An Agent Based Approach to Avoid Selfish Node Dynamically in Mobile Networks

Radhika Garg[1], Sanjay Kumar[2], Sangeeta Malik[3], Deepak Goyal[4], Dr. Pankaj Gupta[5]

[1] M.Tech Student, Vaish College of Engineering, MDU, Rohtak
rockingradhikagarg@gmail.com

[2]Asst. Prof., Vaish College of Engineering, MDU, Rohtak

[3]Asst. Prof., Vaish College of Engineering, MDU, Rohtak

[4]Asso. Prof., Vaish College of Engineering, MDU, Rohtak

[5]Prof. Vaish College of Engineering, MDU, Rohtak

Abstract

Mobile Networks is one of the busy networks on which lot of data is transferred at very high speed. Security and efficiency are the main challenges for such open network. One of the common attacks on such network is the misbehavior of a node as a **Selfish Node**. A selfish node itself utilizes the communication medium and will not help in forwarding the packet. The proposed work is the agent based analysis of network. For this work, an **Agent** is setup which will perform the analysis while communicating over the network. The agent will observe the average communication of each node and based on analysis, a dynamic fuzzy rule will be decided. Now each node transmission will be checked on this fuzzy rule. A specific rule will be decided to identify the selfish node. The work is about to decide the compromising node that will replace the selfish node to improve the throughput over the network.

Keywords: *Mobile Network, Central Coordinator, Selfish Node, Agent.*

Introduction

A Mobile ad hoc network is a group of wireless mobile computers (or nodes)[1]. In which nodes collaborate by forwarding packets for each other to allow them to communicate outside range of direct wireless transmission. Ad hoc networks [2] require no centralized administration or fixed network infrastructure such as base stations or access points, and can be quickly and inexpensively set up as needed.

System Architecture(C. Siva Ram Murthy et al, 2004):

In our architecture, one or more pre-defined nodes act as a *group controller* (GC), which is trusted by all the group nodes. A GC has authority to assign resources to the nodes in MANET. This resource allocation is represented as a Key Note style credential (capability) called *policy token*, and it can be used to express the services and the bandwidth a node is allowed to access. They are cryptographically signed by the GC, which can be verified any node in the MANET. When a node (initiator) requests a service from another MANET [1] node (responder) using the policy token assigned to the initiator, the responder can provide a capability back to the initiator. This is called a *network capability*, and it is generated based on the resource policy assigned to the responder and its dynamic conditions. Figure gives a brief overview of our system. All nodes in the path between an initiator to a responder (*i.e.*, nodes relaying the packets) enforce and abide by the resource allocation encoded by the GC in the policy token and the responder in the network capability. The enforcement involves both accessibility and bandwidth allocation. A responder accepts packets (except for the first one) from an initiator only if the initiator has authorization to send, in the form of a valid network capability. It accepts the first packet only if the initiator's policy token is included. An intermediate node will forward the packets from a node only if the packets have an associated policy token or network capability, and if they do not violate the conditions contained therein. Possession of a network capability does not imply resource reservation; they are the maximum limits a node can use. Available resources are allocated by the intermediate nodes in a fair manner, in proportion to the allocations defined in the policy token and network capability.The capability need not be contained in all packets. The first packet carries the capability, along with a transaction identifier (TXI) and a public key. Subsequent packets contain only the TXI and a packet signature based on that public key. Intermediate nodes cache policy tokens and network capabilities in a *capability database*, treating them as soft state.

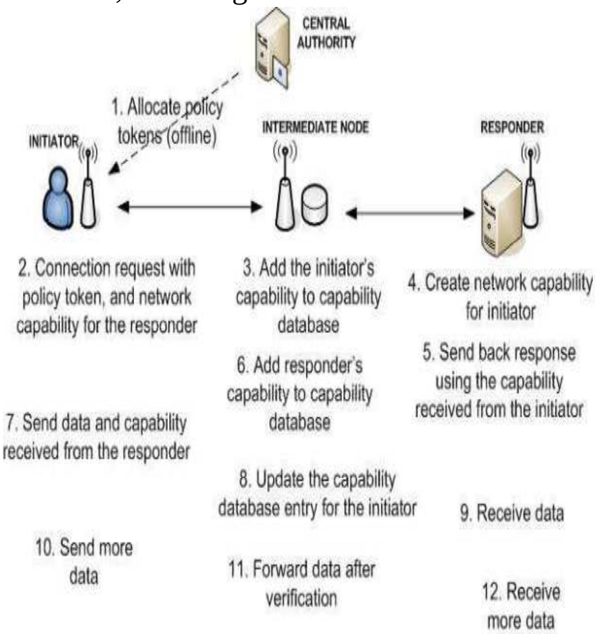

CENTRAL AUTHORITY
1. Allocate policy tokens (offline)
INITIATOR
INTERMEDIATE NODE
RESPONDER
2. Connection request with policy token, and network capability for the responder
3. Add the initiator's capability to capability database
4. Create network capability for initiator
5. Send back response using the capability received from the initiator
6. Add responder's capability to capability database
7. Send data and capability received from the responder
8. Update the capability database entry for the initiator
9. Receive data
10. Send more data
11. Forward data after verification
12. Receive more data

Figure (C. Siva Ram Murthy et al, 2004)

Mobile ad hoc network nodes are furnished with wireless transmitters and receivers using antennas, which may be highly directional (point-to-point), omnidirectional (broadcast), probably steer able.

1. Review of Literature

In Year 2005[3], Yongwei Wang performed a work,**" A Light-weight Solution for Selfish Nodes Problem Considering Battery Status in Wireless Ad-hoc Networks".** In this paper, Author presents a fully distributed solution to force nodes' cooperation. The solution is light-weight in that neighbor monitoring is on-demand, and nodes work in promiscuous mode only part time to save battery. Besides, Author provides fairness to nodes in the network by considering their battery status.

In Year 2007[4], CenkerDemir performed a work,**" An Auction based AODV Protocol for Mobile Ad Hoc Networks with Selfish Nodes".** In this paper, Author proposes an auction mechanism for routing in mobile ad hoc networks with selfish nodes. Presented approach is to promote bidding for end-to-end routes, as opposed to node-by-node bidding, to avoid wasting the source's resources (currency, time and data) by possibly losing a bid at an intermediate node.

In Year 2008[6], R.Gunasekaranperformed a work**,"Detection and Prevention of Selfish and Misbehaving Nodes at Mac Layer in Mobile AD HOC Networks".** In wireless networks all nodes contending to access the medium are supposed to follow the rules of the Medium Access Control (MAC) layer. Wireless Medium Access Control (MAC) protocols such as IEEE 802.11 use a distributed contention resolution mechanism for sharing the wireless channel. The hosts competing for access to the channel are required to wait for a “back off” interval, randomly selected from a specified range, before initiating a transmission. Selfish nodes (or misbehaving nodes) tempt to manipulate their back off parameters to gain more access to the channel, and hence have higher performance than their fair share.

In Year 2010[7]Fahad T. Bin Muhaya performed a work,**” Selfish Node Detection in Wireless Mesh Networks”.** Wireless Mesh Networks (WMNs) are multi-hop network in which each node communicates with each other to increase the performance of network. Security is a big challenge in WMNs. The network always faces different types of security attacks by external and internal intruders. Most of the time these intruders are the internal legitimate nodes of the network that behave abnormally and network becomes unsecure.

In Year 2011[10], Chi Lin performed a work,**” A Selfish Node Preventive Real Time Fault Tolerant Routing Protocol for WSNs”.** Selfish behaviors of nodes in real-time wireless sensor networks can cause massive packet loss or even VOID regions, a game theory based real time fault tolerant routing protocol, namely GTRF, is proposed in this paper.

2. Proposed Work

In the mobile network, thenodes participate in packet transmission [1]. A central coordinator calculates loss rate of all the nodes on network. Packet Loss[10] can occur due to congestion in network or due to selfish behavior of node. To detect the selfish node[6][7], which doesn’t forward the packets, a fuzzy rule is implemented. The implemented fuzzy rule is:if loss rate is less than 0.7 then this loss is bearable. But if it is greater than 0.7 then it shows node’s selfishness. Due to Selfish behavior of Node, the network throughput is decreased [5][8]. After this, Agent comes into role. The central coordinator assignsone of neighboring node of selfish node as an Agent. Now agent applies the algorithm to detect [9] compromising node to transfer selfish node’s load. For this, it shares the routing table of its neighboring nodes and finds its status. After this the route[4]breaks up at selfish node and passes to the detected compromising node. Then agent will continually monitor the selfish node[3] and when selfish node stops its selfish behavior, communication is started on reconstructed network.

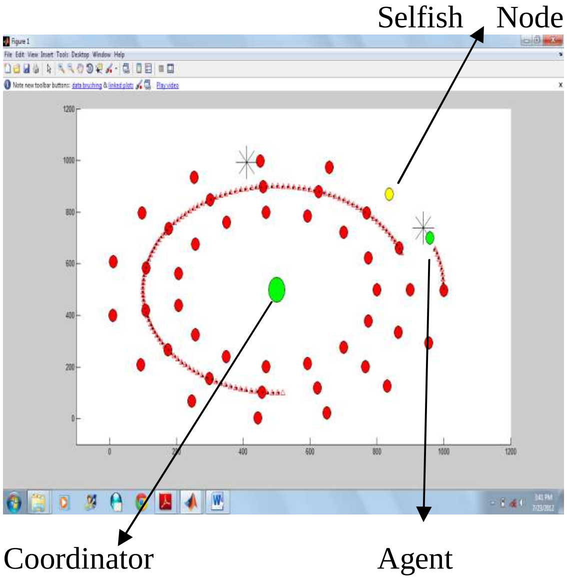
Procedure Followed:

1. **Central Coordinator** calculates packet loss rate of every node on Network.
2. If **Loss Rate**< 0.7 then loss is bearable. It may be due to some network problem like congestion.
3. If **Loss Rate**> 0.7 then the corresponding node is detected as Selfish Node.
4. Central coordinator assigns one of neighboring node of Selfish node as an **AGENT**.
5. Then Agent applies following **algorithm** to find **compromising node** on Network:

    i. Share Routing Table of all its neighboring nodes.
    ii. Find Status of nodes using routing table.
    iii. Transfer the load of selfish node on one of the compromising neighboring node with least load.

6. After then, the Selfish Node is excluded from route and passed to compromising node.
7. Agent continually **monitors** the selfish node by sending packets.
8. If at any time, when Selfish Node leaves its selfish behavior, the original route is reconstructed.

After identification of selfish node route is changed to deliver packet.


Selfish Node
Figure 1
File Edit View Insert Tools Desktop Window Help
Note new toolbar buttons: data brushing & linked plots Play video
1200
1000
800
600
400
200
0
0
200
400
600
800
1000
1200
Coordinator
Agent

Loss Rate is more than 0.7, therefore selfish node is detected.

Selfish Node stops its selfish behavior and route is repaired.

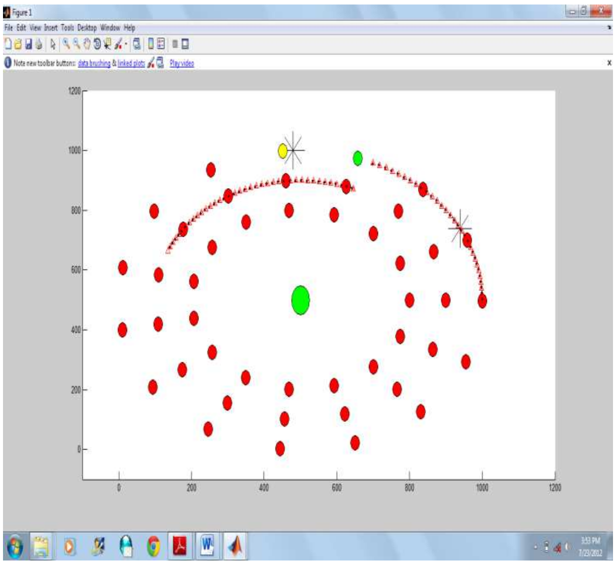
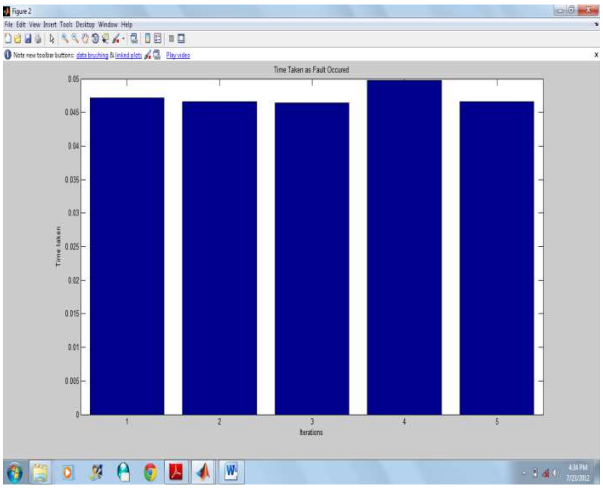


Figure 1
File Edit View Insert Tools Desktop Window Help
Note new toolbar buttons: data brushing & linked plots Play video
1200
1000
800
600
400
200
0
0
200
400
600
800
1000
1200

Agent continually monitors the selfish node by sending packets to it. At any time, if agent detects that selfish node responds correctly, the network is reconstructed by including selfish node into route again.

Analysis of Result



Comparison: Average Elapsed time in our approach is 0.045017 seconds. In another approach in which direction of packet transmission is reversed after detecting selfish node, the Elapsed Time is 1.729893 seconds. Hence our approach is better than other.

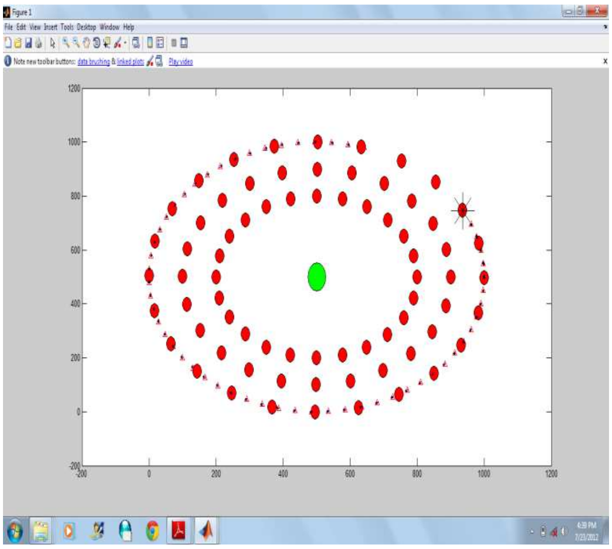


Figure 1
File Edit View Insert Tools Desktop Window Help
Note new toolbar buttons: data brushing & linked plots Play video
1200
1000
800
600
400
200
0
-200
-200
0
200
400
600
800
1000
1200

3. Conclusion

- We can set up number of Agents in network to increase efficiency of network but incurred cost would increase with number of Agents.
- After analysis, we have observed that the elapsed time in successful packet delivery in Agent Based Approach is lesser than time elapsed in other approach.

4. References

[1]Young-BaeKo and Nitin H. Vaidya “Location-Aided Routing (LAR) in mobile ad hoc networks”, Wireless Networks 6 (2000) 307–321.

[2]Ram Ramanathan and Jason Redi, “A brief overview of Ad-hoc Networks: Challenges and Directions”, IEEE Communications Magazine May 2002, pp. 20-22.

[3]Yongwei Wang,” A Light-weight Solution for Selfish Nodes Problem Considering Battery Status in Wireless Ad-hoc Networks”, 0-7803-9182-9/05©2005 IEEE

[4]CenkerDemir,” An Auction based AODV Protocol for Mobile Ad Hoc Networks with Selfish Nodes”, ICC 20071-4244-0353-7/07©2007 IEEE

[5]Ramakant S. Komali,” Effect of Selfish Node Behavior on Efficient Topology Design”, IEEE TRANSACTIONS ON MOBILE COMPUTING 1536-1233/08@ 2008 IEEE

[6]R.Gunasekaran,”Detection and Prevention of Selfish And Misbehaving Nodes at Mac Layer in Mobile AD HOC Networks”, 978-1-4244-1643-1/08© 2008 IEEE

[7]Fahad T. Bin Muhaya,” Selfish Node Detection in Wireless Mesh Networks”, 201O International Conference on Networking and Information Technology 978-1-4244-7578-0© 2010 IEEE

[8]JaydipSen,” An Efiicient Algorithm for Detection of Selfish Packet Dropping Nodes in Wireless Mesh Networks”, 978-1-4244-7818-7/10@ 2010 IEEE

[9]Lien-Wen Wu,” A Threshold-Based Method for Selfish Nodes Detection in MANET”, 978-1-4244-7640-4/10©2010 IEEE

[10] Chi Lin,”A Selfish Node Preventive Real Time Fault Tolerant Routing Protocol for WSNs”, 2011 IEEE International Conferences on Internet of Things, and Cyber, Physical and Social Computing 978-0-7695-4580-6/11© 2011 IEEE