Global Journal of Computer Science and Technology
Network, Web & Security
Volume 13 Issue 6 Version 1.0 Year 2013
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Improving the Performance of Mobile Ad Hoc Network using a Combined Credit Risk and Collaborative Watchdog Method

By S. J. K. Jagadeesh Kumar, R. Saraswathi & R. Raja
Sri Krishna College of Technology, Combat Tamil Nadu, India

Abstract - In mobile ad hoc networks, nodes can move freely and link/node failures occur frequently. This leads to frequent network partitions, which may significantly degrade the performance of data access in ad hoc networks. When the network partition occurs, mobile nodes in one network are not able to access data hosted by nodes in other networks. In mobile ad hoc network, 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 work, the impact of selfish nodes in a mobile ad hoc network from the perspective of replica allocation is examined. We term this selfish replica allocation. A combined credit risk method & collaborative watchdog is proposed to detect the selfish node and also apply the SCF tree based replica allocation method to handle the selfish replica allocation appropriately. The proposed method improves the data accessibility, reduces communication cost and average query delay and also to reduce the detection time and to improve the accuracy of watchdogs in the collaborative method.

Keywords : *mobile ad hoc network, collaborative watchdog, selfish replica allocation, SCF tree, AAS, DCG.*

GJCST-E Classification : *C.2.5*

Strictly as per the compliance and regulations of:

Improving the Performance of Mobile Ad Hoc Network using a Combined Credit Risk and Collaborative Watchdog Method

S. J. K. Jagadeesh Kumar [α], R. Saraswathi [σ] & R. Raja [ρ]

Abstract - In mobile ad hoc networks, nodes can move freely and link/node failures occur frequently. This leads to frequent network partitions, which may significantly degrade the performance of data access in ad hoc networks. When the network partition occurs, mobile nodes in one network are not able to access data hosted by nodes in other networks. In mobile ad hoc network, 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 work, the impact of selfish nodes in a mobile ad hoc network from the perspective of replica allocation is examined. We term this selfish replica allocation. A combined credit risk method & collaborative watchdog is proposed to detect the selfish node and also apply the SCF tree based replica allocation method to handle the selfish replica allocation appropriately. The proposed method improves the data accessibility, reduces communication cost and average query delay and also to reduce the detection time and to improve the accuracy of watchdogs in the collaborative method.

Indexterms : mobile ad hoc network, collaborative watchdog, selfish replica allocation, SCF tree, AAS, DCG.

I. Introduction

a) Mobile Ad Hoc Network

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 [4][6].A "mobile ad hoc network" (MANET) is an autonomous system of mobile routers (and associated hosts) connected by wireless links - the union of which forms an arbitrary graph. The routers are free to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or may be connected to the larger Internet. The strength of the connection can change rapidly in time or even disappear completely. Nodes can appear, disappear and re-appear as the time goes on and all the time the network connections should work between the node that are part of it. As one can easily imagine, the situation in ad hoc networks with respect to ensuring connectivity and robustness is much more demanding than in the wired case.

Mobile Ad hoc Network (MANET) is a collection of independent mobile nodes that can communicate to each other via radio waves. The mobile nodes that are in radio range of each other can directly communicate, whereas others need the aid of intermediate nodes to route their packets. These networks are fully distributed, and can work at any place without the help of any infrastructure. A MANET can be used in many areas such as Military applications, Disaster relief operations, and Robot data acquisition system. A mobile P2P file sharing system is another interesting applications [6][27].The characteristics of these networks are summarized as follows:

1. Communication via wireless means.
2. Nodes can perform the roles of both hosts and routers.
3. No centralized controller and infrastructure.
4. Dynamic network topology. Frequent routing updates
5. Autonomous, no infrastructure needed
6. Can be set up anywhere.
7. Energy constraints and Limited security

b) Selfish Node in Data Replication

In Mobile Ad Hoc Network, the mobile nodes can move freely, so that the network partitions can occur frequently. Hence the data accessibility is very lower in the network and the query delay was increased. By reducing the query delay in the network, replicate the data in some other nodes. In the data replication, the data accessibility also increased. So there is a trade off between query delay and the data accessibility in the network [11]. In mobile ad hoc network, some of the nodes can't forward packets to the other nodes. These nodes are called selfish or malicious nodes. These selfish nodes cannot allocate replica to other nodes and it does not share the memory space for the other nodes. These selfish nodes can lead to a wide range of problems in the networks. To solve such problem, in this paper, we propose a selfish node detection algorithm and reduce the detection time of the node using the collaborative watchdog method.

Author α σ : Department of Computer Science and Engineering, Sri Krishna College of Technology, Combat Tamil Nadu, India.
Author ρ : King College of Technology, Namakkal, India.
E-mail : surswathi@gmail.com

c) Selfish Node in Replica Allocation

In this fig. 1 illustrates an existing replica allocation scheme called, Dynamic Connectivity based Grouping (DCG) [5] [6]. In this method, the access frequency of each data item and the whole network topology are taken into account. In DCG, the nodes N, N_2, . . ., N_6 maintain their memory space M_1,M_2, . . .,M_6, respectively. In Fig. 1, a straight line denotes a wireless link between the nodes and the first data item in each node is the original data and the remaining data items in each node is the replicated data items. Fig. 1, DCG seeks to minimize the duplication of data items in a group to achieve high data accessibility. In this diagram, the node N_3 behaves "selfishly" by maintaining M_3, instead of M_3'. In the original case, D_3, D_9, and D_2 were allocated to N3. However, due to the selfish behavior, D_3, D_5, and D_2, the top three most locally frequently accessed items, are instead maintained in local storage. Thus, other nodes in the same group, i.e., N_1, N_2, and N_4, are no longer able to access D_9. This showcases degraded data accessibility. The proposed system has some advantages such as -

1. Easily detects the selfish node without collision.
2. SCF-tree based replica allocation is performed in a fully distributed manner.
3. Cooperative replica allocation techniques were performed.

In this paper, the problem of selfish node was addressed in the replica allocation. That is the selfish node may not share the memory space to the other nodes. This problem is called as selfish replica allocation. To propose a self centered friendship tree method to handle the selfish replica allocation problem is very efficient manner. This proposed method improves the communication cost, data accessibility and the query delay. The technical contributions of this paper can be summarized as follows:

1. Recognize the selfish replica allocation problem in the mobile ad hoc network.
2. Detect whether the node is partially selfish node or fully selfish node.
3. Apply the self centered friendship tree algorithm for replica allocation.
4. Verify the proposed strategy.

The Section 2 describes the related work .The overview of system model is described in Section 3. The proposed detection method and the replica allocation techniques are presented in Section 4. The simulation scenario is presented in section 5. The performance evaluation is presented in Section 6 and the conclusion of the paper is presented in Sections 7.

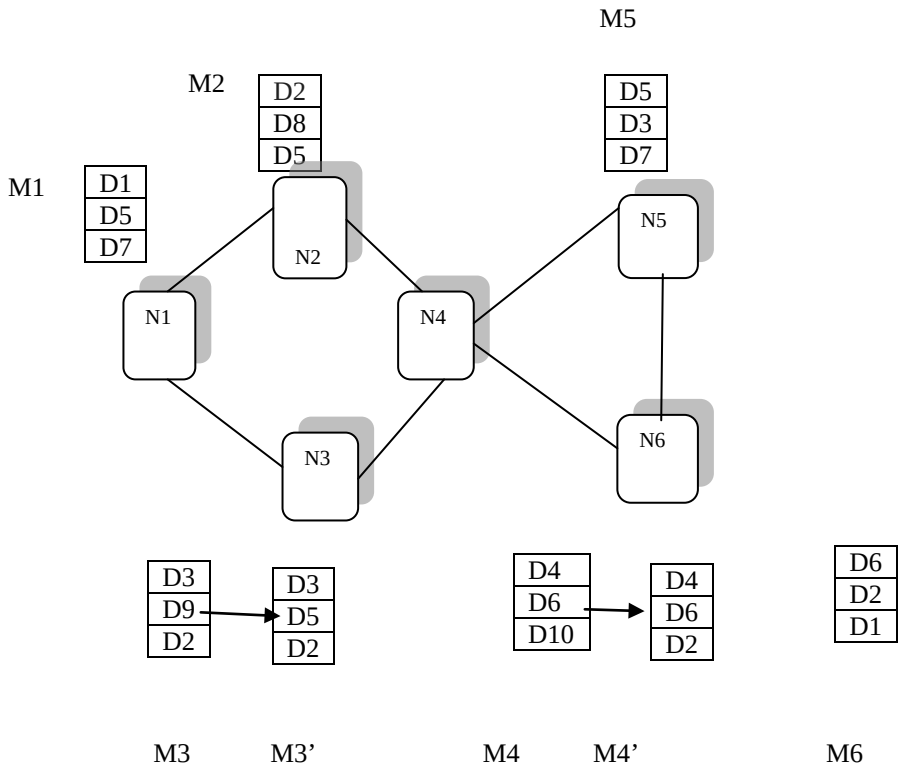

Figure 1 : Selfish Replica Allocation

II. Related Work

a) Selfish Node Detection

Multi-hop communication in mobile ad hoc networks (MANETs) requires collaboration among nodes, which forward packets for one another. In MANET, some of the nodes may refuse to forward packets in order to conserve their limited resources

resulting in traffic disruption. Nodes exhibiting such behaviour are termed selfish. Selfishness is usually passive behaviour. Selfish and malicious behaviours are usually distinguished based on the node's intent.

Various techniques have been proposed to handle the problem of selfish behaviour from the network perspective. As described, the techniques handling selfish nodes can be classified into three categories: reputation-based, credit-payment, and game theory-based techniques [12].

In reputation-based a large number of schemes belong to the first category, with varying implementations [2]. One advantage of such schemes could be their quick convergence in detecting node misbehaviour, especially in a large ad hoc network, due to increased information regarding a particular node's behaviour. However, this approach has two potential drawbacks: they often assume that nodes that send reputation information about their peers are themselves trustworthy; and they are subject to collusion among nodes that misreport reputation information. In credit-payment techniques, each node gives a credit to others, as a reward for data forwarding. The acquired credit is then used to send data to others [1]. The game theory-based techniques assume that all rational nodes can determine their own optimal strategies to maximize their profit. The game theory-based techniques want to find the Nash Equilibrium point to maximize system performance [8]. The AAS scheme is a network-layer technique to detect the selfish nodes and to mitigate their effects [7]. In [14] Sergio Marti et al proposed to mitigate the routing misbehaviour in MANET using watchdog and bathwater method. The watchdog method is used to identifies the misbehaving nodes in the network and the bathwater combines the knowledge of misbehaving nodes with link reliability data to pick the route most likely to be reliable In [15] Miranda et al proposed a novel algorithm that enhances the load balancing while banning the selfish node form the MANET. In [16] Yangchow et al proposed a secure incentive protocol to simulate the cooperation among the possible selfish nodes. It also provides highly secure incentives for the selfish nodes to cooperative in packet forwarding with low overhead. In [17], Hussein et al proposed a BAODV protocol, is an extension of the AODV protocol and this method is used to improve the performance and reliability of the wireless mobile ad hoc network in the presence of malicious or selfish node. It take the behaviour history of member nodes into account, to improve the network reliability. The BAODV model is based on the discovery of communication paths with minimal number of selfish or malicious nodes. In [18] Debut set al proposed a new intrusion detection system based on mobile agents to improve the network bandwidth consumption and reduce the computation overhead in the network.

b) Replica Allocation Methods

Some effective replica allocation techniques are suggested [5], including static access frequency, dynamic access frequency and neighborhoods (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. Although DCG performs best in terms of data accessibility, it causes the worst network traffic. Moreover, DCG does not consider selfish nodes in a MANET. The work [11] proposes data replication techniques that address the tradeoff between both query delay and data accessibility in a MANET. The work [2] introduces three cooperative caching-based data access methods, including Cache Path, Cache Data, and Hybrid. The work [9][11] introduces the non cooperative replica allocation game RAG)which provides the optimal performance of the mobile ad hoc network. The work[4] identifies the issues involved in MANET data replication and attempts to classify existing MANET data replication techniques based on the issues they address the performance of MANET replication techniques.

In [19], Datta, et al propose the Hybrid Replica Control protocol that attempts to maximize the data availability and communication overhead. In [20], Feras et al propose a Constrained Fast Spread (CFS) method to alleviate the main problems encountered in the current replication techniques and mainly concentrating on the feasibility of replicating the requested replica on each node among the network In[21],Chao-Tung et al proposed a One-way Replica Consistency Service (ORCS) for grid environment to resolve the consistency maintenance issues and also balancing the tradeoff between the improving data Access performance and replica consistency. In [22] Jean et al proposed a non cooperative behaviors of the selfish node. In [23] Show yang et al proposed a dynamic replication scheme which employs the user profile for recording user mobility schedules, access behavior and read/write patterns and actively reconfigure the replicas to adapt to changes in user locations, data request and system status. In [24] Padmanabhan et al identifies issues involved in MANET data replication and attempts to classify existing MANET data replication techniques based on the issues they address. In addition, this paper also proposes criteria for selecting appropriate data replication techniques for various application requirements. In [25] Yin et al proposed a various system settings and requirements to balance the tradeoffs between data accessibility and query delay under and also improve the system performance in MANET. Differing from all the above-mentioned replica allocation or caching techniques, we consider selfish nodes in a MANET.

III. System Model

In this paper, we assume that each node has limited memory space. Each node can hold replicas of data items and maintains the replicas in local memory space. There are m nodes, N1,N2, . . .,N_m. Constructing a model for MANET is an undirected graph G= (IN, IL) that consists of a finite set of nodes, IN, and a finite set of communication links, IL, where each element is a topple(N_j,N_k) of nodes in the network. The system environment in MANET is assumed to be the following [5][6]:

1. The mobile hosts access data items held by other mobile hosts (single or multiple hops).
2. Each mobile host creates replicas of the data and maintains the replicas in its memory.
3. Data item available if it is present locally or if it is available at one of the neighbours.
4. Each node has a unique host identifier: Mj (set of all mobile hosts M = {M1, M2,…, Mm}).
5. Each node has a unique data identifier: Dj (set of all data items D = {D1, D2,…, Dm}).
6. Assume all data items are of the same size.
7. Each host has a memory space of C data items for replicas (excluding the space for holding originals).
8. Data remains the same and does not change (simplifying assumption).
9. The access frequencies of the data item for each mobile host are known and it does not change for that node.

In fig 3a shows the communication established between the nodes. When a node N_i request data item means, it will checks its own memory space first. If the memory space is free means, it will allocate the original or replica copy of the data item to that memory. If it does not hold the original or replica of the data item, the request will be forwarded to the other node. If it is present, the node N_i receives reply from that node. Otherwise, the request will fails. In this selfish replica allocation point of view, there are three types of behavioural states for the nodes are available.

1. Type-1 node: These types of node are non selfish nodes. These nodes hold replicas allocated by other node within the limits of their memory space.
2. Type-2 node: These types of node are fully selfish nodes. These nodes do not hold replicas, but it allocates replicas to other nodes for their accessibility.
3. Type-3 node: These types of node are partially selfish nodes. These nodes use their memory space partially. Here the memory space is divided into two parts: i.e. selfish and public area. In this public area, the replica will be allocated.

Figure 3a : Simulation on System Model

IV. Proposed System

In this paper, propose 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. We applied the credit risk method to detect the selfish nodes and also propose the collaborative watchdog method to reduce the detection time of the node. Every node in a MANET calculates credit risk information on other connected nodes individually so that it is used to measure the degree of selfishness. Since the various traditional replica allocation techniques as described in the related work in section 2 was failed to consider the selfish nodes. In this paper, we propose a SCF tree based replica allocation techniques to handle the replica allocation effectively.

First we detect the selfish node by selfish replica allocation. The novel replica allocation techniques are based on the concept of a self-centered friendship tree (SCF-tree) and this method 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 [26]. In the real world, a friendship, which is a form of social bond, is made individually. For example, although A and B are friends, the friends of A are not always the same as the friends of B [26]. The main aim of SCF tree is to reduce the communication cost, while achieving good data accessibility. The technical contributions of this paper can be summarized as follows.

1. Recognizing the selfish replica allocation problem.
2. Detecting the fully or the partially selfish nodes effectively.
3. Allocating replica effectively.
4. Verifying the proposed strategy

a) Selfish Node Detection

The network is modeled as a set of N wireless mobile nodes. The credit risk for the each node can be described by the following equation:

Credit Risk = expected risk / expected value (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 also used to represent the expected value of a node. If the node N_i requests the data to the node no means, the node no share the memory space and the data items for that node N_i. So the node no is treated as a valuable node.

Then the query processing feature is calculated for the node Ni. It is defined as the ratio of N_i's data request being not served by the expected node no. Because the node no 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 §. α is the system parameter, where 0≤ α ≤1.The formula for finding the credit risk is

$$nCR_i = \frac{P_k^i}{\alpha * ss_k^i / s_i + (1-\alpha) * ND_k^i / N_i} \quad (2)$$

In the Eq . (2) where,

1. SS_i^K is the size of N_k's shared memory space.
2. ND^k_i 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. § is the threshold value of node N_i.
6. α is the system parameter ,where 0≤α ≤1.

In this fig 4.1 shows the simulation on the detection of selfish node using credit risk method.When the construction of the topology of the network, initailize the number of nodes in the network and also specify the location of the nodes in the network. After constructing nodes in the network,initialize the data set and the memory space for the node.In the initialization,first data in the node is original data and the remaing data is called replica copy of the data.

When the node request data in the another node, find what are all the nodes contains the data.If the node contains the relevant data means,specify the possible path for the node to the requested node.In these path, the shortest path of the node is selected. Based on the shared data items and shared memory space used for these selfish nodes, the degree of the selfishness is measured.And then find the credit risk for each node.

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 N_k , 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.

Figure 4.1 : Detection of Selfish Node using Credit Risk method

b) Collaborative Watchdog

i. *Identifying Selfish Contact*

In this module, detection time of sells nodes have to be reduced based on contact dissemination. If one node has previously detected as a sells 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 sells 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 modeled 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, 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. Then, when a contact occurs, c can increase by one.

ii. *Collaborative Contact*

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 can 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 sells nodes is lost or that a node temporally does not collaborate. In fig 4.2 shows the detection time of the selfish node by using combined credit risk & collaborative watchdog method.

Algorithm 1: Finding the detection time of the selfish node

1. The probability value p_{ij} is denoting the transition rate from transient state s_i to absorbing state s_j.
2. Given a state s_i =(c) the following transitions can occur:
   a. The state changes form (c) to (c+1). ie., the collaborative node may changes from NOINFO state to POSITIVE state.
   b. Calculate the transition probability

$$t_c = (\lambda p_d + \lambda p_c c)(C - c). \quad (3)$$

   c. In the equation (3), λp_d represents the probability of detection of a selfish node and $\lambda p_c c$ *is* the probability of transmission for the information of the selfish node.
   d. Finally, factor (C − c) represents the number of pending nodes.
3. Otherwise the state does not change to other state. This is the probability of no changes, and its detection time is calculated by the formula is $t_0 = 1 - t_c$..

Figure 4.2 : Simulation of the Collaborative Watchdog Method

c) Self Centered Friendship 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 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.

Before constructing the SCF-tree, each node makes its own partial topology graph G_i = in, 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 N_i € IN_i, IN_i C IN, and IL_i C IL. Since SCF tree consists of only non selfish nodes, then we need to measure the degree of selfishness by using the credit risk (nCR_i) value of each node in the network. Before constructing the SCF tree, node Ni eliminates the selfish node from in. Since Ni removes every link containing the selfish nodes and the replace with the new edge that should not contain the selfishness in data forwarding.

Based on G N_i builds its own SCF-tree, denoted as T_i^{SCF}. 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. 4.3a shows the simulation of the SCF tree.

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.

Figure 4.3a : Simulation of SCF Tree

d) 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 M_s and public area M_p.

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 M_s is not full then, allocate replica of the data to the selfish area M_s. Otherwise, allocate replica of the data to the target node.
5. If the public area of the node M_p is not full then, allocate replica of the data to the public area M_p.
6. If the node N_k requests for the allocation of D_q then if the node N_k is in SCF tree T_i^{SCF} and N_i does not hold the data Dq.
7. If the public area of the node M_p is not full then, allocate the data D_q to M_p.
8. Otherwise, if the node $_{Ni}$ holds any replica of local interest in public area $_{Mp}$ then replace the replica with D_q;
9. Check the credit risk of the node nCR^h_i is greater than nCR^k_i then replace the replica requested by the node N_h with D_q;

V. Simulation Scenario

The simulation model is similar to that employed in [5][9]. In the simulation, the number of mobile nodes is set to 50. Each node has its local memory space and moves with a velocity from 0 ~ 1 (m/s) over 50 (m) x50 (m) .The movement pattern of nodes follows the random waypoint model [5], where each node remains stationary for a pause time and then it selects a random destination and moves to the destination. The radio communication range of each node is a circle with a radius of 1 ~ 19 (m). Suppose that there are 40 individual pieces of data, each of the same size. In the network, node N_i ($1 \leq i \leq 50$) holds data D_i as the original. Table 1 describes the simulation parameters.

The default number of selfish nodes is set to be 80 percent of the entire nodes in our simulation, based on the observation of a real application [1]. We set 75 percent of selfish nodes to be type-3 (i.e., partially selfish) and the remaining to be type-2 (i.e., fully selfish). Type-3 nodes consist of three groups of equal size. Each group uses 25, 50, and 75 percent of its memory space for the selfish area. Type-2 nodes will not accept replica allocation requests from other nodes in the replica allocation phase, thus being expected to create significant selfishness alarm in query processing. Type-3 nodes will accept or reject replica allocation requests according to their local status, thereby causing some selfishness alarms in subsequent query processing. We evaluate the following four 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 system.
2. Communication cost: This is the total hop count of data transmission for selfish node detection and replica allocation/relocation, and their involved information sharing.
3. 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.
4. Data accessibility: This is the ratio of the number of successful data requests to the total number of data requests.

Table 1 : Simulation Parameters

Parameter	Value
Number of nodes	50
Number of data items	50
Radius of communication range	25
Size of the network	50X50
Size of the memory space	60
Relocation period	9000
Percentage of Selfish nodes	80 %
Maximum velocity of a node	1

VI. Performance Evaluation

This section is first devoted to evaluating the performance of our collaborative watchdog method using the combined credit risk method and the collaborative watchdog. All the model were implement and evaluated using NS2.The evaluation shows the impact of the number of nodes ranging from 0 to 100 . Three different sets of values for *pc* and *pd* were used. The first set (1, 0.8) is a full collaborative network with a high probability of detection, the second set has a reduced degree of collaboration (0.7), and finally the last set has a low probability of detection (0.3).

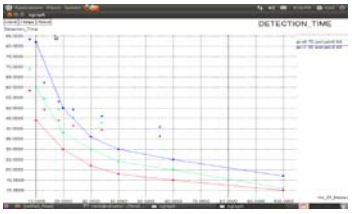

Figure 6a : Evaluation Depending on Node

In fig 6a, we observe that, in general, the greater the number of nodes, the lesser the detection time and the greater the number of messages. As expected, reduced values of collaboration and detection probabilities imply greater detection times. Figure 6b shows the influence of the number of selfish nodes S for $N = 50$. As expected, the detection time decreases when the number of selfish nodes is higher. The results confirm that the increasing the period p implies that the detection time is decreased and the overhead is reduced.

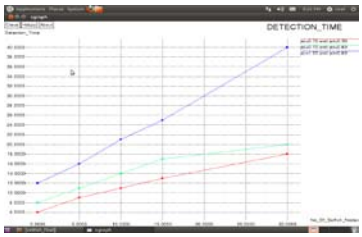

Figure 6b : Evaluation depending on selfish node

a) Communication Cost

Fig. 6c shows the communication cost of DCG, DCG$^+$ and SCF. In all cases, our techniques outperform DCG and DCG$^+$, while SCF shows the best performance in terms of communication cost and average query delay. As the communication range increases, the communication cost of all techniques increases at first, but it gets smaller from a certain point, except SAF. When the communication range is smaller than a certain point in Fig 6a, the communication cost increases as the Communication range gets larger, since the number of nodes connected to each other increases and thus the communication cost caused by replica relocation increases. Conversely, when the communication range is larger than a certain point, the number of hops among connected nodes decreases. Therefore, the communication cost caused by replica relocation decreases.

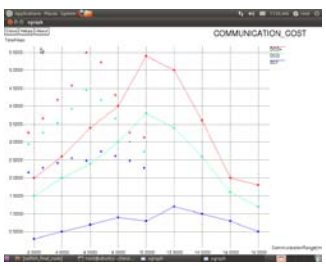

Figure 6c : Communication Cost

We see, in Fig. 6d, that the data accessibility improves with the wide range of communication, since more nodes become connected. In this figure, the performance of the SCF technique is improved than other techniques. This method is fully utilizing the memory space of the nodes.

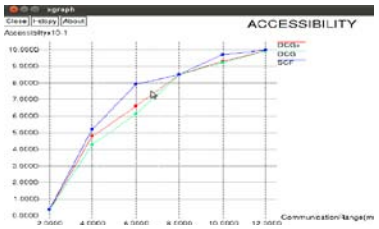

Figure 6b : Accessibility

VII. Conclusion

In this paper, the problem of selfish nodes is addressed from the replica allocation perspective. The selfish replica allocation could reduce the overall data accessibility in a MANET. Thus the solution has been proposed for the selfish node detection method to detect the selfish node appropriately and the detection time for the selfish node is also calculated. 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 combined credit risk and the collaborative watchdog method were applied to detect the selfish nodes. Every node in a MANET calculates credit risk information on other connected nodes individually to measure the degree of selfishness. The collaborative watchdog method is used to reduce the detection time of the each node. We also proposed novel replica allocation techniques. Extensive simulation shows that the proposed strategies outperform existing representative cooperative replica allocation techniques in terms of data accessibility, communication cost, and query delay. We plan to identify and handle false alarms in selfish replica allocation and apply the clustering method to improve the efficiency of the algorithm and also to diagnose the behavior of misbehaving nodes using EDCA method.

References Références Referencias

1. BaL Telecomm. Conf., pp. 178-182. Anderegg.L and Eidenbenz.S (2003), "Ad Hoc-VCG: A Truthful and Cost-Efficient Routing Protocol for Mobile Ad Hoc Networks with Selfish Agents," Proc. ACM MobiCom, pp. 245-259.
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. Chun B.G, Chaudhuri .K, With Barreno.M, Papadimitriou. C.H, and Kubiatowicz.J (2004), "Selfish Caching in Distributed Systems: A Game-Theoretic Analysis," Proc. ACM Sump. Principles of Distributed Computing, pp. 21-30. .
4. Heart .T and Madria.S.k (2006), "Data Replication for Improving Data Accessibility in Ad Hoc Networks," IEEE Trans. Mobile Computing, vol. 5, no. 11, pp. 1515-1532.
5. Heart T (2001), "Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility," Proc. IEEE INFOCOM, pp. 1568-1576.
6. Jae-Ho Chi Kym-Sun Shim, SunKen Lee, and Kun-Lung Wu (2012),"Handling Selfishness in Replica Allocation over a Mobile Ad Hoc Network" IEEE Transactions On Mobile Computing, Vol. 11, No. 2.
7. Jing Deng, cajun lie and Balakrishnan.K (2007) "An Acknowledgement Based Approach for the Detection of Routing Misbehavior in MANETs", IEEE Transactions On Mobile Computing, Vol. 6.
8. Khans and Ahmad. I (2009), "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.
9. Lucky and Yang's Y (2003), "Reputation Propagation and Agreement in Mobile Ad-Hoc Networks," Proc. IEEE Wireless Comm. And Networking Conf., pp. 1510-1515.
10. Paul's and Westhoff.D (2002), "Context Aware Detection of Selfish Nodes in DSR Based Ad-Hoc Networks," Proc. IEEE Glow
11. Final and cgh .G (2004), "Balancing the Tradeoffs between Data Accessibility and Query Delay in Ad Hoc Networks," Proc. IEEE Int'l Sump. Reliable Distributed Systems, pp. 289-298.
12. Yoo.Y and Agrawal .D.P (2006), "Why Does It Pay to be Selfish in a MANET," IEEE Wireless Comm., vol. 13, no. 6, pp. 87-97.
13. Zhai.J, Li. Q, and lax 2005), "Data Caching in Selfish Mantes Proc. Int'l Conf. Computer Network and Mobile Computing, pp. 208-217.
14. S. Marti, T. Guile K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad hoc Networks," Proc. ACM Mobi Com, pp. 255-265, 2000.
15. H. Miranda and L. Rodriguez, "Friends and Foes: Preventing Selfishness in Open Mobile Ad hoc Networks," Proc. IEEE Int'l Conf. Distributed Computing Systems Workshops, pp. 440-445, 2003.
16. Yangchow Zhang and Wing Lou "A Secure Incentive Protocol against selfishness in Mobile Ad Hoc Networks", IEEE Conf in WCNC Vol.3, pp. 1679-1684, 2004.
17. Hussein Haran and eyed A.Shahrestani, "Improving the reliability of ad-hoc on demand distance vector protocol", Journal on WSEAS transactions on communications, Vol.7, pp.695-704.200.
18. Debuts and RituparnaChaka ,"MABHIDS: A New Mobile Agent Based Black Hole Intrusion Detection System", Springer Computer information system, Volume 245, pp 85-94, 2011.
19. A. Data, M. Hauswirth, K.barer, "Updates in highly unreliable, replicated peer-to-peer systems", in: Proceedings of IEEE ICDCS'03, Providence, RI, USA, May 2003.
20. F. Dabber, M.F. Kai-shek, D. Karrer, R. Morris, I. Stoic "Wide-area cooperative storage with CFS ", in: Proceedings of the 18th ACM SOSP'01, Banff, Alberta, Canada, October 2001.
21. Chao-Tung Yang, Chun-Pin Fu, Ching-Hsien Hsu, File replication, maintenance, and consistency management services in data grids, J. Supercomputer. 53 (3) (2009) 411–439.
22. J. Zhao Q. Li, and X. Li, "Data Caching in Selfish Mantes," Proc. Int'l Conf. Computer Network and Mobile Computing, pp. 208-217, 2005.
23. 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. 1606-1619, Nov. 2006.
24. P. Padmanabhan, L. Grunewald, A. Valor and M. Atiquzzaman, "A Survey of Data Replication Techniques for Mobile Ad Hoc Network Databases," The Int'l J. Very Large Data Bases, vol. 17, no. 5, pp. 1143-1164, 2008
25. L. Yin and G. Cao, "Balancing the Tradeoffs between Data Accessibility and Query Delay in Ad Hoc Networks," Proc. IEEE Int'l Sump. Reliable Distributed Systems, pp. 289-298, 2004.
26. 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.
27. G. Ding and B. Bara "Peer-to-Peer File-Sharing over Mobile Ad Hoc Networks," Proc. IEEE Ann. Conf. Pervasive Computing and Comm. Workshops, pp. 104-108, 2004.