International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 3, March - 2013

Evaluation Of Overall Selfishness And Data Accessibility In Replica Allocation Over MANET

Divya S R*, Ciji K R**

*(Department of Information Science, Visvesvaraya Technological University, Bangalore-68

** (Department of Information Science, Visvesvaraya Technological University, Bangalore-68

ABSTRACT

MANET is a self-governing collection of mobile nodes forming a self-motivated topology which keeps changing. Due to the high mobility feature of nodes in ad-hoc network partitioning of networks occur frequently. Hence data accessibility is reduced. Due to disconnections in network there occurs a selfishness alarm known as false alarm. In this paper we concentrate on improving accessibility to data and reducing the overall selfishness alarm in MANET.

Keywords- **data accessibility, MANET, selfish replica allocation, selfishness alarm**

1. INTRODUCTION

In MANET, every node is mobile and acts as a router, and they communicate with each other. Even though the source and the destination nodes are not in the communication range of each other, data are forwarded to the destination node by routing through intermediate mobile hosts which are between the source and destination.

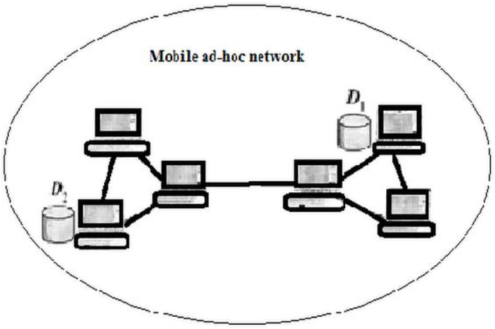


Figure 1 shows mobile ad-hoc network

In MANET [Fig1], as nodes move randomly in a MANET, Frequent Network partitions occur, causes data to be inaccessible to few nodes. So data accessibility is an important measure for performance of MANET. And even disconnections occur frequently. If a network is partitioned due to the mobility of mobile nodes, nodes in one of the divided networks will not be able to access data held by mobile nodes of the other network.

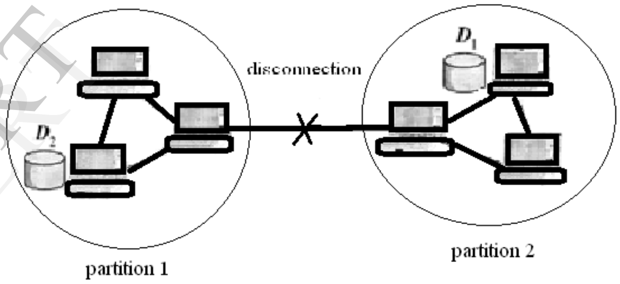


Figure 2 network partition and disconnection

Data accessibility in ad hoc networks is lower than that fixed networks. In Fig 2, if the link between two mobile nodes is disconnected at the central part, the mobile nodes on the left-hand side and those on the right-hand side cannot access data items *D*1 and *D*2, respectively.

A node may act selfish by using its resources for its own use as each node in a MANET has limited resources. A node would like to enjoy the resources provided by the other nodes in the network; however it will not make its own resource accessible by others. The problem in MANET may be as some selfish nodes may not transmit data to others to conserve their own resource constraints. Replica allocation is also important, ever since the main goal of using a MANET is to provide data s accessibility to users .The selfishness alarm occur due to network disconnections in the network. Selfish Replica allocation is referred to as a selfish node which will not share its own memory space to store replica for other nodes to access the data easily [1]. Data are replicated to the non selfish nodes, other than the original nodes, to improve data accessibility to adjust with frequently occurring network partitions.

In this paper we concentrate on reducing the overall selfishness effectively with our detection methods. And we also evaluate the accessibility of data for replica allocation methods under consideration.

2. RELATED WORK

MANET can be considered to be of two categories i.e. closed and open [2][3][4].

- **Closed**- in this all nodes voluntarily participate in and organize the network.
- **Open**-individual nodes may have different objectives. In this case, some nodes can be selfish to preserve their own resources.

A number of techniques have been proposed to handle selfishness behavior from the network perspective. Techniques handling selfish nodes can be classified into three categories:

1. reputation-based,
2. credit-payment, and
3. Game theory-based techniques.

In reputation-based techniques, each node observes the behaviors of others nodes and uses the acquired information for routing [5], [6], [7].

In credit-payment techniques, each node gives a credit to other nodes, as data forwarding [8], [9]. The acquired credit is then used to send data to others.

The game theory-based techniques assume that all rational nodes can determine their own optimal strategies to maximize their profit [10], [11]. The game theory-based techniques will find the Nash Equilibrium point [12] to maximize system performance. All these techniques focused on packet forwarding. In contrast, this paper focuses on the data accessibility and selfishness in replica allocation. The work [13] introduced several trust models and trust management schemes in a MANET that can help mitigate selfishness in a MANET. Although the work introduces several schemes for the detection of selfish nodes, the work also focuses on the selfish behavior from the network perspective, such as dropping or refusing to forward packets.

Some effective replica allocation techniques are suggested [1], including static access frequency, dynamic access frequency and neighborhood (DAFN), and dynamic connectivity-based grouping. It has been reported that DCG provides the highest data accessibility, while SAF incurs the lowest traffic, of the three techniques. As DCG performs best in terms of data accessibility, it causes the worst network traffic. And also DCG does not consider selfish nodes in a MANET. The work [14] proposes data replication techniques that address both query delay and data accessibility in a MANET. The work [15] introduces the cooperative caching-based data access methods, including CachePath, CacheData, and Hybrid.

The work [16] proposes Conquer, a broker-based economic incentive model for mobile peer-to-peer networks. Although the work [17] considers free riders to host data in mobile peer-to-peer networks, it assumes that all peers are trusted and they do not cheat.. We focus on the misbehavior of nodes. The work [18] introduced non-cooperative behaviors in a MANET. The assumption of the work is that each node in a MANET is greedy and self-interested. In the research field of distributed databases, some strategies for handling selfish behavior have been proposed [19], [20]. However, these works cannot be directly applied to a MANET, since they did not consider the constraints of a MANET such as the bandwidth limitation for the detection of selfish nodes and system failures due to frequent node disconnections

3. SYSTEM MODEL

We assume that each node has limited local memory space and acts as a data provider of several data items and a data consumer. Each node holds replicas of data items, and maintains the replicas in local memory space. The replicas are relocated in a specific period. There are m nodes, N1,N2, . . .,Nm..

The following assumptions are made

1. Each node in a MANET has a unique identifier.
2. All nodes that are placed in a MANET are denoted by N = (N1, N2, . . .Nm) where m is the total number of nodes and the set of all data items is denoted by D =(D1,D2, . . .,Dn), where n is the total number of data items.
3. Each node Ni (1 < i< m) has limited memory space for replica and original data items. The size of the memory space is Si. Each node can hold only C, where (1 < C < n), replica in its memory space.
4. Each node Ni (1< i< m) has its own access frequency to data item Dj D (1< j< n), AFi .The access frequency does not change.

We define three types of behavioral states for nodes from the viewpoint of selfish replica allocation.

1. Non selfish nodes**:** The nodes hold replicas allocated by other nodes within the limits of their memory space.

2. Fully selfish nodes**:** The nodes do not hold replicas of other nodes which are allocated, but for their accessibility, allocate replicas to other nodes.
3. Partially selfish nodes: The nodes use their memory space for allocated replicas by other nodes partially. Their memory space may be divided logically into two parts: selfish and public area.

4. STRATEGY PROPOSED

We propose a selfish node detection algorithm and replica allocation techniques to handle the selfish replica allocation properly. Each node in a network calculates credit risk value on other nodes connected individually to ensure the degree of selfishness. The selfish node is detected by the selfish replica allocation. This is based on the perception of a self-centered friendship tree (SCF-tree) and its distinction to attain high data accessibility in the presence of selfish nodes. The technical contributions of this paper can be summarized as follows

1. Recognizing the selfishness in allocation of replica: We see a selfish node in a MANET from the perception of data replication, and identify that selfish replica allocation may lead to reduce data accessibility in an ad-hoc network.
2. Detecting the fully or the partially selfish nodes effectively: We formulate a selfish node detection algorithm that will compute the degree of selfishness.
3. Allocating replica effectively: We define a set of replica allocation methods that use the theory of self-centered friendship tree to reduce communication rate and achieve high data accessibility.
4. Evaluation: we evaluate the overall selfishness and data accessibility.

4.1 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 Node specific features can be used to represent the number of shared items & shared memory space for the node. The formula for finding the credit risk is

$$nCR_i = \frac{P_k^i}{\alpha * ss_k^i / s_i + (1 - \alpha) * ND_k^i / N_i}$$

4.2 SCF-tree construction

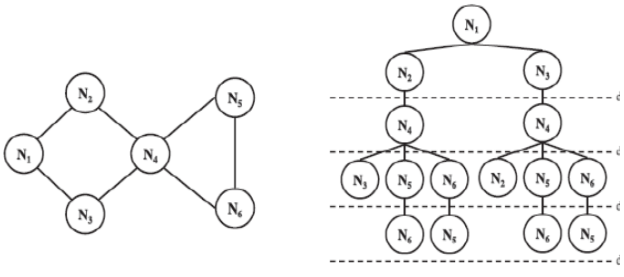
The concept of SCF-tree is based on replica allocation methods which are motivated by human friendship management in the real world [Fig 3], where each person finds his own friends forming a Group and manages friendship by themselves. The main purpose of the new replica allocation methods is to decrease traffic overhead, by achieving good accessibility of data items. Before constructing the SCF-tree, every node makes its own part of topology graph which is a component of the graph G. Topological graph consists of a finite set of the nodes connected to initial node and a finite set of the links . Algorithm 2 describes how to construct the SCF-tree.

Algorithm 1: For detecting selfish nodes

```
Selfish node detection()
{
  for (each connected node )
  {
    if (CR of initial node < threshold)
      connected node is marked as non-selfish;
    else
      connected node  is marked as selfish;
  }
  wait till replica allocation is done;
  for (each connected node )
  {
    if (initial node has allocated replica to connected node)
    {
      Number of connected node's shared data items=the number of allocated replica;
      Number of connected node's shared memory space=the total size of allocated replica;
    }
    else
    {
      Number of connected node's shared  data items=1;
      Number of connected node's shared memory space= the size of a data item;
} } }
```

Each node has a parameter d, the depth of SCF-tree. When initial nodes construct its own SCF-tree, initially first node appends the nodes that are associated to that first node by one hop to initial node’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.

Figure 3 shows the network topology representing SCF tree.



(a) Sample topology G

(b) SCF-tree of N_1

Figure 3 example showing SCF tree

Algorithm 2: SCF tree construction

```
Initial node  makes SCF-tree with a parameter,
depth d
constructScfTree()
{
 append initial node to SCF-tree as the root
node;
check Childnodes of initial node;
 return SCF-tree;
}
 Procedure checkChildnodes
{
for (each node  € set of adjacent nodes to child
nodes)
{
 if (distance between node and the root > d )
continue;
else if (first node is an ancestor of next node
in  SCFtree)
continue;
else
{
append  first node to  SCFtree as a child of
next node;
checkChildnodes(first node);
}
}
}
```

4.3 Allocating replica

The objective of the SCF-tree based replica allocation technique is to achieve good data accessibility with low communication cost in the presence of selfish nodes. Since our replica allocation technique appropriately handles the (partially) selfish nodes, the technique is expected to achieve the objective and each node processes the following procedures

1. Each node allocates replica at its discretion based on each nodes credit risk.

2. When every node receives a request for Allocation of replica from some node during every relocation period, it determines if requests are to be accepted or not.

3. The request is accepted then; each node maintains its memory space based on the credit risk. The highest credit risk held by the node which allocated replica will be replaced with new replica which is requested currently.

Alternative replica allocation techniques can be developed based on the SCF-tree structure. Thus, we propose a following set of replica allocation techniques:

SCF-tree-based replica allocation (SCF): This technique is described in Algorithm 4 and serves as a basic SCF-tree based technique.

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. This policy makes more frequently accessed data items reside on less selfish nodes.

SCF-tree based replica allocation with closer node (SCF-CN): This technique allocates more replicas to the closer nodes in the SCF-tree. That is, more replicas are allocated to the node with lower depth within the SCF-tree of Each Node.

Extended SCF-tree based replica allocation (eSCF): This technique is based on an extended SCF-tree (eSCF-tree). In this technique, all nodes in the network build its own SCF tree. Consequently, eSCF-tree contains selfish nodes with nonselfish nodes. Initial node marks the selfish nodes which are found within its eSCFtree and replicas are allocated to the nonselfish nodes in its eSCF-tree initially.

5. SIMULATION

In the simulation, the number of mobile nodes is set to 30. Each node has its local memory space and moves with a velocity from 0 ~1 (m/s) over 50(m)- 50 (m) flatland. The radio communication range of each node is a circle with a radius of 1 ~19 (m). We suppose that there are 30 individual pieces of data, each of the same size. In the network, node Ni (1 <=i <= 40) holds data Di as the original.. The default relocation period is set to

256 units of simulation time which we vary from 64 to 8,192 units of simulation time.

We evaluate our strategy using the following two Performance metrics:

1. 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 network system

2. Data accessibility: This is the ratio of the number of successful data requests to the total number of data requests.

5.1 Overall selfishness

The expected and connected nodes are only involved in a true selfishness alarm, whereas the expected but disconnected nodes in query processing may lead to a false alarm. Therefore, we plot two additional methods, DCG (selfishness only) and DCG+ (selfishness only) in Figure. 4. The overall selfishness alarm of DCG (selfishness only) and DCG+(selfishness only) is obtained by calculating data requests that will not be served by the expected connected nodes while query is processed, i.e., excluding false alarms caused by disconnections.

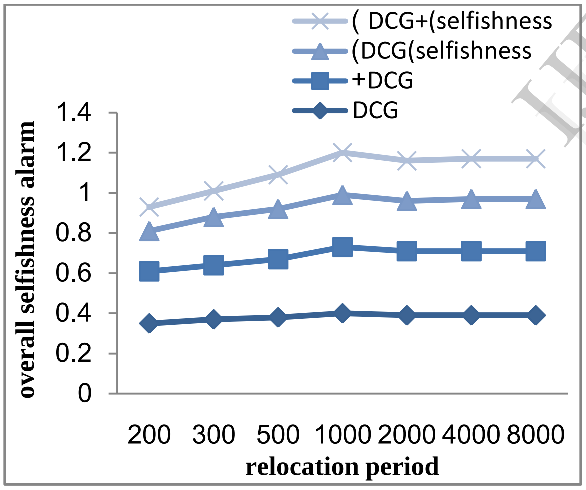


Figure 4 overall selfishness with varying relocation period

Fig 4 presents the overall selfishness alarm with varying relocation period. As expected, the DCG+ technique significantly reduces the selfishness alarms in all cases. This can be explained as follows: fewer selfish nodes become 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. Consequently, more expected nodes serve queries in DCG+ than in DCG.

As expected, the overall selfishness alarm of DCG (selfishness only) and DCG+(selfishness only) is less than that of DCG and DCG+, respectively. We see that, on average, about 62 and 56 percent of the overall selfishness alarm with DCG and DCG+ are caused by node selfishness, not disconnections.

5.2 Data accessibility

We evaluate the data accessibility of replica allocation methods under consideration. We expect that our techniques perform significantly better than other techniques in the presence of selfish nodes. In all cases, our techniques outperform SAF, DCG, and DCG+ considerably, since our techniques can detect and handle selfish nodes in replica allocation effectively and efficiently. Among our techniques, the eSCF technique shows a slightly poorer performance. Our initial intuition was that, data accessibility is stable with relocation periods. This is confirmed by the results in Figure 5a. and 5b shows that data accessibility is proportional to the size of memory space, as expected. The performance of our techniques improves faster than do others, since our techniques fully utilize the memory space of nodes. The profit of DCG technique is considerably hampered by selfish nodes, whereas the SAF technique is insensitive at all.

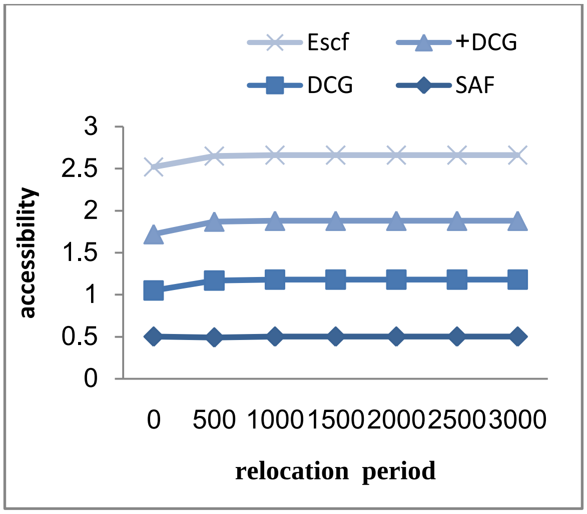


Figure 5 data accessibility with varying relocation period.

6 CONCLUSION

The selfishness alarm may also occur due to network disconnections, i.e., false alarm. After comparing with various replica allocation methods, the overall selfishness alarm of DCG (selfishness only) and DCG+ (selfishness only) is less than that of DCG and DCG+, respectively. The profit of DCG technique is considerably hampered by selfish nodes, whereas the SAF technique is insensitive at all. Here we evaluate overall selfishness and data accessibility with relocation period which keeps varying. This reduces network degradation.

REFERENCE

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

[2] 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

[3] 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.

[4] Y. Yoo and D.P. Agrawal, "

[5] pp. 1510-1515, 2003. Why Does It Pay to be Selfish in a MANET," IEEE Wireless Comm., vol. 13, no. 6, pp. 87-97, Dec. 2006.

[6] Y. Liu and Y. Yang, "Reputation Propagation and Agreement in Mobile Ad-Hoc Networks, Proc. IEEE Wireless Comm. And Networking Conf.,

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

[8] 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.

[9] 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

[10] W. Wang, X.-Y. Li, and Y. Wang, "Truthful Multicast Routing in Selfish Wireless Networks," Proc. ACM MobiCom, pp. 245-259, 2004.

[11] D. Hales, "From Selfish Nodes to Cooperative Networks - Emergent Link-Based Incentives in Peer-to-Peer Networks," Proc. IEEE Int'l Conf. Peer-to-Peer Computing, pp. 151-158, 2004.

[12] V. Srinivasan, P. Nuggehalli, C. Chiasserini, and R. Rao, "Cooperation in Wireless Ad Hoc Networks," Proc. IEEE INFOCOM, pp. 808-817, 2003.

[13] M.J. Osborne, An Introduction to Game Theory. Oxford Univ., 2003.

[14] H. Li and M. Singhal, "Trust Management in Distributed Systems," Computer, vol. 40, no. 2, pp. 45-53, Feb. 2007.

[15] 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

[16] G. Cao, L. Yin, and C.R. Das, "Cooperative Cache-Based Data Access in Ad Hoc Networks," Computer, vol. 37, no. 2, pp. 32-39, Feb. 2004.

[17] 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.

[18] E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati, "Managing and Sharing Servents' Reputations in P2P Systems," IEEE Trans. Knowledge and Data Eng., vol. 15, no. 4, pp. 840-854, July/Aug. 2003.

[19] 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.

[20] 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.

[21] 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