HANDLING SELFISHNESS OVER MOBILE AD HOC NETWORK

Madhuri D. Mane and B. M. Patil
P.G. Dept., MBES College of Engineering Ambajogai, Maharashtra, India

ABSTRACT
Mobile Ad hoc Network (MANET) also called as mobile mesh network. It is a self-configuring network of mobile devices connected by wireless links. Here we assume all mobile nodes cooperate fully with the functionalities of the network. In reality, however, some nodes may selfishly decide only to cooperate partially, or not at all, with other nodes [1]. In this paper we have to find selfish nodes and recover them. These selfish nodes could then reduce the overall data accessibility and increase delay in the network. In this paper, we examine the impact of these selfish nodes and reduce the delay.

KEYWORDS: *MANET, Proactive, Reactive, selfish, SCF, eSCF, SCE-DS, SCF-CN.*

I. INTRODUCTION

Wireless cellular system has been in use since 1980s. Wireless system operates with the aid of a centralized supporting structure such as an access point. These access points assist the wireless users to keep connected with the wireless system, when they roam from one place to other. In wireless system the device communicate via radio channel to share resource and information between devices. Due to presence of a fixed supporting structure, limits the adaptability wireless system is required easy and quick deployment of wireless network. Recent advancement of wireless technologies like Bluetooth, IEEE 802.11 introduced a new type of wireless system known as Mobile ad-hoc network (MANETs) [13], which operate in the absence of central access point. It provides high mobility and device portability's that enable to node connect network and communicate to each other. It allows the devices to maintain connections to the network as well as easily adding and removing devices in the network. User has great flexibility to design such a network at cheapest cost and minimum time. MANETs has shows distinct characteristics, such as:

- Weaker in Security
- Device size limitation
- Battery life
- Dynamic topology
- Bandwidth and slower data transfer rate

Ad hoc routing protocols can be broadly classified as:

- Proactive (table-driven)
- Reactive (on-demand)

Proactive protocols maintained nodes in a MANET and keep track of routes to all possible destinations so that the route is already known and can be immediately used, when a packet needs to be forwarded. On the other hand, reactive protocols employ a lazy approach whereby nodes only discover routes to destinations on demand, i.e., a node does not need a route to a destination until that destination is to be the sink of data packets sent by the node[14].In MANET, mostly we assume that all mobile nodes cooperate fully in the network functionalities. But some nodes decide not to cooperate at all. From Past there is a tendency in the nodes in an ad hoc network to become selfish. The selfish nodes are reluctant to spend their resources such as memory, battery power and CPU time for others but they are not malicious nodes [14]. The problem may become complicated, when with the passage of time the nodes have small amount of residual power and want to conserve it for their own purpose.

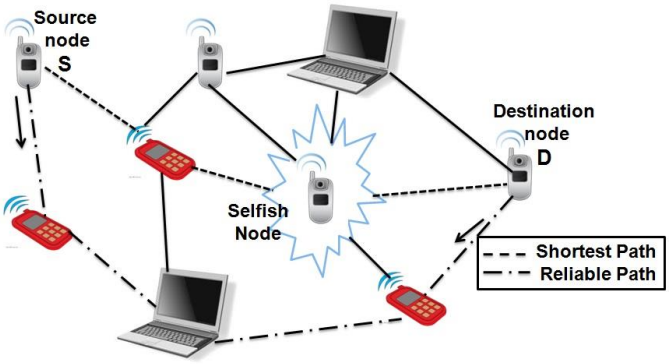

Figure 1. Selfishness in MANET

In general while handling Selfishness in MANET we have to improve data accessibility and reduce query delay, i.e., query Response time, if the mobile nodes in a MANET together have sufficient memory space to hold both all the replicas and the original data. For example, the response time of a query can be substantially reduced, if the query accesses a data item that has a locally stored replica. However, there is often a trade-off between data accessibility and query delay, since most nodes in a MANET have only limited memory space [15]. For example, to reduce its own query delay a node may hold a part of the frequently accessed data items locally. However, if there is only limited memory space and many of the nodes hold the same replica locally, then some data items would be replaced and missing. Thus, the overall data accessibility would be decreased. Hence, to maximize data accessibility, a node should not hold the same replica that is also held by many other nodes [15]. In [1] it defines three types of behavioral states for nodes from the viewpoint of selfish replica allocation:

- **Type-1 node:** are non-selfish nodes i.e. the nodes hold replicas allocated by other nodes within the limits of their memory space.
- **Type-2 node:** called as fully selfish nodes. These nodes do not hold replicas allocated by other nodes, but allocate replicas to other nodes for their accessibility.
- **Type-3 node:** are partially selfish nodes. The nodes use their some part of memory space for allocated replicas by other nodes. Their memory space may be divided logically into two parts as selfish and public area. These nodes allocate replicas to other nodes for their data accessibility.

The detection of the type-3 nodes is complex, because they are not always selfish. In some sense, a type-3 node might be considered as non-selfish, since the node shares part of its memory space here this is considered as (partial) selfish, because the node also leads to the selfish replica allocation problem. Selfish and non-selfish nodes perform the same procedure when they receive a data access request, although they behave differently in using their memory space [1].

II. LITERATURE REVIEW

MANETs rely on the cooperation of all the participating nodes. The more nodes cooperate to transfer traffic, the more powerful a MANET gets. But supporting a MANET is a cost-intensive activity for a mobile node. Detecting routes and forwarding packets consumes local CPU time, memory, network-bandwidth, and last but not least energy [16]. Therefore there is a strong motivation for a node to deny packet forwarding to others, while at the same time using their services to deliver own data. Some resources, namely battery power (energy), are scarce in a mobile environment and can be depleted at fast pace with the device utilization. This can lead to a selfish behavior of the device owner that may attempt to take the benefit from the resources provided by the other nodes without, in return, making available the resources of his own devices. In this scenario, open MANETs will likely resemble social environments. A group of persons can provide benefits to each of its members as long as everyone provides his contribution. For our particular case, each member of a MANET will be called to forward messages and to participate on routing protocols. A selfish behavior threatens the entire

community. Optimal paths may not be available. As a response, other nodes may also start to behave in the same way.

In the [1] existing strategy consists of three parts: 1) detecting selfish nodes, 2) building the SCF-tree, and 3) replica allocation.

The reason is that without forming any group or engaging in lengthy negotiations each node can detect selfish nodes and makes replica allocation at its own discretion [15].

1) Detecting Selfish Node

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

$$\text{Credit Risk} = \frac{\text{expected risk}}{\text{expected value}} \quad (1)$$

i.e., the expected risk is calculated by number of requests served by the node. And the expected value is calculated by number of memory spaces shared by nodes. In the existing strategy, each node calculates a CR score for each of the nodes to which it is connected [1]. The calculated CR value is called as degree of selfishness. Degree of selfishness tells that the node is seems to be Selfish node. Each node shall estimate the selfishness degree for all of its connected nodes based on the CR score. They first describe selfish features that may lead to the selfish replica allocation problem to determine both expected value and expected risk [15].

2) Building SCF-Tree

The SCF Tree build based on human friendship management in the real world, where each person makes their own friends forming a web and manages friendship by their self. They do not have to discuss these with others to maintain the friendship [1]. The decision is only at their discretion. The main goal of the replica allocation techniques discussed is to reduce traffic overhead, achieving data accessibility to maximum level. If this replica allocation technique can allocate replica without considering with other nodes, it will decrease the traffic overhead.

3) Allocating Replica

A node allocates replica at every relocation period, after building the SCF-tree. Within its SCF-tree [1] each node asks non selfish nodes to hold replica when it cannot hold replica in its local memory space. Each node determines replica allocation individually without any communication with other nodes, since the SCF-tree based replica allocation is performed in a fully distributed manner. At first, a node determines the priority for allocating replicas. The priority is based on Breadth First Search (BFS) order of the SCF-tree. The dotted arrow in represents the priority for allocating replica.

III. PROPOSED SYSTEM

Although network issues are important in a MANET, replica allocation is also crucial, since the ultimate goal of using a MANET is to provide data services to users. A selfish node may not share its own memory space to store replica for the benefit of other nodes. We can easily find such cases in a typical peer-to-peer application. A selfish node may not share its own memory space to store replica for the benefit of other nodes. We can easily find such cases in a typical peer-to-peer application. They are based on the concept of a self-centered friendship tree (SCF-tree) and its variation to achieve high data accessibility with low delay in the presence of selfish nodes. We first describe selfish features that may lead to the selfish replica allocation problem to determine both expected value and expected risk. Here we can consider threshold value as main part for detecting selfishness. If node transfers the packet within the threshold value, then it is a non selfish. If exceeds the threshold value, then it is a selfish node.

Credit risk value can be calculated by means of threshold value. For every node ,we can store the data and sent to another node, If there is a selfish node present, then the data will be lost, At that time, Neighboring node will have copy of that data, so we can come one step backward route the data to the another path. As stated in [1] the detection of the type-3 nodes is complex, because they are not always selfish. In some sense, a type-3 node might be considered as non-selfish, since the node shares part of its memory space, due to this here we can consider only two types of nodes as selfish and non-selfish. A selfish node can silently drop some or all of the data packets sent to it for further forwarding even when no congestion occurs. Selfish node attack presents a new threat to wireless ad hoc networks since they lack physical protection and strong access control mechanism. An adversary can easily join the network or capture a mobile node and then starts to disrupt network communication by

silently dropping packets. Selfish node attack is a serious threat to the routing infrastructure of both MANET and the Internet since it is easy to launch and difficult to detect. For this Session Key is used for security purpose.

IV. SIMULATION

Simulations are made using NS-2 [17] simulation program that consists of the collection of all network protocols to simulate many of the existing network topologies. 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 1600X1000 meter flatland area. The movement pattern of nodes follows the random waypoint model, where each node remains stationary for a pause time and then it selects a random destination and moves to the destination [1]. After reaching the destination, it again stops for a pause time and repeats this movement behavior. The radio communication range of each node is a circle with a radius of 1 _ 19 (m). We suppose that there are 50 individual pieces of data, each of the same size. In the network, node N_i ($1 < i < 50$) holds data D_i as the original. The data access frequency is assumed to follow Zipf distribution. And threshold value is considered as 4ns. The default relocation period is set to 256 units of simulation time which we vary from 64 to 8,192 units of simulation time. Nodes were set to use 802.11 radios with 2 Mbps bandwidth and 250 meters nominal range. We considered only static scenario so link breakage due to mobility is zero. The simulated time was 100 seconds. TABLE I describes the simulation parameters.

Table 1. Simulation Parameters

Parameter (unit)	Value (default)
Number of nodes	50
Number of data items	50
Radius of communication range (m)	1 ~ 19 (7)
Size of network (m)	1600 *1000
Percentage of selfish node (%)	0 ~ 100 (70)
Relocation period (ms)	64 ~ 8,192
Threshold (ns)	0 ~ 5 (4)

V. RESULT

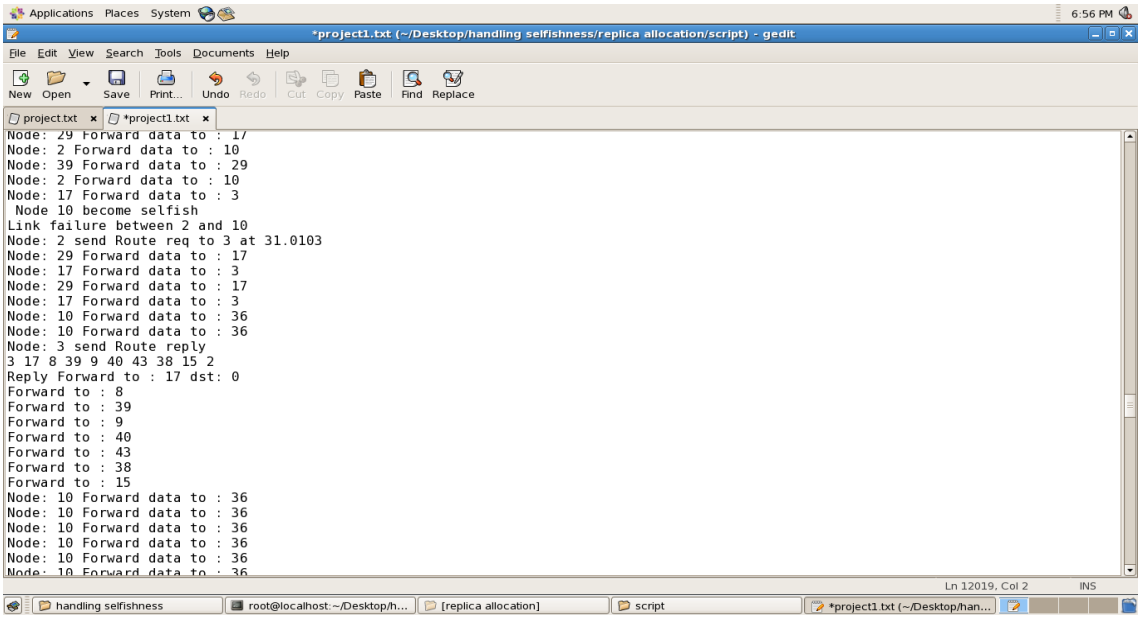

Figure 2. Forwarding Data to handle selfishness

In Figure 2 we have to maintain the data flow, where node 10 becomes selfish. Due to this selfishness behavior packet send over link of node 2-10 gets lost. To handle this situation link cut between these nodes, node 2 send request to node 3 for further data delivery. Hence Selfishness can be handled.
Effect of handling Selfishness over MANET under NS2 simulator is given below.

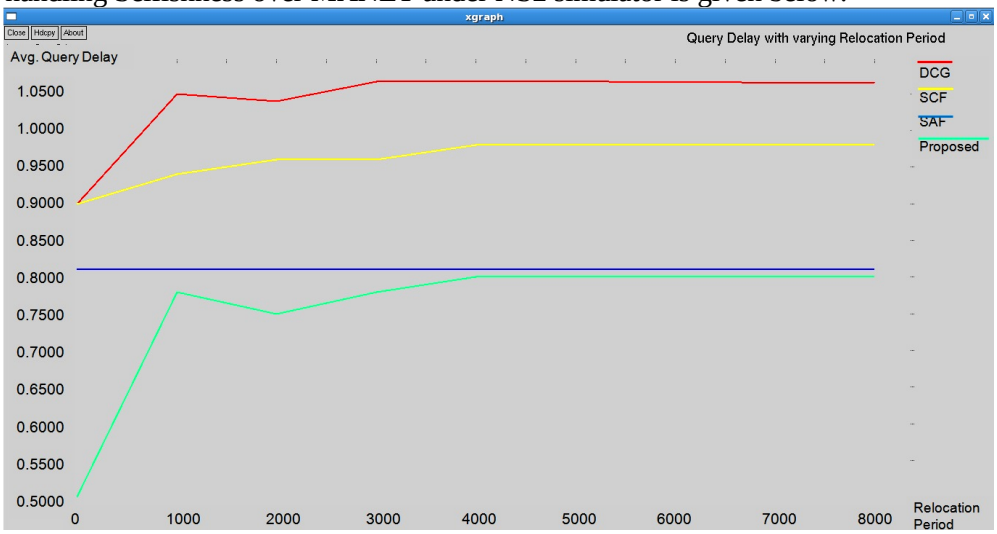

Figure 3. Query delay with respect to relocation period

Figure 3 shows average query delay with respect to relocation period. As expected, Our Proposed technique shows the best performance in terms of query delay, since most successful queries are served by local memory space. While in [1] SAF shows best result which gives minimum delay. Our graph shows better than this as it passes below to SAF technique. In above DCG technique shows the worst performance. This can be explained as follows: the distance in average query delay counts among group members in the DCG technique is longer than that in the SCF technique. Since most successful queries are served by group members in these techniques, the long distance among group members affects query delay negatively [1].

VI. CONCLUSION AND FUTURE WORK

In this paper, we have assumed an environment in which nodes simultaneously issue access requests to correlated data items in MANETs. We have proposed method to handle selfishness condition that is actually extensions of our previously proposed methods to adapt such an environment. The simulation results show that the methods proposed in this paper which give lower query delay than the corresponding methods used in [1]. The results also show that the proposed method is better than DCG, SCF and SAF methods. Here delay is minimum than all this methods.
As part of our future work, we plan to address data replication in an environment where access requests for correlated data items are issued with some intervals. And increase data accessesability. We also plan to extend our proposed methods to adapt data updating.

REFERENCES

[1] Jae-Ho Choi, Kyu-Sun Shim, "Handling Selfishness in Replica Allocation over a Mobile Ad Hoc Network", *IEEE Transactions on Mobile Computing*, vol. 11, no. 2, pp 278-291, Feb 2012.
[2] Takahiro Hara And Sanjay Kumar Madria "Consistency Management Strategies For Data Replication In Mobile Ad Hoc Networks", *IEEE Transactions On Mobile Computing*, Vol. 8, No. 7, pp. 950-967, July 2009.
[3] Nikolaos Laoutaris, Orestis Telelis, Vassilios Zissimopoulos, And Ioannis Stavrakakis "Distributed Selfish Replication" *IEEE Transactions On Parallel And Distributed Systems*, vol. 17, no. 12, pp.1401-1413, Dec 2006.
[4] Takahiro Hara and Sanjay Kumar Madria, "Data Replication for Improving Data Accessibility in Ad Hoc Networks," *IEEE Transaction On Mobile Computing*, vol. 5, no. 11, pp. 1515-1532, Nov. 2006

[5] Alessandro Mei, Luigi V. Mancini, and Sushil Jajodiasecure "Dynamic Fragment And Replica Allocation In Large-Scale Distributed File Systems", *IEEE Transactions On Parallel And Distributed Systems*, Vol. 14, No.9, September 2003.
[6] G. Tamilarasi, Devi Selvam "Allocation of Replicas by Solving Selfishness in MANET", *International Journal of Engineering Research & Technology (IJERT)* Vol. 2 Issue 1, January- 2013 ISSN: 2278-0181.
[7] Jim Solomon , Immanuel John "A Survey on Selfishness Handling In Mobile Ad Hoc Network", *International Journal of Emerging Technology and Advanced Engineering*, Volume 2, Issue 11, November 2012, ISSN 2250-2459.
[8] Shiow-yang Wu, Member, IEEE, and Yu-Tse Chang "A User-Centered Approach to Active Replica Management in Mobile Environments", *IEEE Transactions On Mobile Computing*, Vol. 5, No. 11, November 2006.
[9] Martin Schutte "Detecting Selfish and Malicious Nodes in MANETs", Seminar*: Sicherheit in Selbstorganisierenden Netzen, Hpi/University Potsdam*, sommersemester 2006.
[10] Kashyap Balakrishnan, Jing Deng and Pramod K. Varshney "TWOACK: Preventing Selfishness in Mobile Ad Hoc Networks", *IEEE Conference*, 0-7803-8966-2/05, 2005.
[11] Prasanna Padmanabhan, Le Gruenwald, Anita Vallur and Mohammed Atiquzzaman "A survey of data replication techniques for mobile ad hoc network databases", *The VLDB Journal*, ISSN 1143–1164 May 2008.
[12] Takahiro Hara "Effective Replica Allocation in Ad Hoc Networks for Improving Data Accessibility", *IEEE INFOCOM*, 2001
[13] Y. Hu, A. Perrig and D. Johnson, "Ariadne: A Secure On-demand Routing Protocol for Ad Hoc Networks", in *Proceedings of ACM MOBICOM'02*, 2010.
[14] Devi Selvam and Tamilarasi. G, "Selfish Less Replica Allocation in MANET", *International Journal of Computer Applications, Vol. 63– No.19, pp.33-37 February 2013.*
[15] K.P.Shanmuga Priya and V.Seethalakshmi, "Replica Allocation In Mobile Adhoc Network For Improving Data Accessibility Using SCF-Tree", *International Journal of Modern Engineering Research (IJMER)*, Vol.3, Issue.2, pp-915-919 March-April. 2013.
[16] T.V.P.Sundararajan and Dr.A.Shanmugam, "Performance Analysis of Selfish Node Aware Routing Protocol for Mobile Ad Hoc Networks", *ICGST-CNIR Journal*, Volume 9, Issue 1, July 2009.
[17] "The network simulator - ns2," http://www.isi.edu/nsnam/ns/.

AUTHORS

Madhuri D. Mane presently working P.G. student at the Department of Computer Science & Engineering in MBES College of Engineering, Ambajogai, India. She completed her Bachelor's degree in Computer Science & Engineering Department from M.B.E. society's College of Engineering, Ambajogai under Dr. B.A.M.University, Aurangabad, India. She is pursuing her Master's Degree from the College of Engineering, Ambajogai. Her areas of research interest include Computer Networks &Wireless Network.

B.M. Patil is currently working as a Professor in P.G. Computer Science & Engineering Department in M.B.E. Society's College of Engineering, Ambajogai, India. He received his Bachelor's degree in Computer Engineering from Gulbarga University in 1993, MTech Software Engineering from Mysore University in 1999, and PhD Degree from Indian Institute of Technology, Roorkee, 2011. He has authored several papers in various international journals and conferences of repute. His current research interests include data mining, medical decision support systems, intrusion detection, cloud computing, artificial intelligence, artificial neural network, wireless network and network security.