2) SCF-tree based replica allocation with closer node (SCF-CN):

This technique allocates more replicas to the closer nodes in the SCF-tree.

3) Extended SCF-tree based replica allocation (eSCF):

In this technique eSCF-tree includes selfish nodes, as well as non-selfish nodes we simulate and compare the proposed replica allocation strategies (i.e., SCF, SCF-DS ,and eSCF) with the following techniques:

4) Static Access Frequency (SAF) [8] :

Each node allocates replica based only on its own access frequency, without considering or detecting selfish nodes.

5) Dynamic Connectivity-based Grouping with detection (DCG) [9]:

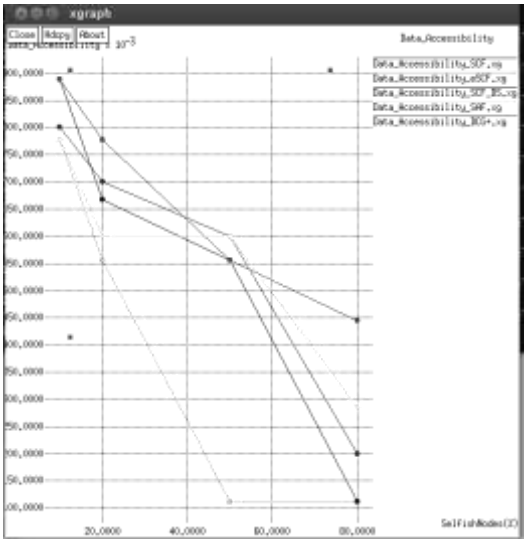

Fig. 5: Data Accessibility

The technique combines DCG with our 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.

The graph shows the Data accessibility of the mobile ad hoc network under different methods .The data accessibility is find out by varying selfish nodes .As the percentage of selfish nodes increases in the network the data accessibility reduces .The data accessibility is high when the selfish node percentage is 20.

The graph 6 shows the Communication Cost of the mobile ad hoc network under different methods .The Communication Cost is find out by varying selfish nodes .As the percentage of selfish nodes increases in the network the Communication Cost increases, because the number of hopes needed to reply a request is increases as the selfish node increases. The Communication Cost is high when the selfish node percentage is 80.

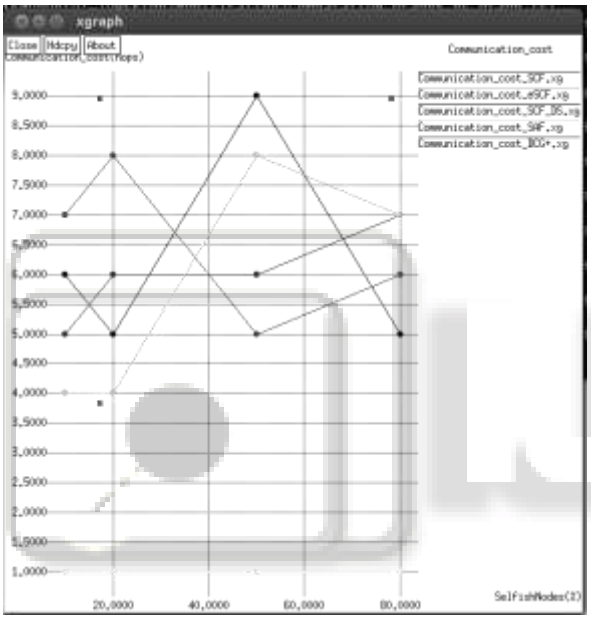

Fig. 6: Communication cost

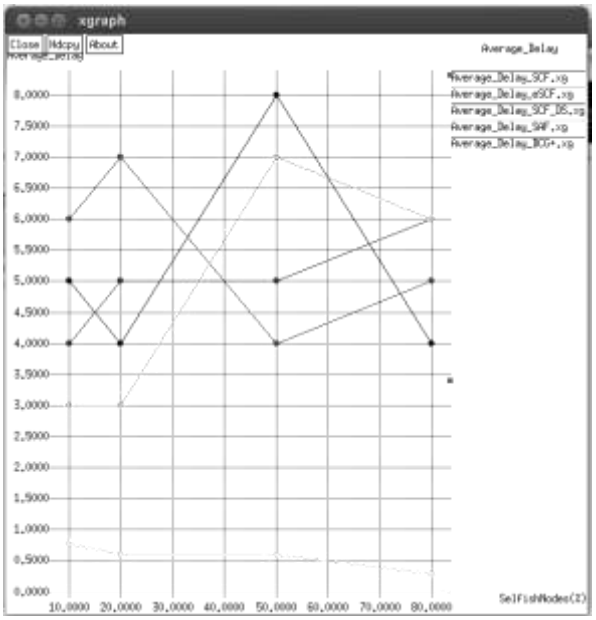

Fig. 7: Average Query Delay

The graph shows the Average query delay of the mobile ad hoc network under different methods .The Average query delay is finding out by varying selfish nodes. As the percentage of selfish nodes increases in the network the Communication Cost increases, because the number of hopes needed to reply a request is increases as the selfish node increases so the query delay also increases.

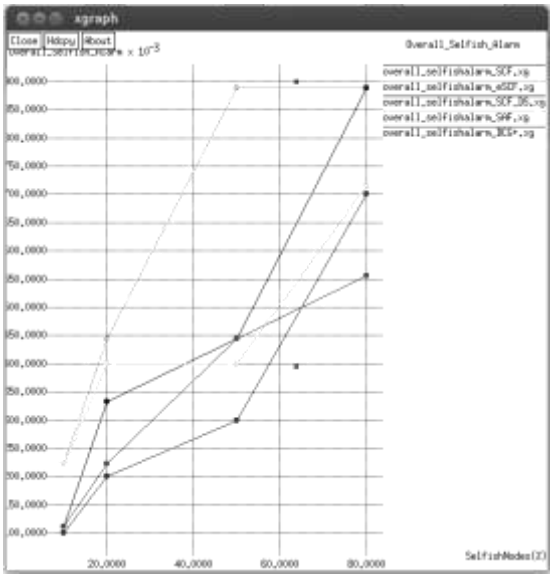

Fig. 8: Overall Selfish Alarm

The graph shows the overall selfish alarm of the mobile ad hoc network under different methods. The Overall selfish alarm is finding out by varying selfish nodes. As the percentage of selfish nodes increases in the network the Overall selfish alarm increases, because the number of selfish nodes increases the data items held them are not shared to others so the request will not be satisfied by the particular node. The Overall selfish alarm is high when the selfish node percentage is 80.

IV. CONCLUSION

In contrast to the network viewpoint the addressed the problem of selfish nodes from the replica allocation perspective. this problem is termed as selfish replica allocation .selfish replica allocation could lead to overall poor data accessibility in a MANET. The proposed a selfish node detection method and novel replica allocation techniques to handle the selfish replica allocation appropriately. 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 ones friends completely at one's own discretion .The credit risk from economics is used to detect selfish nodes. Every node in a MANET calculates credit risk information on other connected nodes individually to measure the degree of selfishness .Then create the topology for replication allocation using the SCF tree .Then replication allocation is performed .False alarm which is falsely notifying genuine node as selfish node is detected by using overhearing process. Therefore proposed approach accurately identifies the selfish behaviour. Proposed approach is compared with existing approach by evaluating and comparing the results obtained during the simulation done using Network

Simulator ns-2. Proposed approach yields superior performance when compared to existing approach.

REFERENCE

[1] E. Adar and B.A. Huberman, "Free Riding on Gnutella," First Monday, vol. 5, no. 10, pp. 1-22, 2000.
[2] H. Li and M. Singhal, "Trust Management in Distributed Systems," Computer, vol. 40, no. 2, pp. 45-53, Feb. 2007.
[3] A. Mondal, S.K. Madria, and M. Kitsuregawa, "An Economic Incentive Model for Encouraging Peer Collaboration in Mobile- P2P Networks with Support for Constraint Queries," Peer-to-Peer Networking and Applications, vol. 2, no. 3, pp. 230-251, 2009.
[4] L.J. Mester, "What's the Point of Credit Scoring?" Business Rev.,pp. 3-16, Sept. 1997
[5] R.F. 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.
[6] P. Michiardi and R. Molva, "Simulation-Based Analysis of Security Exposures in Mobile Ad Hoc Networks," Proc. European Wireless Conf., pp. 1-6, 2002.
[7] T. Hara, "Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility," Proc. IEEE INFOCOM, pp. 1568- 1576, 2001.
[8] 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.
[9] 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.
[10] 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.
[11] H. Miranda and L. Rodrigues, "Friends and Foes: Preventing Selfishness in Open Mobile Ad hoc Networks," Proc. IEEE Int'l Conf. Distributed Computing Systems Workshops, pp. 440-445, 2003
[12] S.U. Khan and I. Ahmad, "A Pure Nash Equilibrium-Based Game Theoretical Method for Data Replication across Multiple Servers,"IEEE Trans. Knowledge and Data Eng., vol. 21, no. 4, pp. 537-553, Apr. 2009.
[13] 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.
[14] 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.

