

Replica Allocation In Mobile Adhoc Network For Improving Data Accessibility Using SCF-Tree

K.P.Shanmuga Priya¹ V.Seethalakshmi²

Sri Shakthi Institute of Engineering and Technology, Coimbatore, Tamilnadu

Abstract: In a mobile ad hoc network, the mobility and resource constraints of mobile nodes may lead to network partitioning or performance degradation. 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. In this paper, the impact of selfish nodes in a mobile ad hoc network from the perspective of replica allocation is examined. A selfish node detection algorithm was developed that considers partial selfishness and novel replica allocation techniques to properly cope with selfish replica allocation. The conducted simulations demonstrate the proposed approach outperforms traditional cooperative replica allocation techniques in terms of data accessibility, communication cost and average query delay.

Keywords: Manet, Selfish nodes, data accessibility, communication cost and query delay.

I. Introduction

Mobile ad hoc networks (MANETs) have attracted a lot of attention due to the popularity of mobile devices and the advances in wireless communication technologies. A MANET is a peer-to-peer multihop mobile wireless network that has neither a fixed infrastructure nor a central server. Each node in a MANET acts as a router, and communicates with each other. A large variety of MANET applications have been developed. For example, a MANET can be used in special situations, where installing infrastructure may be difficult, or even infeasible, such as battlefield or a disaster area. A mobile peer-to-peer file sharing system is another interesting MANET application [4].

Network partitions can occur frequently, since nodes move freely in a MANET, causing some data to be often inaccessible to some of the nodes. Hence data accessibility is often an important performance metric in a MANET. Data are usually replicated at nodes, other than the original owners, to increase data network partitions. A considerable amount of research has recently been proposed for replica allocation in a MANET [6][7].

In general replication can simultaneously improve data accessibility and reduce query delay, i.e., query Response time, if the mobile nodes in a MANET together have sufficient memory space to hold both all the replicas and the original data. For example, the response time of a query can be substantially reduced, if the query accesses a data item that has a locally stored replica. However, there is often a trade-off between data accessibility and query delay, since most nodes in a MANET have only limited memory space. For example, a node may hold a part of the frequently accessed data items locally to reduce its own query delay. However, if there is only limited memory space and many of the nodes hold the same replica locally, then some data items would be replaced and missing. Thus, the overall data accessibility would be decreased. Hence, to maximize data accessibility, a node should not hold the same replica that is also held by many other nodes. However, this will increase its own query delay. The nodes can be divided into three types. They are, Type-1 node: The nodes are non-selfish nodes. The nodes hold replicas allocated by other nodes within the limits of their memory space.

- Type-2 node: The nodes are fully selfish nodes. The nodes do not hold replicas allocated by other nodes, but allocate replicas to other nodes for their accessibility.
- Type-3 node: The nodes are partially selfish nodes. The nodes use their memory space partially for allocated replicas by other nodes. Their memory space may be divided logically into two parts: selfish and public area. These nodes allocate replicas to other nodes for their accessibility.

II. Overview of Existing Methods

This section deals with the replica allocation methods in MANET environment having selfish nodes which influence the performance of data accessibility.

2.1 Static Access Frequency (SAF) Method

In SAF method, the nodes allocate replica of data items according to the access frequencies of that data items. 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 them. 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.

2.2 Dynamic Connectivity Based Grouping Method (DCG)

The DCG method shares replicas in larger groups of mobile hosts than DAFN. At every relocation period, each mobile host broadcasts its host identifier. After all mobile hosts complete the broadcasts; every host knows the connected mobile hosts and the network topology from the received host identifiers. In each set of mobile hosts connected to each other, the mobile host with the lowest host identifier suffix executes an algorithm to find bi-connected components with the network topology known by received messages. Even if a mobile host belongs to more than one bi-connected component, it can only belong to one group in which the corresponding bi-connected component was found first. By grouping mobile hosts as bi-connected components, the group is not divided even if one mobile host disappears from the network or one link is disconnected in the groups. Thus, it is assumed that the group has high stability affected, unless they were in sleep mode and also if the selected routes are via specific host, the battery of this host will be exhausted quickly.

2.3 Dynamic Connectivity-Based Grouping with Detection (DCG +)

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. 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. 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 other approaches to determine selfishness, including the agreement of 1) at least one and 2) the majority of nodes.

2.4 Dynamic Access Frequency and Neighbourhood (DAFN)

The algorithm of this method is as follows:

- 1) At a relocation period, each mobile host broadcasts its host identifier and information on access frequencies to data items. After all mobile hosts complete the broadcasts, from the received host identifiers, every host shall know its connected mobile hosts.
- 2) Each mobile host preliminary determines the allocation of replicas based on the SAF method.
- 3) In each set of mobile hosts which are connected to each other, the following procedure is repeated in the order of the breadth first search from the mobile host with the lowest suffix (i) of host identifier (Mi). When there is duplication of a data item (original/replica) between two neighbouring mobile hosts, and if one of them is the original, the host which holds the replica changes it to another replica. If both of them are replicas, the host whose access frequency value to the data item is lower than the other one changes the replica to another replica. When changing the replica, among data items whose replicas are not allocated at either of the two hosts, a new data item replicated is selected where the access frequency value to this item is the highest among the possible items. This eliminates replica duplication among neighboring hosts. The above procedure is executed every relocation period. Overhead and traffic is much higher than SAF.

III. Problem Definition

The problem of selfishness in the context of replica allocation in a MANET, i.e., a selfish node may not share its own memory space to store replica for the benefit of other nodes. A node may act selfishly, i.e., using its limited resource only for its own benefit, since each node in a MANET has resource constraints, such as battery and storage limitations. 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 behavior can potentially lead to a wide range of problems for a MANET. Existing research on selfish behaviors in a MANET mostly focus on network issues. For example, selfish nodes may not transmit data to others to conserve their own batteries. Although network issues are important in a MANET, replica allocation is also crucial, since the ultimate goal of using a MANET is to provide data services to users [10].

IV. Proposed Method (SCF-Tree)

Novel replica allocation techniques for selfish node detection and elimination were devised. They are based on the concept of a self-centered friendship tree (SCF-tree) and its variation to achieve high data accessibility with low communication cost in the presence of selfish nodes. The SCF-tree is inspired by our human friendship management in the real world. Main aim is to reduce the communication cost, while still achieving good data accessibility. The technical contributions of this paper can be summarized as follows

- i) **Recognizing the selfish replica allocation problem:** selfish nodes in a MANET from the perspective of data replication are viewed, and recognize that selfish replica allocation can lead to degraded data accessibility in MANET.
- ii) **Detecting the fully or the partially selfish nodes effectively:** A selfish node detection method is devised to measure the degree of selfishness.
- iii) **Allocating replica effectively:** Set of replica allocation techniques are proposed, that use the self-centered friendship tree to reduce communication cost, while achieving good data accessibility.
- iv) **Verifying the proposed strategy:** The simulation results verify the efficacy of the proposed strategy.

4.1 Proposed Strategy

The paper consists of three parts: 1) detecting selfish nodes, 2) building the SCF- Tree, and 3) allocating replica. At a specific period, or relocation period, each node executes the following procedures:

- Each node detects the selfish nodes based on credit risk scores.
- Each node makes its own topology graph and builds its own SCF-tree by excluding selfish nodes.
- Based on SCF-tree, each node allocates replica in a fully distributed manner.

4.1.1 Detecting Selfish Node

The notion of credit risk can be described by the following equation:

$$\text{Credit Risk} = \frac{\text{expected risk}}{\text{expected value}}$$

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. First, selfish features may lead to the selfish replica allocation problem were both expected value and expected risk are determined.

4.1.2 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. The decision is solely at his/her discretion. The main objective of our novel replica allocation techniques is to reduce traffic overhead, while achieving high data accessibility. If the novel replica allocation techniques can allocate replica without discussion with other nodes, as in a human friendship management, traffic overhead will decrease.

4.1.3 Allocating Replica

After building the SCF-tree, 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. Replica can be allocated at each node in descending order of its own access frequency. This is quite different from existing group-based replica allocation techniques (e.g., DCG) where replicas are allocated based on the access frequency of group members. Each node N_i 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.

V. Results and Discussion

In the simulation, the number of mobile nodes is set to 40. Each node has its local memory space and moves with a velocity from 0~1 (m/s) over 50(m) 50 (m) flatland. The movement pattern of nodes follows the random waypoint model, where each node remains stationary for a pause time and then it selects a random destination and moves to the destination. After reaching the destination, it again stops for a pause time and repeats this behaviour. The radio communication range of each node is a circle with a radius of 1~ 19 (m). We suppose that there are 40 individual pieces of data, each of the same size. In the network, node N_i ($1 \leq i \leq 40$) holds data D_i as the original. The data access frequency is assumed to follow Zipf distribution. The default relocation period is set to 256 units of simulation time which we vary from 64 to 8,192 units of simulation time. Table 1 describes the simulation parameters.

5.1 Simulation Parameters

Table 1: Simulation Environment

| Parameter (unit) | Value (default) |
|-----------------------------------|-----------------|
| Number of nodes | 40 |
| Number of data items | 40 |
| Radius of communication range (m) | 1~19 (7) |
| Size of the network (m) | 50*50 |
| Size of memory space (data items) | 2~40(10) |
| Percentage of selfish nodes | 0~100(70) |
| Maximum velocity of a nodes | 1 |
| Relocation period | 64~8192 (256) |

5.2 Metrics used for Evaluation

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.

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

Average query delay: This is the number of hops from a requester node to the nearest node with the requested data item. If the requested data item is in the local memory of a requester, the query delay is 0. We only consider successful queries, i.e., it is the total delay of successful requests divided by the total number of successful requests.

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

5.3 Simulation Results

The Communication cost (fig 5.1) of a network using SCF-tree is minimum compared to normal replica (DCG). Where as Data Accessibility, Overall Selfishness, Average Query delay of a network is maximum (fig 5.2, 5.3, 5.4)

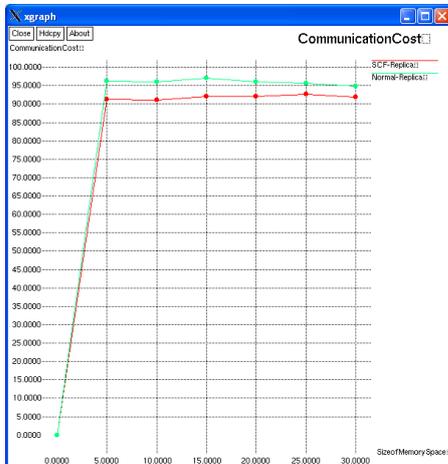


Fig 5.1 Comparison of Communication cost Vs Size of memory space using 40 nodes

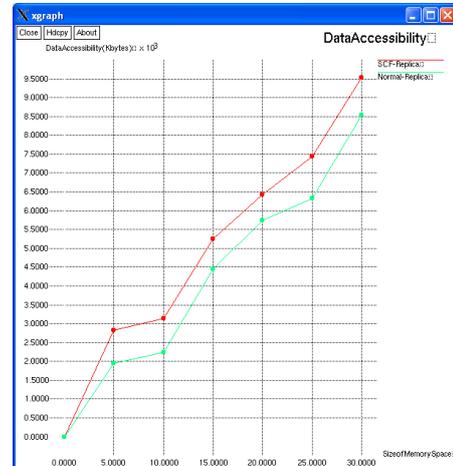


Fig 5.2 Comparison of Data Accessibility Vs Size of memory space using 40 nodes

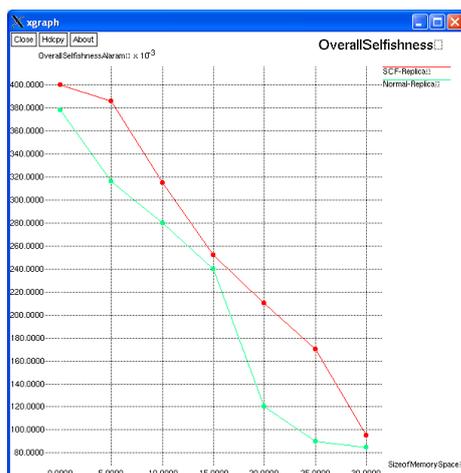


Fig 5.3 Comparison of Overall Selfishness Vs Size of memory space using 40 nodes

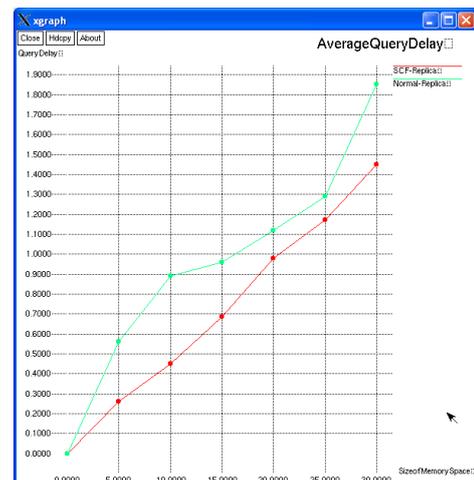


Fig 5.4 Comparison of Average Query delay Vs Size of memory space using 40 nodes

VI. Conclusion

In contrast to the network viewpoint, the problem of selfish nodes from the replica allocation perspective is addressed. This problem is known as selfish replica allocation. The work was motivated by the fact that a selfish replica allocation could lead to overall poor data accessibility in a MANET. A selfish node detection method and novel replica allocation techniques to handle the selfish replica allocation appropriately have been proposed. 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 notion of credit risk from economics to detect selfish nodes is applied. 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, novel replica allocation techniques also proposed. Extensive simulation shows that the proposed strategies outperform existing representative cooperative replica allocation techniques in terms of data accessibility, communication cost, and query delay.

VII. Future work

This paper proposes further research into more replica allocation technique that can improve the performance of MANETs. Currently working on the impact of different mobility patterns, and plan to identify and handle false alarms in selfish replica allocation.

References

- [1] 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.
- [2] 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
- [3] B.-G. Chun, K. Chaudhuri, H. Wee, M. Barreno, C.H. Papadimitriou, and J. Kubiawicz, "Selfish Caching in Distributed Systems: A Game-Theoretic Analysis," Proc. ACM Symp. Principles of Distributed Computing, pp. 21-30, 2004
- [4] G. Ding and B. Bhargava, "Peer-to-Peer File-Sharing over Mobile Ad Hoc Networks," Proc. IEEE Ann Conf. Pervasive Computing and Comm. Workshops, pp. 104-108, 2004
- [5] M. Feldman and J. Chuang, "Overcoming Free-Riding Behavior in Peer-to-Peer Systems," SIGecom Exchanges, vol. 5, no. 4, pp. 41-50, 2005.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] M.J. Osborne, An Introduction to Game Theory. Oxford Univ., 2003.
- [10] 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