ISSN 2320-6802

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

SELFISH NODE DETECTION USING REPLICA ALLOCATION TECHNIQUES AND SCF-TREE IN MANET

[1]K.Indumathi, [2]N.Jayalakshmi, [3]S.Kartiga

[1,2,3]M.Tech Networking

[1,2]Sri Manakula Vinayagar Engineering College, Madagadipet, Puducherry-605107.

indumathy2610@gmail.com[1], jaya.nasa40@gmail.com[2], sparkarthi@gmail.com[3]

Abstract: *In Mobile ad-hoc Network (MANET) 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. The selfish nodes are engaged to reduce data availability and produce high communication cost in terms of query processing. Many selfish node detection methods are found to detect the nodes which do not participate in packet forwarding but they fail to detect the selfish nodes which does not allocate replica for the purpose of other nodes. The methods are provided to detect selfish nodes in terms of allocating replica to other nodes. The methods are categorized according to detect the selfish nodes and reduce the impact of that nodes in mobile ad hoc network. The key features discussed are the selfish nodes and numerous replica allocation techniques. The selfish node detection algorithm that considers partial selfishness and novel replica allocation techniques to properly cope with selfish replica allocation. The proposed approach is the Self-Cantered Friendship tree (SCF-Tree) performs traditional cooperative replica allocation techniques in terms of data accessibility, communication cost, and average query delay.*

Keywords: *Mobile ad hoc Networks, Degree of Selfishness, Selfish Replica Allocation.*

1. INTRODUCTION

In MANETs (Mobile Ad Hoc Networks) every node acts as a router and communicates with each other nodes. If the source and the destination mobile hosts are not in the coverage area, data packets are forwarded to the destination host through other nodes which exist between the two mobile hosts. MANETs does not require any infrastructure and base station. According to [1] [8] MANETs are applicable in many situations such as battlefield and disaster area. In ad hoc network, as all the nodes are having mobility, they move freely. This mobility causes frequent network partitions hence data accessibility in ad hoc networks is lower than the fixed networks. The nodes which are not willing to forward packets and share their memory space are called selfish nodes. The selfish node that does not allocate data items for the purpose of other nodes is called selfish replica allocation.

The selfish nodes allocate data items that are highly accessed by it and do not consider other nodes during replica allocation. Selfish nodes reduce the data accessibility of other nodes in query processing. The selfish nodes do not satisfy neighbour nodes by giving required data to them. According to [1] [9] nodes can be divided into three types they are,

- Non selfish nodes
- Fully selfish nodes
- Partially selfish nodes

Non selfish nodes allocate their memory space completely for the purpose of other nodes. Selfish nodes do not allocate their memory space for the purpose of other nodes. Partially selfish nodes allocate minimum portion of their memory space for the purpose of other nodes and remaining for the benefit of own node. Minimizing the effects of selfish nodes will be important to increase the data accessibility between the nodes.

2. EXISTING REPLICA ALLOCATION TECHNIQUES

The replica allocation techniques such as Static Access Frequency (SAF)[2], Dynamic Access Frequency and Neighbourhood (DAFN)[2][3], and

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

Dynamic Connectivity-based Grouping (DCG) [2] failed to consider the selfish nodes, Hence improvements have to be made in replica allocation techniques that consider selfish replica allocation. The friendship manner replication has to be done in relocation period produce the new technique called SCF-tree based replica allocation [1]. The SCF Tree based allocation techniques are inspired by human friendship management in the real world. The main objective of these SCF Tree replica allocation techniques is to reduce traffic overhead, while achieving high data accessibility. [10] [11].

A. The SAF (Static Access Frequency) Method

In SAF method [2], the nodes allocate replica of data items according to the access frequencies of that data items. The access frequency of each mobile host to each data item is shown in Fig.1 [1] shows the result of executing the SAF method. Mobile nodes with the same access frequencies to data items allocate the same replica. The SAF method causes low data accessibility when many mobile hosts have the similar access characteristics hence some of the data items to be duplicated in many nodes.

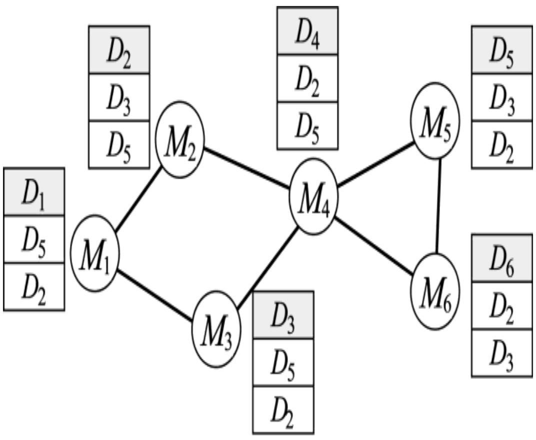

Figure1: SAF Replica Allocation

B. The DAFN (Dynamic Access Frequency and Neighbourhood) Method

To overcome the problem of replica duplication in the SAF method, a new method of replica allocation called DAFN method [4] was developed. It eliminates the replica duplication among neighbouring mobile hosts [6].

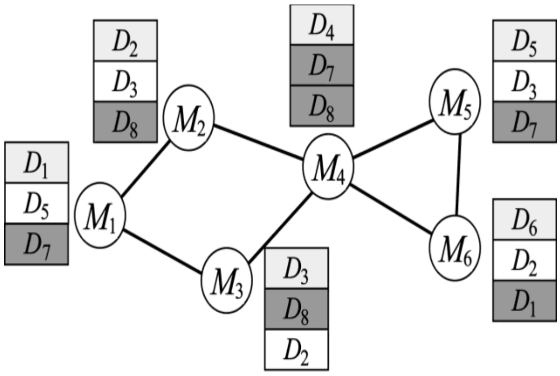

Figure 2: DAFN Replica Allocation

The algorithm of DAFN [2] 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 having the same data item then the node having replica changes it to another replica which having high access frequency.

At each relocation period, the mobile nodes exchange information about replicas allocated in the memory space. So the overhead and the traffic are high compared with the SAF method.

C. The DCG (Dynamic Connectivity and Grouping) Method

The DCG method [2-3] 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.

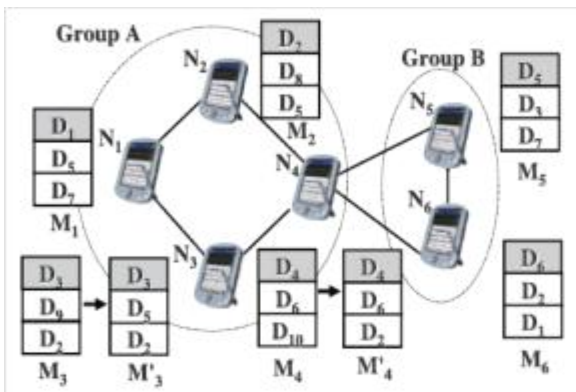

Figure 3: Dynamic Connectivity Grouping

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

Fig. 3 illustrates an existing replica allocation scheme, DCG [1], where nodes N1; N2..; N6 maintain their memory space M1; M2...M6, respectively. (In Fig. 3, a straight line denotes a wireless link, a gray rectangle denotes an original data item, and a white rectangle denotes a replica allocated. DCG seeks to minimize the duplication of data items in a group to achieve high data

3. PROPOSED SYSTEM

A. Overview

Proposing 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 one's friends completely at one's own discretion [1] .We applied the notion of credit risk from economics to detect selfish nodes. Every node in a MANET calculates credit risk information on other connected nodes individually to measure the degree of selfishness. Since traditional replica allocation techniques failed to consider selfish nodes, we also proposed novel replica allocation techniques. The selfish node is detected by the self replica allocation. They are based on the concept of a self-centred friendship tree (SCF-tree) and its variation to achieve high data accessibility with low communication cost in the presence of selfish nodes. With the measured degree of selfishness, we propose a novel tree that represents relationships among nodes in a MANET, for replica allocation, termed the SCF-tree. The SCF-tree models human friendship management in the real world. 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 [12] [1].

The technical contributions of this paper can be summarized as follows.

- *Recognizing the selfish replica allocation problem:* We view a selfish node in a MANET from the perspective of data replication, and recognize that selfish replica allocation can lead to degraded data accessibility in a MANET.
- *Detecting the fully or the partially selfish nodes effectively:* We devise a selfish node detection method that can measure the degree of selfishness.
- *Allocating replica effectively:* We propose a set of replica allocation techniques that use the self-centred friendship tree to reduce communication cost, while achieving good data accessibility.

B. Proposed Strategy

Our proposed strategy consists of 1) detecting the selfish nodes, 2) building the SCF-Tree, 3) Allocating the replica

1. Selfish Node Detection

The notion of credit risk can be described by the following equation: Credit Risk = expected risk / expected value in this strategy, 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. The Selfish features are divided into two categories: node specific and query processing-specific. The Node specific features can be used to represent the number of shared items & shared memory space for the node. The size of N_k's shared memory space, denoted as SSi^K , and the number of N_k's shared data items, denoted as ND^k i , observed by a node Ni ,are used as node-specific features. The node-specific features can be used to represent the expected value of a node. For instance, when node N_i observes that node N_k shares large SSi^K and ND^k i , node N_k may be treated as a valuable node by node Ni. As the query processing-specific feature, utilize the ratio of selfishness alarm of N_k on N_i, denoted as Pi^k , which is the ratio of N_i's data request being not served by the expected node N_k due to N_k selfishness in its memory space (i.e., no target data item in its memory space).Thus, the query processing-specific feature can represent the expected risk of a node. For instance, when Pi^k gets larger, node Ni will treat N_k as a risky node because a large Pi^k means that N_k' cannot serve Ni's requests due to selfishness in its memory usage. The value of the Cri^k is the credit risk of node Ni. § is the threshold value of node N_i. α is the system parameter, where $0 \leq \alpha \leq 1$.

Algorithm 1

Pseudo code for selfish node detection

```
detection()
{
  for (each connected node Nk )
{
   if (nCR^k i < §)
   Nk is marked as non-selfish;
   else Nk is marked as selfish;
}
```

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

wait until replica allocation is done;
for (each connected node N_k)
{
if (N_i has allocated replica to N_k)
{
ND^k i= the number of allocated replica;
SS^k i = the total size of allocated replica;
}
else
{
ND^k i =1;
SS^k i = the size of a data item;
} } }

2. Building SCF-Tree

The SCF-tree based replica allocation techniques are inspired by human friendship management in the real world, where each person makes his/her 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. Since the multiple routes confer high stability we allocate more replicas to the nodes that have multiple routes from the root node. [7] The main objective of the novel replica allocation techniques is to reduce traffic overhead, while achieving high data accessibility. Before constructing the SCF-tree, each node makes its own partial topology graph G_i = (IN_i, IL_i), which is a component of the graph G. Gi consists of a finite set of the nodes connected to Ni and a finite set of the links, where Ni € IN_i , IN_i _ IN, and IL_i _ IL. Based on Gi^{ns}, Ni builds its own SCF-tree, denoted as Ti^{SCF}. Algorithm 2 describes how to construct the SCF-tree. Each node has a parameter d, the depth of SCF-tree. When Ni builds its own SCF-tree, Ni first appends the nodes that are connected to Ni by one hop to Ni's child nodes. Then, Ni checks recursively the child nodes of the appended nodes, until the depth of the SCF-tree is equal to d. Fig.4 illustrates the network topology

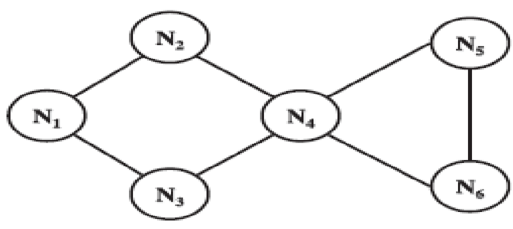

Figure 4: Sample Topology G

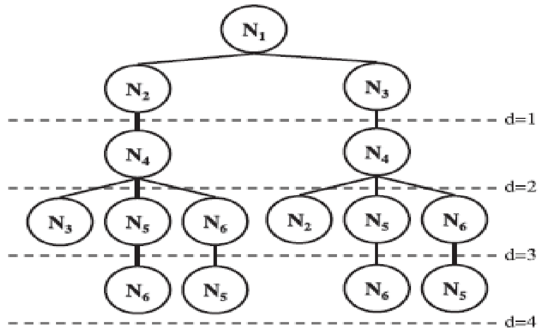

Figure 5: SCF-Tree of N_1

Algorithm 2

Pseudo code to build SCF-tree

constructScfTree()
{
append Ni to SCF-tree as the root node;
checkChildnodes(Ni);
return SCF-tree;
}
Procedure checkChildnodes(Nj)
{
for (each node Na € INaj)
{
if (distance between Na and the root > d)
continue;
else if (Na is an ancestor of Nj in TiSCF)
continue;
else
{
append Na to Ti^{SCF} as a child of Nj;
checkChildnodes(Na);
} } }

4. Allocating the Replica

After building the SCF-tree, [7] [1] a node allocates replica at every relocation period. Each node asks non-selfish nodes within its SCF-tree to hold replica when it cannot hold replica in its local memory space. Since the SCF-tree based replica allocation is performed in a fully distributed manner, each node determines replica allocation individually without any communication with other nodes. Since every node has its own SCF-tree, it can perform replica allocation at its discretion.

Algorithm 3

Pseudo code for replica allocation
/*Ni executes this algorithm at relocation period */
replica_allocation()
{
L_i =make_priority($_{Ti}{}^{SCF}$);
for (each data item belongs to ID_i)

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

```
{
  if (M_s is not full)
  allocate replica of the data to Ms ;
  else{/*Ms is full */
  allocate replica of the data to the target node;
  /*the target node is selected from Li*/
  if (M_p is not full)
  allocate replica of the data to M_p;
} }
  while (during a relocation period)
{
  if (N_k requests for the allocation of D_q)
  replica_allocation_for_others (N_k;D_q);
} }
  Procedure make_priority (T_i^SCF)
{
 for (all vertices in (T_i^SCF)
{
 select a vertex in T_i^SCF in order of BFS;
 append the selected vertex_i d to L_i;
}
 return Li; }
 Procedure replica_allocation_for_others(N_k;D_q)
{
 if (N_k is in T_i^SCF and N_i does not hold D_q)
{
 if (M_q is not full) allocate D_q to M_p ;
 else{/*Mp is full*/
 if(N_i holds any replica of local interest in M_p)
 replace the replica with D_q ;
 else
{
/*N_h is the node with the highest _nCR_i^h
 among the nodes which allocated replica to M_p*/
  if (_nCR_i^h > _nCR_i^k )
  replace the replica requested by N_h with D_q;
} } } }
```

4. CONCLUSION

The traditional selfish node detection replica allocation techniques such as DCG, SAF and DAFN are failed to consider selfish nodes in terms of replica allocation. Hence improvements have to be made in replica allocation techniques that consider selfish replica allocation. A selfish node detection method and SCF tree replica allocation techniques to handle the selfish replica allocation is proposed. So the proposed technique SCF-tree can minimize the communication cost, while achieving high data accessibility.

References

[1] Jae-Ho Choi Kyu-Sun Shim, Sang Keun Lee, and Kun-Lung Wu, Fellow, "Handling Selfishness in Replica Allocation over a Mobile Ad Hoc Network" IEEE Transaction on Mobile Computing, Voume 11, No.2, February 2012.

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

[3] 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, November 2006.

[4] 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 Engineering., vol. 21, no. 4, pp. 537-553, April. 2009.

[5] S.U. Khan and I. Ahmad, "A Pure Nash Equilibrium-Based Game Theoretical Method for Data Replication across Multiple Servers," IEEE Transaction on Knowledge and Data Engineering., Volume. 21, No. 4, pp. 537-553, April. 2009.

[6] Jim Solomon Raja.D, Immanuel John Raja.J ,"A Survey on Selfishness Handling In Mobile Ad Hoc Network", International Journal of Emerging Technology and Advanced Engineering ISSN 2250-2459 Volume 2, Issue 11, November 2012.

[7] S.J.K. Jagadeesh Kumar, R.Saraswathi, "A Combined Credit Risk and Collaborative Watchdog Method for Detecting Selfish Node over Mobile Ad Hoc Network" International Journal of Advanced Research in Computer Science and Software Engineering ISSN: 2277 128X Volume 2, Issue 11, November 2012

[8]M.siva Lakshmi, R.Raja Sekhar "ODRS: Optimal Data Replication Scheme for Time Efficiency In MANETs" IOSR Journal of Computer Engineering IOSRJCE) ISSN: 2278-0661 Volume 5, Issue 1 (Sep-Oct. 2012), PP 21-27

[9] Y. Yoo and D.P. Agrawal, "Why Does It Pay to be Selfish in a MANET," IEEE Wireless Communication., Volume. 13, No. 6, pp. 87-97, December. 2006.

[10] T. Hara and S.K. Madria, "Consistency Management Strategies for Data Replication in Mobile Ad Hoc Networks," IEEE Transaction on Mobile Computing, Voume l. 8, No. 7, pp. 950-967, July 2009.

[11] S.-Y. Wu and Y.-T. Chang, "A User-Centered Approach to Active Replica Management in Mobile Environments," IEEE Transaction on Mobile Computing, Volume 5, No. 11, pp. 1606-1619, November 2006.

[12] L. Yin and G. Cao, "Balancing the Tradeoffs between Data Accessibility and Query Delay in Ad Hoc Networks," Proceedings IEEE International Symposium. Reliable Distributed Systems, pp. 289-298, 2004.