Selfish Less Replica Allocation in MANET

Devi Selvam
Assistant Professor
Sri Shakthi Institute of Engineering and Technology
Coimbatore, Tamilnadu

Tamilarasi. G
PG scholar
Sri Shakthi Institute of Engineering and Technology
Coimbatore, Tamilnadu

ABSTRACT

Most of the protocols and algorithms used in MANET, are assuming that all mobile nodes cooperate fully with the functionalities of the network. But some nodes are cooperate partially or not at all cooperating. Network performance and data accessibility, accessing time, query delay are affected by these selfish nodes. The discussion of this paper is about, representing a Replica server which solves the selfishness. The Replica server monitors and maintains the status of all mobile nodes in the network. If it finds any selfish node in the shortest path between source and destination, the replica server will analyze the reason for selfishness and it finds solution. The conducted simulations demonstrate the proposed approach based on proxy method which outperforms in terms of network parameters such as accessing data, time and cost and also it improves the network performance of MANET.

Keywords

MANET, data accessibility, accessing time, query delay, selfish nodes, Replica server, proxy method.

1. INTRODUCTION

Mobile ad-hoc networks (MANET) are wireless networks consisting entirely of mobile nodes that communicate on-the-move without base stations. Each node in a MANET acts as router, and communicates with each other nodes in the network. Nodes in mobile ad hoc networks will both generate user and application traffic and carry out network control and routing protocols. Network partitions, changing connectivity rapidly, highest error rates, bandwidth , collision interference, and power constraints together cause new problems in network control particularly in the design of higher level protocols such as routing and in implementing applications with quality of service requirements. There are different types of MANETs including:

- Intelligent vehicular ad hoc networks (InVANETs) – Make use of artificial intelligence to tackle unexpected situations like vehicle collision and accidents.
- Vehicular ad hoc networks (VANETs) – Enables effective communication with another vehicle or helps to communicate with roadside equipments.
- Internet Mobile Ad hoc Networks (iMANET) – helps to link fixed as well as mobile nodes.

A MANET environment wants to solve certain issues of limitation and less efficiency which includes:

- Time-varying wireless link characteristics in nature.
- Wireless transmission is limited range.
- An error in transmission causes packet loss.
- Frequent path breaks due to dynamic nature of network topology.
- Intermediate nodes are affected by the random movement of nodes which often leads to network partitioning.

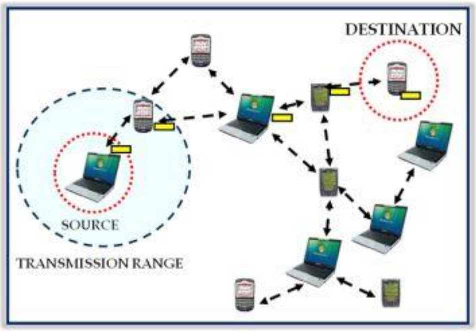


Fig 1 Mobile Ad hoc Network

Routing in a MANET is intrinsically different from traditional routing found on wired networks. MANET routing depends on many factors including router selection, topology and type of request initiation, and specific underlying characteristic that could serve as a heuristic in finding the path quickly and efficiently.

The low resource availability in these networks demands efficient utilization and hence the motivation for optimal routing in ad hoc networks. Also, high dynamic nature of these networks imposes severe restrictions on protocols. One of the major challenges in ad hoc networks stems for designing a routing protocol from the fact that a node needs to know at least the network topology can change quite often in an ad hoc network and on the other hand the reachability information to its neighbors for determining a packet route and, on the other hand,

Ad hoc routing protocols can be broadly classified as being Proactive (or table-driven) or Reactive (on-demand). Proactive protocols mandate that nodes in a MANET should 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.

Data are usually replicated at nodes, to increase data accessibility to cope with frequent network partitions, other than the original owners. MANETs are very popular solution in the situation where network infrastructure is not available. The replication system duplicates and maintains the consistency of multiple copies of objects in different sites so that each client node can visit a local copy of an object instead of remote ones. In this way, replication can significantly improve a distributed system's availability, reliability and scalability.

In general, a good replication management technique for MANETs should be efficient to deliver requested data items from the neighbors node and capable to decide which data items can be replicated at a node. Further there should be a replica replacement algorithm to replace the old copy of data items when there is an update in the original copy of the data item.

In MANET, most of the Replica allocation techniques are assuming that all mobile nodes cooperate fully in the network functionalities. But some nodes decide to cooperate partially or not at all. Network performance and data accessibility are affected by these selfish nodes. The time passes 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. The problem is especially complicated, when with the passage of time the nodes have little residual power and want to conserve it for their own purpose.

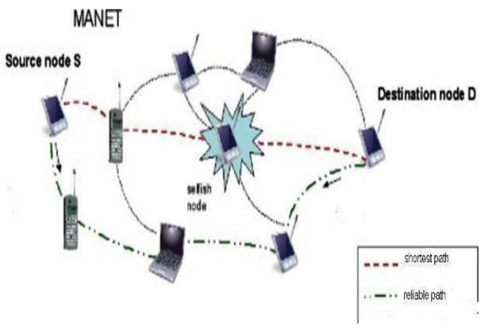


Fig 2 Selfishness in MANET

Thus in MANET environment there is a strong motivation for a node to become selfish. The characteristics of selfish nodes as follows:

- A selfish node drops routing messages
- A selfish node may modify the Route Request
- A selfish node may reply packets by changing TTL value to smallest possible value.
- A selfish node may not send response to hello messages; hence other nodes may not be able to detect its presence when they need it.
- A selfish node may delay the RREQ packet up to the maximum upper limit time.
- A selfish node will certainly avoid itself from routing paths.
- A selfish node may participate in routing messages but may not relay data packets.
- A selfish node may affect the replication in network.

2. PERFORMANCE OF NODES

It is necessary to further consider the partial selfish behavior to handle the selfish replica allocation. Therefore, the node is classified into define three types of behavioral states for nodes from the viewpoint of selfish replica allocation.

- *Non-selfish node:* The nodes hold replicas allocated by other nodes within the limits of their memory space.
- *Fully selfish node*: The nodes do not hold replicas allocated by other nodes, but allocate replicas to other nodes for their accessibility.
- *Partially selfish node:* Their memory space may be divided logically into two parts: selfish and public area. These nodes use their memory space partially for allocated replicas by other nodes for improving their data accessibility.

The identification of the partially selfish nodes is a tedious work, because they are not always behaving selfishly. In some situation, partially selfish node may also be considered as non-selfish nodes, because the node shares part of its memory space. Also note that selfish and non-selfish nodes perform they behave differently in using their memory space and they use same procedure when they receive a data access request.

3. TYPE EXISTING SYSTEM BY USING SCF TREE AND CR VALUE

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.

3.1 Detecting Selfish Node

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

Credit Risk =expected risk / expected value

i.e., the expected risk is calculated by number of requests nit served by the node. And the expected value is calculated by number of memory spaces shared. In the existing strategy, each node calculates a CR score [1] for each of the nodes to which it is connected. The calculated CR value is known as degree of selfishness. Degree of selfishness is high means, 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.

3.2 Building SCF-Tree

It was 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 solely at their discretion. The main goal of the replica allocation techniques are reducing traffic overhead, achieving data accessibility to maximum level. If this replica allocation technique can allocate replica without considering with other nodes, as in a human friendship management, it will decrease the traffic overhead.

3.3 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.

4. PROPOSED SYSTEM BY USING REPLICA SERVER

In the existing strategies [2][4][8][13] there is still having a problem of selfish nodes which creates problem in accessing data and slow down the network performance. And also they are considering partial selfish nodes as selfish nodes which may not create problem sometimes so there may be an inconvenience. Also there is no server or control to monitor the replica allocation of nodes. The major disadvantage is that if any node become selfish to protect their resources there is no way to identify that selfish node. To overcome these disadvantages the following technical contribution of the paper is used.

- Designing replica server
- Monitoring nodes
- Identifying the selfishness
- Rectifying the selfishness

4.1 Designing Replica Server

The In MANET, all the nodes are handling data and they are having the dynamic counter value. The counter value is dynamic. So the size of the counter is changing dynamically. Each node will have their own counter. So the main functionality of the mobile nodes is that,

- Transmitting data
- Updating the counter value

The disadvantages of the existing strategy are solved by using the proxy replica server. The server will keep on monitoring the nodes which are allocated to that particular server and it will check whether the node is transmitting data or not. If the server finds that any node is not transmitting the data or in the idle state the server will check the counter value. The counter is overflowed means the server identifies that the node cannot transmit the data. It will maintain the previous or past value of the counter of the each node. If the counter value remains same means the server can know that the node behaves selfishly.

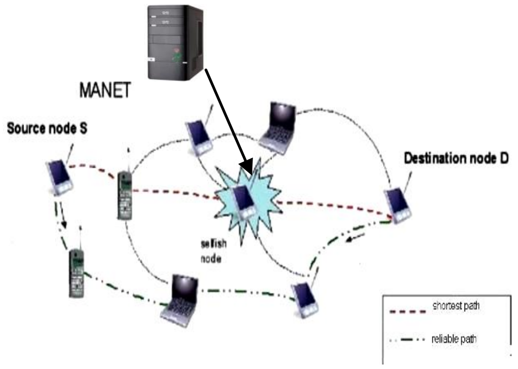


Fig 3 Proxy Replica Server in MANET

After that the server will refresh or clear the particular mobile node’s counter value and helps to the nodes to transmit the data. So the main functionalities of the replica server,

- Allocating dynamic counter for each node
- Monitoring the status of counter value
- It maintains the status of the each node
- Finding selfishness of the nodes
- Refreshing the selfish node counter value

The above mentioned functionalities are only carrying by the server. Only functionality of the node is that it will update the counter value after sending the data. So the nodes will not get any functional overhead. The server will refresh the counter only it finds that there is no data transmission. So the network performance will not be affected by the functions of the replica server.

4.2 Monitoring nodes

The monitoring is the process of supervising the counter value and data transmission of the nodes in the mobile ad hoc network. The intrusion detection system (IDS) for mobile ad hoc networks (MANET) consists in monitoring the nodes’ behavior, in order to detect the activity of nodes which behaving maliciously. The replica server will overhear the data transmission of the network. And also it will check the counter value of the each node in the network.

The previous status of each node i.e., the counter value of the nodes is maintained in the server. The table is called as status table. By using the status table the replica server can easily monitor and compare it with the previous values. Monitoring can be done by several ways. Here referring the following two ways,

4.2.1 Mobile Agent

Intrusion Detection System (IDS) based on Mobile Agents. The approach uses a set of Mobile Agent (MA) that can move from one node to another node within a network [7]. This as a whole reduces network bandwidth consumption by moving the computation for data analysis to the location of the intrusion. The Mobile Agent maintains the following table to perform the computation and comparison with threshold value

Server node ID	Destination Node ID	HOP count	Counter

The table contains the Server node ID, destination node id that will be initiated by the source node i.e., server node. The HOP count field in the table denotes number of HOP between the source node and destination node. Counter value signifies the value of counter value of each destination node. The forward path is generated by any routing algorithm. Besides, it has been established that the proposed method also decreases the computation overhead in each node in the network.

4.2.2 Watch Dog

Watchdogs are used to detect selfish nodes in computer networks these are initiated by Replica server. A way to reduce the detection time and to improve the accuracy of watchdogs is the collaborative approach [6]. A collaborative watchdog based on contact dissemination of the detected selfish nodes. Although some of the aforementioned papers introduced some degree of collaboration on their watchdog schemes, the diffusion was very costly (usually based on sending periodic messages). If one server node has previously detected a selfish node using its watchdog it can spread this information to other nodes. Formally, there having a network of *N* wireless mobile nodes, with *C* collaborative nodes and *S* selfish nodes. Initially, the collaborative nodes have no information about the selfish nodes. A collaborative node can have a positive when a contact occurs in the following way:

- *Contact with Selfishness*: One of the nodes is the selfish node. Then, the collaborative node [6] can detect it using its watchdog and have a positive about this selfish node.
- *Contact Collaboration:* If two nodes are collaborative that a node has a positive if it knows the selfish node [6]. 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. As in the selfish contact case, a contact does not always imply collaboration.

The watch dog will collect the information and returns to the server. The information will contain counter value and address of the selfish node. The server will update the status table by using that information.

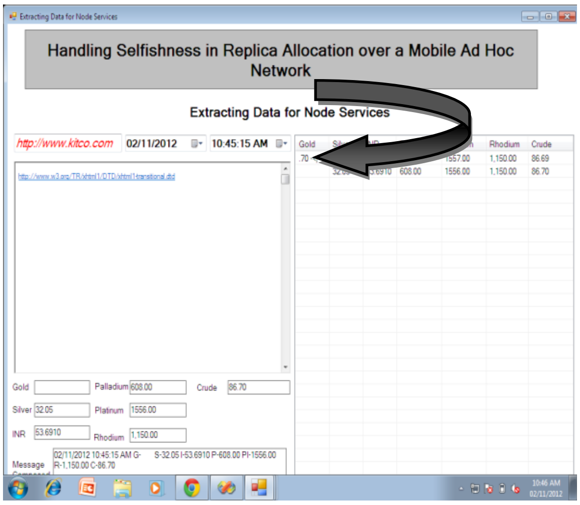
4.3 Identifying the Selfish node

The counter value is monitoring by the replica server and also status of data transmission in the network. The node can update the counter value after transmitting the data otherwise the counter value will be the same. Fig 4 shows the simulation about identification of selfishness in MANET. So the identification of selfish nodes in the MANET will be in the following ways:

- If any node is not participating in data transmission, that can be identified by the mobile agent or watch dog means the server can identify that there is selfishness occurred in that node.
- If the counter value is same as in the status table means the server can identify that the node is behaving selfishly.
- If the counter size is exceeded or it is full, in this case also the server can identify the selfishness of the node.

4.4 Rectifying the selfishness

The server finds selfish node by using mobile agent or watch dog. After finding the selfish node the replica server will decide the rectification of selfishness.



Extracting Data for Node Services

Handling Selfishness in Replica Allocation over a Mobile Ad Hoc Network

Extracting Data for Node Services

http://www.kitco.com 02/11/2012 10:45:15 AM

http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd

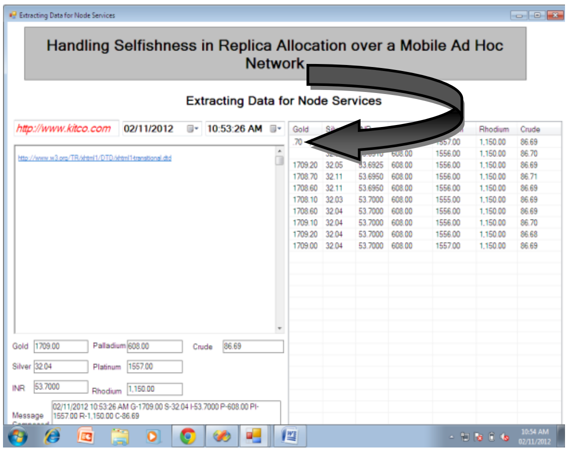
Gold	[illegible]	[illegible]	[illegible]	[illegible]	Rhodium	Crude
.70	[illegible]	[illegible]	[illegible]	1557.00	1,150.00	86.69
[illegible]	32.05	[illegible]	608.00	1556.00	1,150.00	86.70

Gold
Palladium 608.00
Crude 86.70
Silver 32.05
Platinum 1556.00
INR 53.6910
Rhodium 1,150.00
Message 02/11/2012 10:45:15 AM G- S-32.05 I-53.6910 P-608.00 PI-1556.00 R-1,150.00 C-86.70

Fig 4 Identifying Selfishness in MANET

Fig 5 shows the simulation about rectification of selfishness in MANET. For rectification,

- The server will send the signal to that particular selfish node in order to allow the nodes in the shortest path.
- It will refresh counter i.e., it clear the counter value so that the selfishness can be removed.



Extracting Data for Node Services

Handling Selfishness in Replica Allocation over a Mobile Ad Hoc Network

Extracting Data for Node Services

http://www.kitco.com 02/11/2012 10:53:26 AM

http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd

Gold	[illegible]	[illegible]	[illegible]	[illegible]	Rhodium	Crude
.70	[illegible]	[illegible]	[illegible]	1557.00	1,150.00	86.69
[illegible]	[illegible]	[illegible]	608.00	1556.00	1,150.00	86.70
1709.20	32.05	53.6925	608.00	1556.00	1,150.00	86.69
1708.70	32.11	53.6950	608.00	1556.00	1,150.00	86.71
1708.60	32.11	53.6950	608.00	1556.00	1,150.00	86.69
1708.10	32.03	53.7000	608.00	1555.00	1,150.00	86.69
1708.60	32.04	53.7000	608.00	1556.00	1,150.00	86.69
1709.10	32.04	53.7000	608.00	1556.00	1,150.00	86.70
1709.20	32.04	53.7000	608.00	1556.00	1,150.00	86.68
1709.00	32.04	53.7000	608.00	1557.00	1,150.00	86.69

Gold 1709.00
Palladium 608.00
Crude 86.69
Silver 32.04
Platinum 1557.00
INR 53.7000
Rhodium 1,150.00
Message 02/11/2012 10:53:26 AM G-1709.00 S-32.04 I-53.7000 P-608.00 PI-1557.00 R-1,150.00 C-86.69

Fig 5 Rectifying Selfishness in MANET

5. CONCLUSION

In contrast to the MANET viewpoint, this paper has addressed the problem of selfish nodes from the replica allocation perspective. This paper terms this problem selfish replica allocation. The idea was motivated by the fact that a selfish replica allocation will lead to poor data accessibility in a MANET as overall. The idea has proposed a selfish node detection method and method to solve selfishness to handle the selfish replica allocation appropriately. By using Proxy replica server the selfishness of MANET nodes can be removed. This proposed system is capable of handling selfishness in small size network. Based on the server's capacity the selfishness can be handled by the server. The current works are related for different mobility patterns and improving the scalability of proposed system. The future plan is to identify and handle false alarms in selfish replica allocation. False alarm is the problem that the nodes are not transmitted to the destination not because of selfishness. The failure will occur due to the network failure.

6. REFERENCES

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

[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] Shailender Gupta, C. K. Nagpal and Charu Singla, "IMPACT OF SELFISH NODE CONCENTRATION IN MANETS" International Journal of Wireless & Mobile Networks (IJWMN) Vol. 3, No. 2, April 2011.

[4] Amit Saxena, J.L Rana, "Analysis of Selfish and Malicious Nodes on DSR Based Ocean Protocol in MANET", TECHNIA International Journal of Computing Science and Communication Technologies, VOL. 3, NO. 1, July 2010. (ISSN 0974-3375).

[5] Yang Zhang, Student Member, IEEE, Liangzhong Yin, Jing Zhao, and Guohong Cao, Fellow, IEEE, “Balancing the Tradeoffs between Query Delay and Data Availability in MANETs”, IEEE Transactions On Parallel And Distributed Systems, Vol. X, No. X, January 20xx.

[6] Enrique Hern´andez-Orallo, Manuel D. Serrat, Juan-Carlos Cano, Carlos T. Calafate, and Pietro Manzoni, “Improving Selfish Node Detection in MANETs Using a Collaborative Watchdog”, IEEE COMMUNICATIONS LETTERS, VOL. 16, NO. 5, MAY 2012.

[7] Debdutta Barman Roy and Rituparna Chaki, “MADSN: Mobile Agent Based Detection of Selfish Node in MANET”, International Journal of Wireless & Mobile Networks (IJWMN) Vol. 3, No. 4, August 2011.

[8] S. Usha, *Member, IACSIT* and S. Radha, “Multi Hop Acknowledgement Scheme based Selfish Node Multi Hop Acknowledgement Scheme based Selfish Node”, International Journal of Computer and Electrical Engineering, Vol. 3, No. 4, August 2011.

[9] Rajeev Kumar, and Prashant Kumar, “Replica Allocation Technique Based on Clusters for MANETs” International Conference on Emerging Trends in Computer and Electronics Engineering (ICETCEE'2012) March 24-25, 2012 Dubai.

[10] Rashid Azeem and Muhammad Ishfaq Ahmad Khan, “Techniques about Data Replication for Mobile Ad-hoc Network Databases”, International Journal Of Multidisciplinary Sciences And Engineering, Vol. 3, No. 5, May 2012.

[11] Hongxun Liu, José G. Delgado-Frias, And Sirisha Medidi, “Using A Two-Timer Scheme To Detect Selfish Nodes In Mobile Ad-Hoc Networks”, proceedings of the sixth IASTED.

[12] Zaiba Ishrat, “Security issues, challenges & solution in MANET”, IJCST Vol. 2, Iss ue 4, Oct . - Dec. 2011 ISSN : 0976-8491 (Online) | ISSN : 2229-4333(Print).

[13] Dipali Koshti, Supriya Kamoji, “Comparative study of Techniques used for Detection of Selfish Nodes in Mobile Ad hoc Networks”, International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-4, September 2011.