IMPROVING DATA ACCESSIBILITY RATE BY EFFICIENTLY USING SELFISH NODES IN REPLICA ALLOCATION OVER A MANET

[1]S.Prakash, [2]V.Arthi, [3]M.Aswini, [4]P.Sivaranjini

[1,2,3,4]Department of ECE, Alpha college of Engineering and Technology, Pudhucherry, (India)

ABSTRACT

A mobile ad-hoc network is a network that allows mobile servers and clients to communicate in the absence of a fixed infrastructure. In this network some of the nodes do not sharing the packets to other node to conserve their resources such as energy, bandwidth and power. The nodes which are act selfishly to conserve their resource are called selfish node. Since traditional selfish node detection methods are found to detect the nodes which do not participate in packet forwarding but they fail to detect the selfish nodes which does not allocate replica for the purpose of other nodes. Here we propose a new method of credit payment technique and SCF tree. By using the credit payment technique can identify the selfish node in the network and increase the data accessibility. By simulation result, we show that the proposed SCF tree protocol achieves high packet delivery ratio while attaining low delay, high speed and overhead.

***Keywords: Mobile Ad-Hoc Networks, Fixed Infrastructure, Data Accessibility*.**

1 INTRODUCTION

Mobile ad hoc networks have potential applications in civilian and military environments such[12] as disaster recovery efforts, group conferences, wireless offices, mobile info stations (in tourist centers, restaurants, and so on), and battlefield maneuvers, making them a focus of current research.

A battlefield ad hoc network might consist of several commanding officers and a group of soldiers. The soldiers could access officers' information centers for detailed geographic information, information about the enemy, new commands, and so on. Because neighboring soldiers tend to have similar missions and thus common interests, several soldiers might need to access the same data at different times. Having a nearby soldier serve later accesses to this data instead of the faraway information center saves battery power, bandwidth, and time.

In ad hoc networks, mobile nodes communicate with each other using multi hop wireless links. Due to a lack of infrastructure support, each node acts as a router, forwarding data packets for other nodes.[1] Most previous research in ad hoc networks focused on the development of dynamic routing protocols that can efficiently find routes between two communicating nodes. Although routing is an important issue, the ultimate goal of ad hoc networks is to provide mobile nodes with access to information. If mobile users around info stations, which have limited coverage, form an ad hoc network, a mobile user who moves out of the range of a particular info station can still access the data it contains. If one of the nodes along the path to the data source has a cached copy of the requested data, it cans forward the data to the mobile user, saving bandwidth and power. Thus, if

mobile nodes can work as request forwarding routers, they can save bandwidth and power and reduce delays. There are two types of MANETs: open and closed. An open MANET comprises of different users, having different goals, [13] sharing their resources to achieve global connectivity, as in civilian applications. This is different from closed MANETs where the nodes are all controlled by a common authority, have the same goals, and work toward the benefit of the group as a whole. Open environment of a MANET may lead to misbehaving nodes.

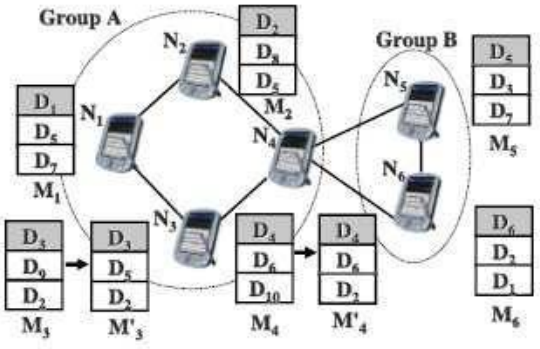


Fig.1. Example of Selfish Replica Allocation.

Let us consider the case where N3 behaves "selfishly" by maintaining M3', instead of M3, to prefer the locally frequently accessed data for low query delay. In the original case, D3, D9, and D2 were allocated to N3. However, due to the selfish behavior, D3, D5, and D2, the top three most locally frequently accessed items, are instead maintained in local storage. Thus, other nodes in the same group, i.e., N1, N2, and N4, are no longer able to access D9. This show cases degraded data accessibility, since N1, N2, and N4 cannot fully leverage N3's memory space as intended in cooperative replica sharing.

We believe that the partially selfish nodes (e.g., N4 in Fig. 1) should also be taken into account, in addition to the fully selfish nodes (e.g., N3 in Fig. 1), to properly handle the selfish replica allocation problem. We therefore need to measure the "degree of selfishness" to appropriately handle the partially selfish nodes. Motivated by this concept of "partial selfishness," we borrow the notion of credit risk (CR) from economics to detect selfish nodes. Since the credit risk is calculated from several selfishness features in this paper, it can measure the degree of selfishness elaborately. In our scheme, a node can measure the degree of selfishness of another node, to which it is connected by one or multiple hops in a MANET.[3]

The proposed selfish node detection method and novel replica allocation technique 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.

First we detect the selfish node by self replica allocation. Use those replica we devise novel replica allocation techniques with the developed selfish node detection method. 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. For example, although A and B are friends, the friends of A are not always the same as the friends of B. With the help of SCF tree, we aim to reduce the communication cost, while still achieving good data accessibility. 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.

1.1 Node Behavioral Model

Three types of behavioral states for nodes summarized as follows:

- 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 into two parts: selfish and public area.

The detection of the type-3 nodes is complex, because they are not always selfish. In some cases, a type-3 node might be considered as non selfish since the node shares part of its memory space.

1.2 Detecting Selfish Node

The network is modeled as a set of N wireless mobile nodes with C collaborative nodes and S selfish nodes (N = C +S). At a specific period, or relocation period, each node executes the following procedures:

- Each node detects the selfish nodes based on credit risk scores (CR).
- Each node makes its own (partial) 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. The CR score is updated accordingly during the query processing phase to effectively measure the “degree of selfishness”.

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

A node wants to know if another node is believable, in the sense that a replica can be paid back, or served upon request to share a memory space in a MANET. With the measured degree of selfishness, a novel tree that represents relationships among nodes in a MANET is proposed for replica allocation termed the SCF-tree. The key strength of the SCF-tree-based replica allocation techniques is that it can minimize the communication cost, while achieving high data accessibility. This is because each node detects selfishness and makes replica allocation at its own discretion, without forming any group or engaging in lengthy negotiations. At each relocation period, node N_i detects selfish nodes based on nCR_i^k. Each node may have its own initial value of P_i^k as a system parameter. Interestingly, the initial value of P_i^k can represent the basic attitude toward strangers. For instance, if the initial value equals zero, node N_i always treats a new node as a non selfish node. Therefore,N_i can cooperate with strangers easily for cooperative replica sharing. Replicas of data items are allocated by allocation techniques. After replica allocation, N_i sets ND_i^k and SS_i^k accordingly. Recall that both ND_i^k and are SS_i^k stimulated values, not accurate ones.

1.3 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.

Prior to building the SCF tree each node makes it own partial topology graph $G_i = (IN_i, IL_i)$ which is a component of the graph G,G_i consists of finite set of the nodes connected to N_i and a finite set of the links, the SCF tree consists of only non selfish nodes, we need measure the degree of selfishness to apply real world friendship management to replica allocation in a MANET.

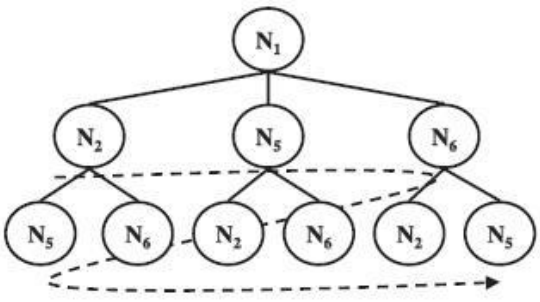


Fig. 2.SCF tree of N1

1.4 Allocating Replica

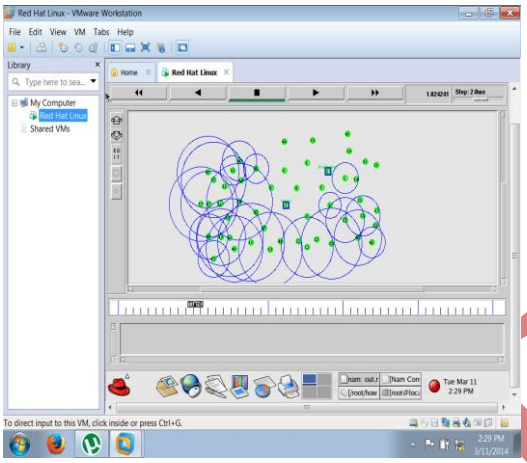
After building the SCF-tree, a node allocates replica at every relocation period. Each node asks non selfish nodes within its SCF-tree to hold replica when it cannot hold replica in its local memory space. Since the SCF-tree based replica allocation is performed in a fully distributed manner, each node determines replica allocation

individually without any communication with other nodes.

The objective of the SCF tree based replica allocation technique is to achieve good data accessibility with low communication cost in the presence of selfish nodes; each node processes the following procedures:

- Each node allocates replica at its discretion
- When each node receives a request for replica allocation from N_k during a relocation period, it determines whether to accept the request.
- If the request is accepted, each node maintains its Mp based on nCR_k^i.

II EXISTING SIMULATION



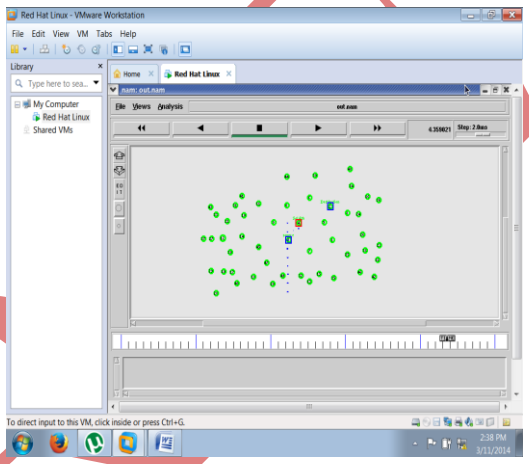


Fig.3.Absorbs the Shortest Path **Fig.4.Packet Dropping Occurs In Node 4**

We have used AODV routing protocol in 10 mobile nodes. For evaluation purpose, we mostly consider source, destination and attacker node whereas other nodes assist in routing of the packets and have their own purpose. The basic node scenario is shown in figure 3. In figure 4 and figure 6 node 4 is selfish nodes. Here the source node 3 wants to send data packets to the destination node 2, so it will initiate route discovery process by sending route request packets in the network. Whenever selfish node receives route request packet it will immediately send false route reply message with highest destination sequence number and minimum hop count to the source node 3. When the route reply reaches first to the source node, the source node assumes that the route discovery process is completed and it will avoid the S reply from other nodes. Source begins to sends all the data packets to the selfish node. So selfish node, consumes all the data packets passed by source node. Similarly, node 4 consumes all the data packets sent by the source node 3. Figure 3 and figure 4 shows the simulation results generated after detecting selfish node and avoiding the effect of it through our implemented IDS detection technique.

When the simulation is compiled, we saw that sending node is sending the messages to receiving node properly. Figure 4 shows that CBR packets are reaching from source node 3 to the destination node 2 as expected. With our intrusion detection technique, we have nullified the effect of intrusion. So even if a malicious node 4 present nearest to source node 0, it is not able to capture the packets passing through its neighbor. Similarly, figure 8 shows that CBR packets are reaching from source node 3 to the destination node 2 as expected. In this way, we

have detected an intrusion in ad hoc network and avoided its effect in the network. As we have nullified the effect of selfishness in the network, the performance of the network is improved.

III STRATEGIES FOR HANDLING SELFISH BEHAVIOUR IN NODES

3.1. Reputation Based Technique

In reputation based technique [5] a node receives one unit of credit for forwarding a message of another node and such credits are deducted from the sender or the destination. In reputation based technique, a node monitors the transmission of a neighbor to make sure that the neighbor forwards others traffic. If the neighbor does not forward others traffic, it is considered as selfish node and this uncooperative reputation is propagated throughout the network. Each node in the network runs the Confidant protocol. It observes the behavior of neighbor nodes to detect misbehavior such as packet dropping. This requires nodes to run in promiscuous mode. When the monitor finds misbehavior, it notifies the reputation system, which manages a table containing nodes and their ratings. If the number of times a node misbehaves exceeds a threshold, the reputation system updates the node's rating. If a node's rating falls below a threshold, the system considers it a malicious node. The reputation system maintains a list containing the selfish nodes. When forwarding packets, nodes avoid next nodes on the list. When the reputation system detects a selfish node, it notifies the trust manager to broadcast an alarm message in the network. Trust managers also receive alarms from other trust managers. The path manager ranks the path according to the ratings of the nodes on the path. It deletes all paths containing malicious nodes and drops route requests received from selfish nodes.

3.1.1. Watchdog mechanism

The watchdog [6] is one of the mechanisms which detect selfish nodes by running a misbehaving node locator on every host that maintains a buffer of recently sent packets. It overhearing packets transmitted and compares it with the packets in the buffer to found if there is a match between the packets sent. If the packet has been sent from the buffer then watchdog removes the packet from the buffer.

If there is any mismatch occurs and certain packets occupy the buffer for more than particular time, the watchdog increases a failure count for the node responsible for forwarding the packet. If the count exceeds a threshold value, the watchdog considers that host as a misbehaving node.

3.1.2. Path rater method

A path rater [6] is a mechanism which maintains a rating for every other host in the network. To choose a route that is considered to be reliable, it calculates a path metric by averaging the rating of the nodes on the paths and chooses the path with the highest metric.

If any node gets very low rating, it should be considered as a selfish node and thus excludes them from routing. It increases throughput by 17% in a network with moderate mobility and increases network throughput by 27%, with extreme mobility. Path rater also having some of the draw backs such as increases overhead in the transmissions from 9% to 17% with moderate mobility. Without using watchdog path rater is inefficient. Using watch dog is necessary in all the detection systems.

3.2. Credit-payment technique

Ad hoc-VCG [7] is one of the reactive routing protocols, which starts discovering routing paths when a network node initiates a session. Ad hoc-VCG uses a DSR like route discovery protocol that provides all information about shortest paths to the destination node. The destination node calculates the shortest path and the VCG payments and sends this information back to the source. In the data transmission phase, the source sends packets combined with electronic payments to the destination along the shortest path. Ad hoc-VCG is reliable against a single cheating node but it may fail in the presence of coalitions of nodes (coalition forming) which try to maximize their total payments. It provides truthfulness and assures cost efficiency but it having some disadvantages such as excessive overhead and Coalition-forming.

3.3. Game theory based technique

Selfish nodes are sometimes called as freeloaders [8] getting resources from the network and did not upload any resources to the network. Minimizing the effects of freeloaders require the services of some external centralized authority. The inclusion of third party produces overhead in tracking, storing and processing the behavior of other nodes.

IV PROPOSED STRATEGY

Our strategy consists of four parts: 1) path finding, 2) detecting selfish nodes, 3) building the SCF-tree, and 4) 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 (partial) 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.

The CR score is updated accordingly during the query processing phase. We borrow the notion of credit risk from economics to effectively measure the "degree of selfishness." In economics, credit risk is the measured risk of loss due to a debtor's nonpayment of a loan. A bank examines the credit risk of an applicant prior to approving the loan. The measured credit risk of the applicant indicates if he/she is creditworthy. We take a similar approach. A node wants to know if another node is believable, in the sense that a replica can be paid back, or served upon request to share a memory space in a MANET.

With the measured degree of selfishness, we propose a novel tree that represents relationships among nodes in a MANET, for replica allocation, termed the SCF-tree. The SCF-tree models human friendship management in the real world. The key strength of the SCF-tree-based replica allocation techniques is that it can minimize the communication cost, while achieving high data accessibility. This is because each node detects selfishness and makes replica allocation at its own discretion, without forming any group or engaging in lengthy negotiations.

4.1 Flow Chart

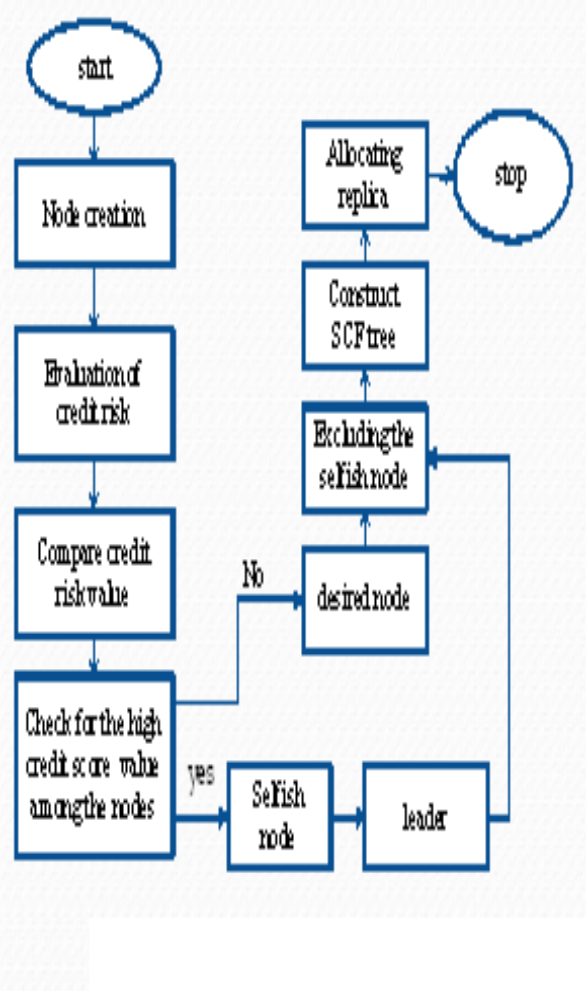


Fig.5.Proposed Flow Chart

4.2 Algorithm

01: creating the node

02: for{set i 0}{$i < $val(nn)}{incr i}{

03: set node_($i)[$ns_node]

04: }

05: evaluation of credit risk

06: for{set i 0}{$i < $val(nn)}{incr i}{

07: $node_($i) set x_$x1($i)

08: $node_($i) set y_$y1($i)

09: $node_($i) set z_0.0

10:}

11:compare the credit risk value for each nodes

12: detection(){

13: for(each connected node N_k)

14: if(n$CR_i^k < \delta$)N_k is marked as non selfish;

15: else N_k is marked as selfish;}

16: constructScfTree(){

17: append set I to SCF-tree as the root node;

18: checkChildnodes(set i);

19: return SCF-tree;

20:}

4.3 Simulation Study and Result

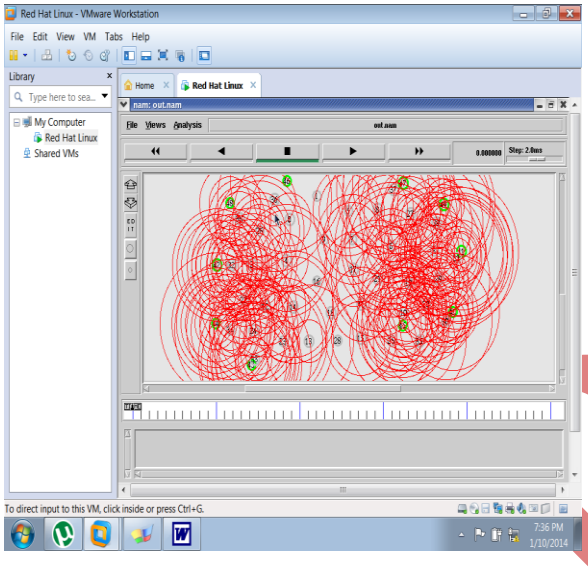


Fig.6.Detection of selfish nodes

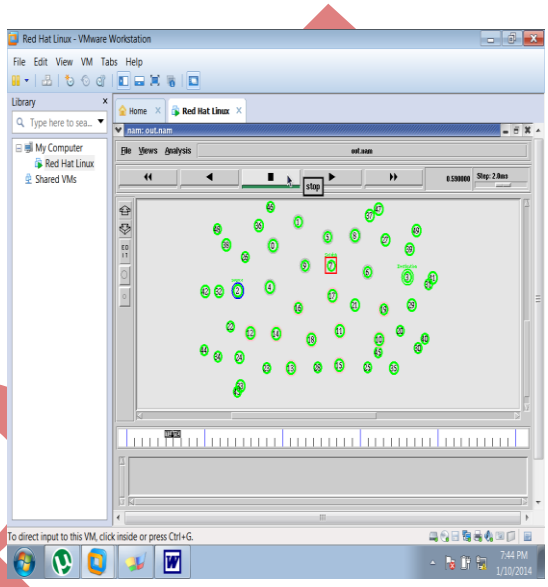


Fig.7.Packet transmission

V PERFORMANCE EVALUATION METRICS AND RESULTS

5.1 Simulation Parameter

Parameters	Values
No. of nodes	50
No. of packets	1000
Start Simulation time	0.1msec
Stop Simulation time	5.02sec
Routing Protocol	AODV
Dimension area	800x600
Mac protocol	IEEE 802.11

The following five important performance metrics are considered for evaluation of these routing protocols.

Throughput: Throughput is the measure of how fast we can actually send through network. The number of packets delivered to the receiver provides the throughput of the network.

Packets dropped*:* Some of the packets generated by the source will get dropped in the network due to high mobility of the nodes, congestion of the network etc.

Packet delivery ratio: The ratio of the data packets delivered to the destinations to those generated by the CBR sources.

Normalized routing overhead: The number of routing packets transmitted per data packet delivered at the destination. Each hop-wise transmission of a routing packet is counted as one transmission

Optimal path length: It is the ratio of total forwarding times to the total number of received packets.

5.2 Simulation Results

The Communication cost (fig 9) of a network using SCF-tree is minimum compared to normal replica (DCG). Whereas Data Accessibility, Overall Selfishness, Average Query delay of a network is maximum Communication Cost: We evaluate several replica allocation techniques in terms of communication cost. Our intuition was that our techniques outperform SAF, while being inferior to SCF. This intuition is confirmed by the results in figure 9.SAF shows the worst performance in all cases, since group members need to communicate with each other in detecting selfish nodes and allocating/relocating replica. We report that, on average, about 70 percent of total communication cost in the SAF technique is caused by replica allocation/relocation, while about 30 percent is caused by selfish node detection. As expected, DCG shows the best performance, since no detection of selfish nodes or group communication is made. Although SCF and DCG techniques show better performance than SAF in communication cost, they are expected to show poor performance in data accessibility in the presence of selfish nodes. Interestingly, our analysis reveals that our techniques, which detect selfish nodes, considerably outperform DCG, which does not perform the selfishness detection procedure. This verifies the efficacy of our fully distributed way of detecting selfish nodes and allocating replica, i.e., no group communication.

Throughput: In figure 8, the graph shows that the difference of normal network, when selfish node comes in the network and after detection and elimination of selfish node from the network. It clearly shows that the performance improvement in the network.

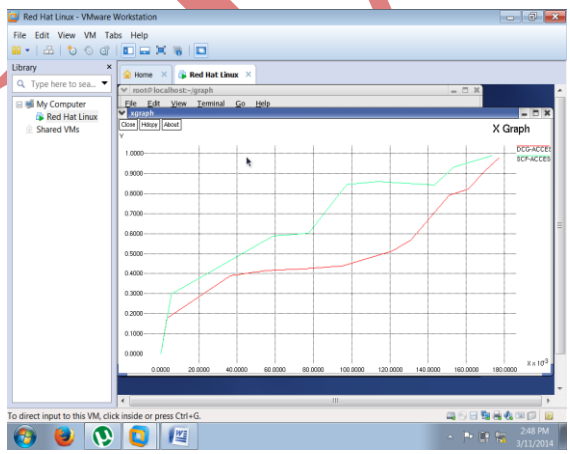


Fig.8.Throughput Graph

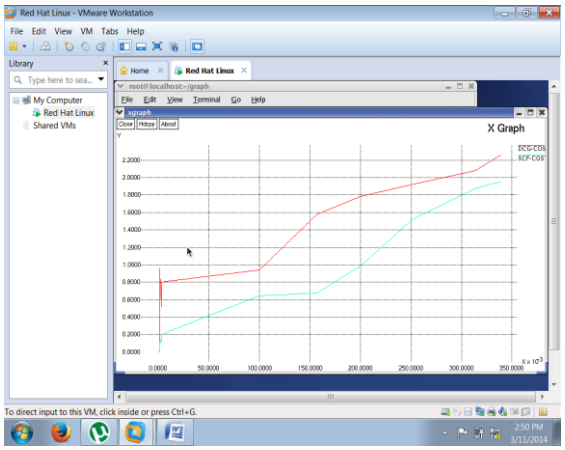


Fig.9.Communication Cost Graph

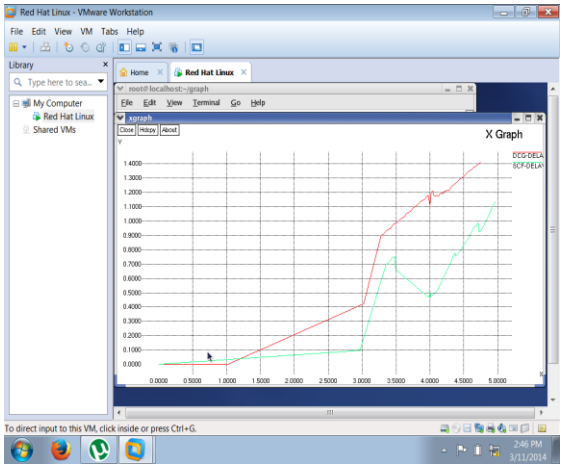


Fig.10.Delay graph

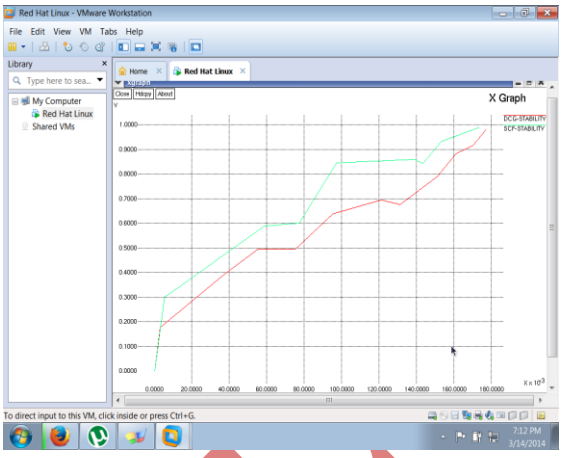


Fig.11.Stability graph

Packet delay: In figure 10, the graph shows that the packet delay in all three cases that is normal network, when selfish node comes in the network and after detection of selfish node in the network.

Fig 8 shows the throughput graph. Red colored graph shows the throughput when there is attack in the network and green colored graph shows the improved throughput after applying our credit payment technique. The throughput is increasing as the number of normal increases with our technique. Increase in throughput improves the network performance. Fig 10 shows the Delay graph. Red colored graph shows the average Delay in milliseconds when there is attack in the network and green colored graph shows the improved average Delay in milliseconds after applying our credit payment technique. The packet delay is reduced with our technique. Fig 9 shows the communication cost graph. Red colored graph shows the communication cost when there is attack in the network and green colored graph shows that reduce the communication cost after applying our credit payment technique. Fig 11 shows the Stability graph. Red colored graph shows the stability when there is attack in the network and green colored graph shows the improved throughput after applying our credit payment technique. The stability is increasing as the number of normal increases with our technique. Increase in stability improves the network performance.

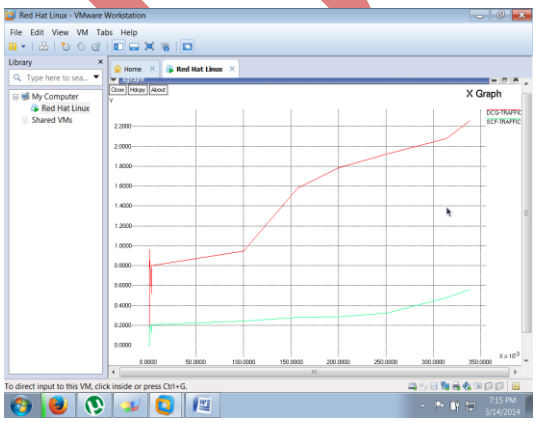


Fig.12.Traffic Load Graph

Fig 12 shows the traffic load graph. Red colored graph shows the traffic load when there is attack in the network and green colored graph shows that reduce the traffic load after applying our credit payment technique. In this way, we have improved three network parameters namely: throughput, packet delay and packet communication cost. By improving these three parameters of the network, we enhanced the network performance.

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] E. Adar and B.A. Huberman, "*Free Riding on Gnutella*," First Monday, vol. 5, no. 10, pp. 1-22, 2000.

[2] 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.

[3] 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.

[4] R.F. Baumeister and M.R. Leary, "*The Need to Belong: Desire for Interpersonal Attachments as a Fundamental Human Motiva-tion*," Psychological Bull., vol. 117, no. 3, pp. 497-529, 1995.

[5] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, and J. Jetcheva, "*A Performance Comparison of Multi-Hop Wireless Ad Hoc Net-work Routing Protocols*," Proc. ACM MobiCom, pp. 85-97, 1998.

[6] 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.

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

[8] E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati, "*Managing and Sharing Servents'*

Reputations in P2P Systems," IEEE Trans. Knowledge and Data Eng., vol. 15, no. 4, pp. 840-854, July/Aug. 2003.

[9] 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.

[10] M. Feldman and J. Chuang, "*Overcoming Free-Riding Behavior in Peer-to-Peer Systems*," SIGecom Exchanges, vol. 5, no. 4, pp. 41-50, 2005.

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

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

[13] 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.

[14] 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.

[15] 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.

[16] N. Laoutaris, G. Smaragdakis, A. Bestavros, I. Matta, and I. Stavrakakis, "*Distributed Selfish Caching*," IEEE Trans. Parallel and Distributed Systems, vol. 18, no. 10, pp. 1361-1376, Oct. 2007.

[17] N. Laoutaris, O. Telelis, V. Zissimopoulos, and I. Stavrakakis, "*Distributed Selfish RepLication*," IEEE Trans. Parallel and Distributed Systems, vol. 17, no. 12, pp. 1401-1413, Dec. 2006.

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

[19] M. Li, W.-C. Lee, and A. Sivasubramaniam, "*Efficient Peer-to-Peer Information Sharing over Mobile Ad Hoc Networks*," Proc. World Wide Web (WWW) Workshop Emerging Applications for Wireless and Mobile Access, pp. 2-6, 2004.

[20] 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.

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

[22] L.J. Mester, "*What's the Point of Credit Scoring?"* Business Rev., 3-16, Sept. 1997.

[23] P. Michiardi and R. Molva, "*Simulation-Based Analysis of Security Exposures in Mobile Ad Hoc Networks*," Proc. European Wireless Conf., pp. 1-6, 2002.

[24] 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.

[25] A. Mondal, S.K. Madria, and M. Kitsuregawa, "*An Economic Incentive Model for Encouraging Peer Collaboration in Mobile-P2P Networks with Support for Constraint Queries*," Peer-to-Peer Networking and Applications, vol. 2, no. 3, pp. 230-251, 2009.

[26] M.J. Osborne, An Introduction to Game Theory. Oxford Univ., 2003.P. Padmanabhan, L. Gruenwald,

A. Vallur, 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.

[27] 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.

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

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

[30] 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.

[31] L. Yin and G. Cao, "*Balancing the Tradeoffs between Data Accessibility and Query Delay in Ad Hoc Networks*," Proc. IEEE Int'l Symp. Reliable Distributed Systems, pp. 289-298, 2004.

[32] 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.

[33] J. Zhai, Q. Li, and X. Li, "*Data Caching in Selfish Manets,"* Proc. Int'l Conf. Computer Network and Mobile Computing, pp. 208-217, 2005.