Volume 2, Issue 11, November 2012 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Combined Credit Risk and Collaborative Watchdog Method for Detecting Selfish Node over Mobile Ad-Hoc Network

S.J.K. Jagadeesh Kumar[1] ,R. Saraswathi[2]

[1]Professor

[2]PG Student

[1,2]Dept. of Computer Science and Engineering

[1,2]Sri Krishna College of Technology

Abstract- MANET is a peer-to-peer multihop mobile wireless network that has neither a fixed infrastructure nor a central server. The resource and mobility constraints of mobile nodes in mobile ad hoc network may lead to network partitioning or performance degradation. Several data replication techniques have been proposed to minimize performance degradation. 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. The selfish replica allocation could lead to reduce the overall data accessibility in a MANET. A combined credit risk & collaborative watchdog method 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 approach.

Keywords: Mobile ad hoc network, Selfish replica allocation, collaborative watchdog.

I. Introduction

A Mobile Ad Hoc Network, also called a MANET, is an autonomous collection of mobile nodes forming a dynamic wireless network. The administration of such a network is decentralized, *i.e.* each node acts both as host and router and forwards packets for nodes that are not within transmission range of each other. A MANET provides a practical way to rapidly build a decentralized communication network in areas where there is no existing infrastructure or where temporary connectivity is needed, *e.g.* emergency situations, disaster relief scenarios, and military applications.

The various challenges of the MANET are Autonomous, Dynamic topology, Device discovery, Bandwidth optimization, limited resources, Scalability, limited physical security, Infrastructure-less and self operated, Poor Transmission Quality, Ad hoc addressing, Network configuration and Topology maintenance challenge.In mobile ad hoc network, the 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 [8].

In general, replication can simultaneously improve data accessibility and reduce query delay, if the mobile nodes in a MANET together have sufficient memory space to hold both all the replicas and the original data. A mobile node may hold a part of the frequently accessed data items locally to reduce its own query delay. 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. 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 [3].

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 behaviour can potentially lead to a wide range of problems for a MANET. For example, selfish nodes may not transmit data to others to conserve their own batteries [1][5].

Fig. 1 illustrates an existing replica allocation scheme, DCG [4][6], where nodes N_1,N_2, . . .,N_6 maintain their memory space M_1,M_2, . . .,M_6, respectively, with the access frequency information in Table 1.In Fig. 1, a straight line denotes a wireless link, a gray rectangle denotes an original data item, and a white rectangle denotes a replica allocated.

Fig. 1, DCG seeks to minimize the duplication of data items in a group to achieve high data accessibility.

In this diagram, where N_3 behaves "selfishly" by maintaining M_3's, instead of M_3, to prefer the locally frequently accessed data for low query delay. 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.

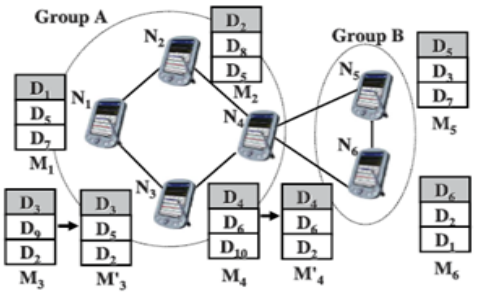


Fig:1 Example of selfish replica allocation

Data	Mobile host						Group	
	M_1	M_2	M_3	M_4	M_5	M_6	G_1	G_2
D_1	0.65	0.25	0.17	0.22	0.31	0.24	1.29	0.55
D_2	0.44	0.62	0.41	0.40	0.42	0.46	1.87	0.88
D_3	0.35	0.44	0.50	0.25	0.45	0.37	1.54	0.82
D_4	0.31	0.15	0.10	0.60	0.09	0.10	1.16	0.19
D_5	0.51	0.41	0.43	0.38	0.71	0.20	1.73	0.91
D_6	0.08	0.07	0.05	0.15	0.20	0.62	0.35	0.82
D_7	0.38	0.32	0.37	0.33	0.40	0.32	1.40	0.72
D_8	0.22	0.33	0.21	0.23	0.24	0.17	0.99	0.41
D_9	0.18	0.16	0.19	0.17	0.24	0.21	0.70	0.45
D_{10}	0.09	0.08	0.06	0.11	0.12	0.09	0.34	0.21

Table 1-Access Frequency of Nodes

In this paper, address 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. Easily find such cases in a typical peer-to-peer application. In this paper, refer to such a problem as the selfish replica allocation. The selfish replica allocation problem 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 technical contributions of this paper can be summarized as follows:

- Recognizing the selfish replica allocation problem
- Detecting the fully or the partially selfish nodes effectively
- Allocating replica effectively
- Apply the collaborative watchdog method to detect the selfish node.
- Verifying the proposed strategy

The Section II describes the related work .The overview of system model and the node behaviour model from the viewpoint of selfish replica allocation is described in Section III. The proposed detection method and the replica allocation techniques are presented in Section IV. The simulation scenario is presented in section V. The performance evaluation was presented in Section VI and the conclusion of the paper is presented in Sections VII.

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 behavior are termed selfish. Selfishness is usually passive behavior. Selfish and malicious behaviors are usually distinguished based on the node's intent. Various techniques have been proposed to handle the problem of selfish behavior 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 [11]. In reputation-based a large number of schemes belong to the first category, with varying implementations. One advantage of such schemes could be their quick convergence in detecting node misbehavior, especially in a large ad hoc network, due to increased information regarding a particular node's behavior. 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 [7].In credit-payment techniques [11], each node gives a credit to others, as a reward for data forwarding. 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. The game theory-based techniques want to find the Nash Equilibrium point [12] to maximize system performance. There are many disadvantages above those methods such as

- Reputation information about their peers is themselves trustworthy in selfishness finding nodes.
- Collusion among nodes that misreport reputation information.
- Selfish nodes may not transmit data to others to conserve their own batteries.

B. Replica Allocation and Caching Techniques

In this related work [11], some of the effective replica allocation techniques are suggested, including static access frequency, dynamic access frequency and neighbourhood (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 proposes data replication techniques that address both query delay and data accessibility in a MANET. The work demonstrates such a trade-off and proposes techniques to balance it. The work introduces the cooperative caching-based data access methods, including CachePath, CacheData, and Hybrid. Differing from all the above-mentioned replica allocation or caching techniques, consider selfish nodes in a MANET [3].

III. System Model

In this paper, 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. 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 tuple(Nj,Nk) of nodes in the network. The following assumptions are made and it is similar to those in [6].

- Each node in a MANET has a unique identifier.
- 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.
- 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.
- 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.

Therefore, the three types of behavioural states for nodes from the viewpoint of selfish replica allocation is described.

1. Type-1 node: The nodes are non selfish nodes. The nodes hold replicas allocated by other nodes within the limits of their memory space.
2. 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.
3. 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.

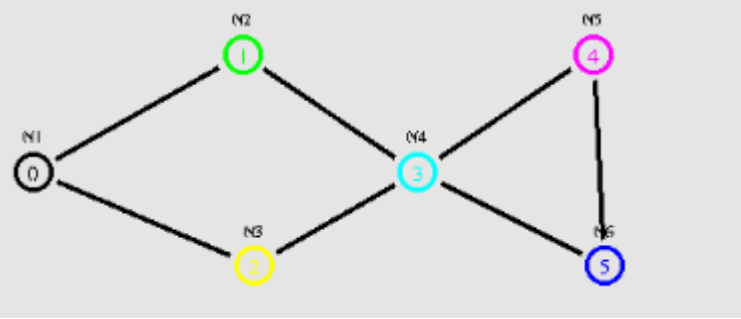


Fig: 2 System Model

IV. Proposed Strategy

A. Overview

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 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 detecting by the self replica allocation. 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. In the real world, a friendship, which is a form of social bond, is made individually. 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-centered friendship tree to reduce communication cost, while achieving good data accessibility.
- Verifying the proposed strategy: The simulation results verify the efficacy of our proposed strategy.

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.

Enhance the work to collaborative watchdog based on contact dissemination of the detected selfish nodes. If one node has previously detected a selfish node using its watchdog it can spread this information to other nodes when a contact occurs. We say that a node has a *positive* if it knows the selfish node. The detection of contacts between nodes is straightforward using the node's watchdog. Notice that the watchdog is overhearing the packets of the neighborhood; thus, when it starts receiving packets from a new node it is assumed to be a new contact. Then, the node transmits one message including all known positives it knows to this new contacted node.

Advantages:

- Easily detects the selfish node without collision.
- SCF-tree based replica allocation is performed in a fully distributed manner.
- Cooperative replica allocation techniques in terms of data accessibility, communication cost, and query delay so it is efficient.

B. 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 SS_i^K , and the number of N_k's shared data items, denoted as ND^k_i , observed by a node N_i ,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 SS_i^K and ND^k_i , node N_k may be treated as a valuable node by node N_i.

As the query processing-specific feature, utilize the ratio of selfishness alarm of N_k on N_i, denoted as P_i^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 P_i^k gets larger, node N_i will treat N_k as a risky node because a large P_i^k means that N_k' cannot serve N_i's requests due to selfishness in its memory usage. The value of the CR_i^k is the credit risk of node N_i. § is the threshold value of node N_i. α is the system parameter, where $0 \leq \alpha \leq 1$.The formula for finding the credit risk is

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

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;
   }
  wait until replica allocation is done;
  for (each connected node Nk)
   {
     if (Ni has allocated replica to Nk)
      {
          ND^k_i= the number of allocated replica;
          SS^k_i = the total size of allocated replica;
       }
```



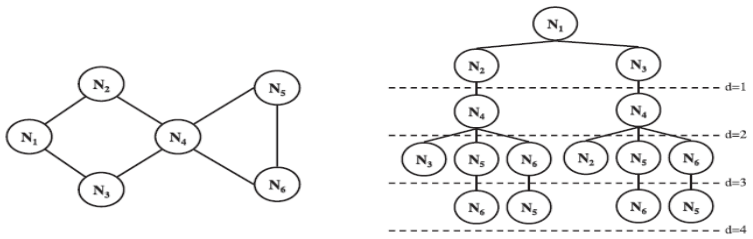
```
        else
        {
            ND^k_i =1;
            SS^k_i = the size of a data item;

    } } }
```

C. 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 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 $N_i \in IN_i$, $IN_i \_ IN$, and $IL_i \_ IL$. Based on G_i^{ns}, N_i builds its own SCF-tree, denoted as T_i^{SCF}. Algorithm 3 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. 2 illustrates the network topology and some SCF-trees of N_1 and N_2 in Fig. 1.



(a) Sample topology G (b) SCF-tree of N_1

Fig 2: Examples of self centred friendship tree

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 € IN^a_j)
   {
        if (distance between Na and the root > d )
        continue;
        else if (Na is an ancestor of Nj in T_i^SCF )
        continue;
        else
        {
            append Na to T_i^SCF as a child of Nj;
            checkChildnodes(Na);
        } } }
```

D. Implementing Collaborative Watchdog

1. *Identifying Selfish contact*

In this module, detection time of selfish nodes have to be reduced based on contact dissemination. If one node has previously detected a selfish node using its watchdog it can spread this information to other nodes when a contact occurs. We say that a node has a positive if it knows the selfish node. The watchdog is overhearing the packets of the neighborhood. Thus, when it starts receiving packets from a new node it is assumed to be a new contact. Then, the node transmits one message including all known positives it knows to this new contacted node. The number of messages needed for this task is the overhead of the collaborative watchdog.

The collaborative nodes have no information about the selfish nodes. A collaborative node can have a positive when a contact occurs in the system. In the model one of the nodes is the selfish node. Then, the collaborative node can detect it using its watchdog and have a positive about this selfish node. Nevertheless, a contact does not always simply detection. To model this fact, we introduce a probability of detection (pd).This probability depends on the effectiveness of the watchdog and the type of contact (for example if the contact time is very low, the watchdog does not have enough information to evaluate if the node is selfish or not).

A 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. 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.

2. Implementation of 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 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 reflect that either a message with the information about the selfish nodes is lost or that a node temporally does not collaborate. The following steps are given below:

- In this module, detection time of selfish nodes have to be reduced based on contact dissemination. If one node has previously detected a selfish node using its watchdog it can spread this information to other nodes when a contact occurs. So that a node has a positive if it knows the selfish 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 modelled 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).

Algorithm3: *Finding the detection time of the selfish node*

- p_{ij} denoting the transition rate from transient state si to absorbing state sj.
- Given a state si =(c) the following transitions can occur:
- (c) to (c+1): This case takes place when a collaborative node changes from NOINFO state to POSITIVE state.
- The transition probability is $t_c = (\lambda p_d + \lambda p_c c)(C - c)$.
- λp_d represents the probability of detection of a selfish node.
- $\lambda p_c c$ *is* the probability of transmission for the information of the selfish node.
- Finally, factor (C − c) represents the number of pending nodes.
- (c) to (c): This is the probability of no changes, and its value is $t_0 = 1 - t_c$.

V. Simulation Scenario

Our simulation model is similar to that employed in [5].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) 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. After reaching the destination, it again stops for a pause time and repeats this behavior. 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 Ni ($1 \leq i \leq 40$) holds data Di 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 2 describes the simulation parameters.

TABLE: 2
Simulation Parameters

Parameter	Value
Number of nodes	40
Number of data items	40
Radius of communication range	19
Size of the network	50 X 50
Size of the memory space	40
Relocation period	8192
Selfish nodes	(0-100)

VI. PERFORMANCE EVALUATION

This section is first devoted to evaluating the performance of our collaborative watchdog 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). 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 3 shows the influence of the number of selfish nodes *S* for $N = 50$. As expected, the detection time increases when the number of selfish nodes is higher.

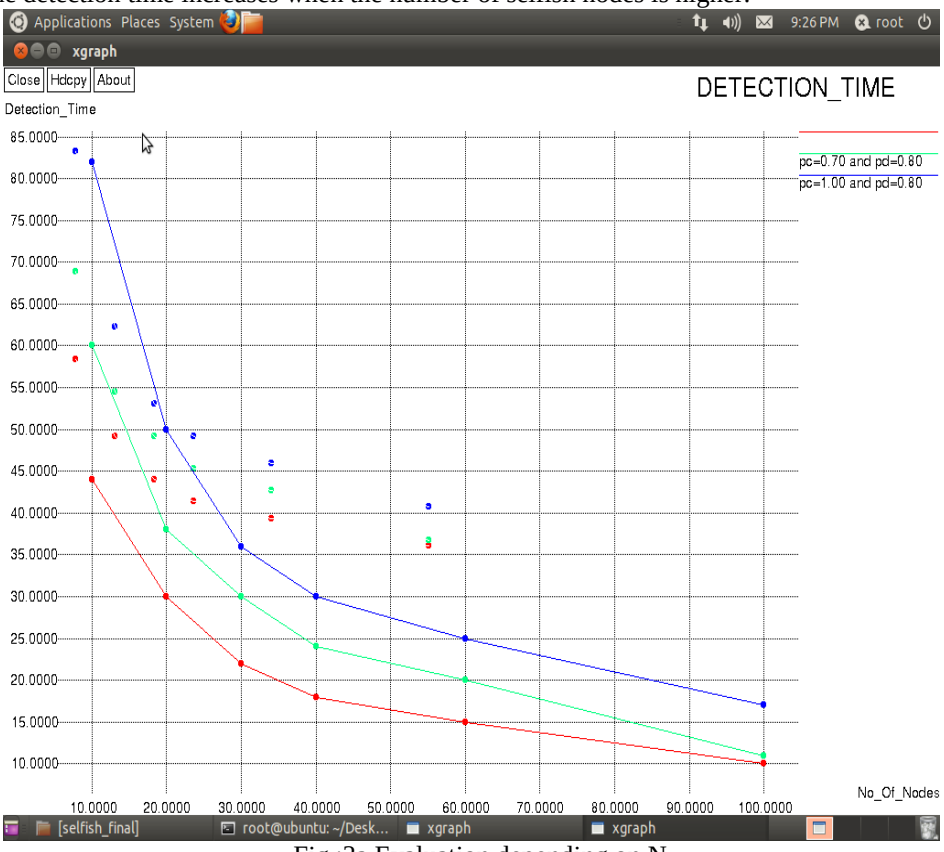


Fig :3a Evaluation depending on N

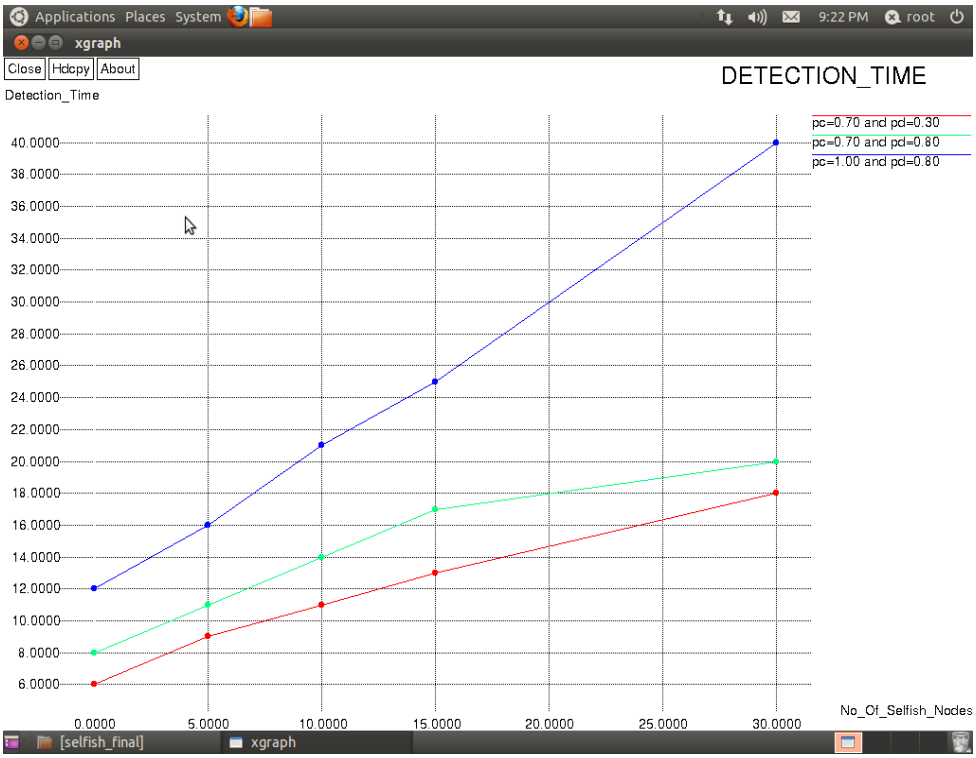


Fig.3b Evaluation depending on selfish node

VII. Conclusion

In contrast to the network viewpoint, address the problem of selfish nodes from the replica allocation perspective. The selfish replica allocation could reduce the overall data accessibility in a MANET. The proposed 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. Applied the notion of credit risk and the collaborative watchdog method to detect 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 & cost of the each node. Since traditional replica allocation techniques failed to consider selfish nodes, also proposed novel replica allocation techniques. The proposed strategies improves the data accessibility, reduces communication cost, and average query delay and also to reduce the detection time of the selfish node.

REFERENCES

[1] 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.

[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]. B.-G. Chun, K. Chaudhuri, H. Wee, M. Barreno, C.H. Papadimitriou,and J. Kubiatowicz, "Selfish Caching in Distributed Systems:A Game-Theoretic Analysis," Proc. ACM Symp. Principles of Distributed Computing, pp. 21-30, 2004.

[4]. Jae-Ho Choi, Kyu-Sun Shim, SangKeun Lee, and Kun-Lung Wu, "Handling Selfishness in Replica Allocation over a Mobile Ad Hoc Network" IEEE Transactions On Mobile Computing, Vol. 11, No. 2, February 2012.

[5]. 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.

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

[7].Y.Liu and Y. Yang,"Reputation Propagation and Agreement in Mobile Ad-Hoc Networks," Proc. IEEE Wireless Comm. And Networking Conf., pp. 1510-1515, 2003.

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

[9]. 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.
[10]. 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.
[11]. Y. Yoo and D.P. Agrawal, "Why Does It Pay to be Selfish in a MANET," IEEE Wireless Comm., vol. 13, no. 6, pp. 87-97, Dec. 2006.
[12] S.U. Khan and I. Ahmad, "A Pure Nash Equilibrium-Based Game Theoretical Method for Data Replication across Multiple Servers," IEEE Trans. Knowledge and Data Eng., vol. 21, no. 4, pp. 537-553,Apr. 2009.
[13] L.J. Mester, "What's the Point of Credit Scoring?" Business Rev.pp. 3-16, Sept. 1997.

