INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

NEIGHBORING NODE BASED SYSTEM FOR MANET TO DETECT SELFISH NODES USING NS2

SAGAR PADIYA[1], RAKESH PANDIT[2], SACHIN PATEL[3], TANUJ ROHANKAR[4]

1. M. Tech (IT) Scholar, Patel College of Science & Tech., RGPV, Bhopal, Indore, India.
2. Assistant Professor, Department of IT, Patel College of Science & Tech, Indore, India.
3. HOD, Department of IT, Patel College of Science & Tech, Indore, RGPV, Bhopal, India.
4. Assistant Professor, Department of IT, Sipna College of Engg. & Tech, Amravati India.

Accepted Date:

27/02/2013

Publish Date:

01/04/2013

Keywords

Network Simulator,

Mobile Ad-hoc Network (MANET) area;

Nodes;

Selfish Nodes,

Misbehaving Nodes.

Corresponding Author

Mr. Sagar Padiya

Abstract

An Ad-hoc network is a collection of mobile nodes dynamically forming a temporary network without the use of any existing network infrastructure or centralized administer. Because of limited communication range among MANET, several network hopes may be needed to deliver a packet from one node to another node in the wireless network. In such a network each node acts as an end system as well as a relay node (or router). Most of the routing algorithms designed for MANET such as AODV and DSR are based on the assumption that every node forwards every packet [01]. But in practice some of the nodes may act as the selfish nodes. These nodes use the network and its services but they do not cooperate with other nodes. Such selfish nodes do not consume any energy such as CPU power, battery and also bandwidth for retransmitting the data of other nodes and they reserve them only for themselves. The original AODV and DSR routing algorithms can be modified to detect such selfish nodes [07]. In this paper, we present a new neighboring node based mechanism to detect those selfish nodes in MANET and perform the simulation using Network Simulator-2(Version-2.32). Each node is expected to contribute to the network on the continual basis within a time frame. Those which fail will undergo a test for their suspicious behavior. Simulation results show that our neighboring node based system could be used to detect selfish nodes in a network.

INTRODUCTION

Mobile ad hoc network is a network consisting of mobile nodes (Laptop, Personal Digital Assistants (PDAs) and wireless phones) with the characteristics of self-organization and self-configuration which enable it to form a new network quickly [01]. A Mobile Ad hoc Network or in short, MANET, is a relatively new communication paradigm. A MANET network consists of a group of mobile devices (nodes) communicating through a wireless medium. Unlike a traditional infrastructure network, the network is established solely by the MANET devices themselves without the need of any fixed infrastructure such as an access point or base station. A node may be able to communicate with other nodes far away with the cooperation of intermediate nodes, forwarding the packets to the destination. In this multi hop communication, each node operates as both host and router.

Routing protocol such as Dynamic Source Routing [DSR] and AODV have been designed to handle such environment [02]. Minimal configuration, quick deployment and the absence of central governing authority make MANET suitable for emergency situations such as natural disasters, military conflicts and emergency medical situations. However, since there is no centralized administration, the performance of a MANET greatly depends on the cooperation of all nodes in the network.

A MANET is a self-configuring system of mobile nodes connected by wireless links. In a MANET, the nodes are free to move randomly, changing the networks topology rapidly and unpredictably. MANETs are decentralized, and therefore all network activities are carried out by nodes themselves.

In this paper, we propose and simulate a new neighboring node based system to detect selfish nodes that refuse to cooperate but at the same time still use the network for their own benefits. The rest of the paper is organized as follows. In Section-II, we introduced the DSR (Dynamic Source Routing) protocol which will use in our simulation. In Section-III, we categorized two types of node misbehavior in a MANET. In Section-IV, we briefly summarized the various innovated techniques for node misbehavior detection. In Section-V, we proposed our new neighboring node based system to detect selfish nodes. In Section-VI, we describe simulation tools and review our simulation parameters, environment and system for simulation. In section-VII, we shows result after the performance of our mechanism using NS2. In section-VIII, we conclude the work.

DYNAMIC SOURCE ROUTING

The performance of this algorithm was very good at all mobility rates and movement speeds, although its use of source routing increases the number of routing overhead bytes required by the protocol.

DSR [03, 04, 05] uses source routing rather than hop-by-hop routing, with each packet to be routed carrying in its header the complete, ordered list of nodes through which the packet must pass. The key advantage of source routing is that intermediate nodes do not need to maintain up-to-date routing information in order to route the packets they forward, since the packets themselves already contain all the routing decisions. This fact, coupled with the on-demand nature of the protocol, eliminates the need for the periodic route advertisement and neighbour detection packets present in other protocols.

Basic Mechanism:

Dynamic Source Routing (DSR) is one of routing protocols proposed within the Mobile Ad Hoc Networks (MANET) working group of the Internet Engineering Task Force (IETF) [2]. The protocol is divided into two main mechanisms:

1) Route Discovery and

2) Route Maintenance,

Both of these are operate entirely on-demand. A source node which wishes to formed communication with a destination node. Therefore it will first search its own route cache table. If no route to the destination is found, it will initiate Route Discovery by broadcasting a RREQ (Route Request) packet to its neighbors. Each intermediate node receiving the RREQ, and adds its address to the RREQ and then rebroadcast the modified RREQ.

If the destination node receives the RREQ, it constructs a RREP (Route Reply) packet and sends the RREP back to the source node using the reverse path. Upon receiving the RREP, the source node updates its route cache table with an entry for the destination node and can start sending the data packet. Route maintenance on the other hand is used to handle link break. If a node detects there is a link break from data link layer, it will generate a RERR (Route Error) packet and send back to the source node using the part of the route traversed so far. The notified source node must delete the broken link from its route cache table. If the source node has another packet to send to the same destination, it must try another route or invoke route discovery process again if it does not have any other routes.

I. MISBEHAVIORS OF NODES IN MANETS

As per the mechanism of DSR, all other routing protocols designed for MANET naively assume that all the nodes in the network are cooperative in performing the networking tasks. This can be guaranteed if all of the nodes belong to a single authority where all of them have the same common objective. However that is not the case such as in civilian applications, some of the nodes may behave selfishly and only act towards those that add to their own benefits. In general, there are two types of node misbehaving: misleading and selfish.

1) Misleading Nodes :

A misleading node is selective in choosing which packet it wants to respond. It is honest node, responding to all control packets during route discovery process. However when the node receives a data packet to be further forwarded, the misleading node silently drops it. The reasons for choosing data packets for dropping is because data packets are generally greater in term of size and number than the control packets and thus consumes more energy to forward. This type of behavior is also called "Gray Hole Attack" [06].

2) Selfish Nodes:

The second type of node misbehaving is selfish node, which aims to save its resources to the maximum. This type of misbehaving node discards all incoming packets (control and data) except those which are destined to it. By dropping control packets, the nodes would not be included in the routing and then be released from being requested to forward data packets.

The similarity of these two types of misbehaving is that they both use the network to forward their own packets but refuse to provide the same services back. Misbehaving nodes can significantly degrade the performance of a MANET while Selfish nodes, have no big impact on PDR. However, this type of misbehaving can increase the average end to end delay. As the number of selfish nodes been increased, the source node will have less option on which route the data packets should travel. As a result, less attractive route will be selected which means longer delays. It also means that the remaining cooperative nodes have to take the extra burden of forwarding packets. If 50% of the nodes become selfish, the average end to end delay increases by 60%.

II. INNOVATED TECHNIQUES

Several techniques have been proposed to detect misbehaving nodes in mobile ad hoc network. These techniques can be classified into three categories:

1) A Reputation-Based Technique:

Reputation based technique on the other hand rely on building a reputation metric for each node according to its behavioral pattern. A monitoring method used by most systems in this category is called a watchdog. Watchdog was proposed by Marti et al. [08] to detect data packet non forwarding by overhearing the transmission of the next node. [09], [10] use similar monitoring technique but then propagate collected information to nearby nodes and are susceptible to false praise and false accusation attacks.

Mr. Bansal and Mr. Baker proposed a system called OCEAN [11] where the reputation of a neighbor is evaluated using only locally available information and thus avoid sophisticated and potentially vulnerable techniques of reputation propagation throughout the network. It is reported that even with direct observations of the neighbor; OCEAN performs almost as well and sometimes even better compared to schemes that share second-hand reputation information.

2) Credit Based Technique :

The basic idea of credit based technique is to provide incentives for nodes to faithfully perform networking functions. In order to achieve this goal, virtual (electronic) currency or similar payment system may be set up. Nodes get paid for providing services to other nodes. When they request other nodes to help them for packet forwarding, they use the same payment system to pay for such services. Credit based schemes can be implemented using two models:

i. The Packet Purse Model (PPM) and

ii. The Packet Trade Model (PTM) [12].

3) Acknowledgement Based Technique:

The last category is acknowledgment based technique, it rely on the reception of an acknowledgment to verify that a packet has been forwarded. Liu et al. [03] proposed the 2ACK system where nodes explicitly send acknowledgment two hops upstream to verify co-operation. This system is susceptible to collusion of two or more consecutive nodes. Furthermore, colluding nodes can frame honest ones by claiming not to receive the acknowledgment. All of the mechanisms mentioned above are designed to detect and handle misleading nodes.

III. OUR SYSTEM

In review of MANET, we observed that if a node merely intends to save own resources for itself, it is easier for the node to become a selfish node, ignoring all packets (data and control) that are not destined for it. The techniques used to detect Misleading nodes (monitoring data forwarding) are not effective for detecting selfish nodes.

The main reason is that, nodes never participate in the route request and thus would not be used to forward data packets. Furthermore, some well-behaved nodes in the network might not be required to forward data packet. Examples of those scenarios are listed as the following.

1) The node is located at the edge of the network. At that location, the node does not have any other node to forward data packet to.

2) The network is already matured where all routing to every possible destination has been established. A new node then enters the network and wishes to use the network to establish communication to another node. As long as there is no link error, there would be no changes in the routing table.

The new node would not get any RREQ packet. As a result, the new node would not be required to do data forwarding.

In our proposal, each monitoring node operates in promiscuous mode and would monitor both data and control packets that are send around within its receiving range. Each monitoring node will keep a record for each of its neighboring node. In the INETMANET [02] framework, there is already a specific table to store the information about the neighboring nodes. We add extra fields to the table as the following.

i) Neighboring nodes last_action :

It is the time the neighboring node is last seen contributing or providing services to the network.

ii) Neighbouring nodes last_request :

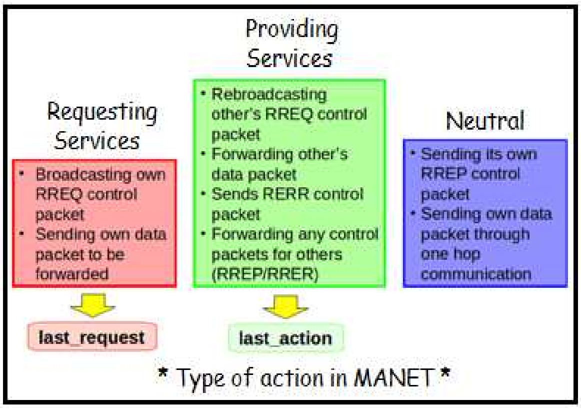
Neighbouring node is last seen utilizing or requesting for services from the network. These two fields would be updated for every action observed due to the promiscuous mode monitoring.

iii) Neighbouring nodes current_status

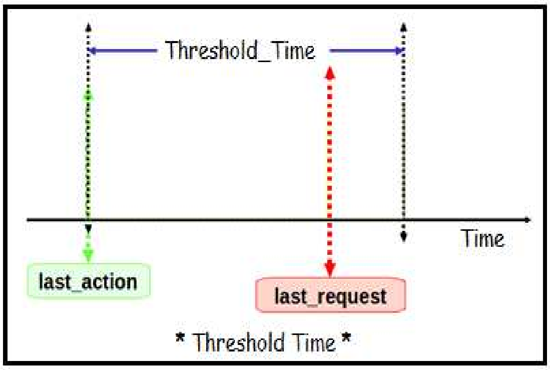
Finally, status is the current behavior of the neighboring node detected by the monitoring node. The initial status for any node is set to zero as for unknown and could later be changed to suspicious or behaved as will be explained later in this paper.

Mechanism:

To perform the detection of selfish nodes we identified type of actions that are considered as contributing, utilizing or neither of that as shown:


Providing Services
Requesting Services
Neutral
• Broadcasting own RREQ control packet
• Sending own data packet to be forwarded
• Rebroadcasting other's RREQ control packet
• Forwarding other's data packet
• Sends RERR control packet
• Forwarding any control packets for others (RREP/RRER)
• Sending its own RREP control packet
• Sending own data packet through one hop communication
last_request
last_action
* Type of action in MANET *

Whenever a monitoring node hears a request from its neighbouring node to forward a data packet, it will first check the time difference between neighbouring nodes last_request and neighbouring nodes last_action of the requestor. If it is still within a threshold as shown in following diagram, the status for the node is set to behave. We call this Threshold_Time as action hold off time.



If the time difference exceeds the threshold, the status for the node will be set to suspicious and further testing would be conducted as this suspicious node might be wrongly accused due to the special scenarios as explained above. To perform testing, a fake RREQ packet will be broadcasted into the MANET. To minimize traffic flooding in the network, only the node that receives the data forwarding request from the suspicious node would conduct this testing. In addition, this fake RREQ packet should be only allowed to pass through one hop (TTL=1).

All monitoring nodes in the neighbourhood that detect this potential misbehaviour would wait for the suspicious node to rebroadcast the fake RREQ packet within a certain timeout. If it responds to the RREQ packet, the status of the node is set to behave and the time of its neighbouring nodes last_action will be updated. If it discards the packet and does not respond, the monitoring nodes will label the suspicious node as selfish.

IV. SIMULATION

We used Network Simulation-2 (Version-2.32) tool with some nodes and selfish nodes of MANET to simulate our proposal. This network simulation-2 contain following some performance factor which makes its very useful.

- ➢ Real-system not available, it is complex/costly or dangerous.
- ➢ Quickly evaluate design alternatives.
- ➢ Evaluate complex functions for which closed form formulas or numerical techniques not available.
- ➢ NS version2 is a discrete-event driven and object-oriented network simulator.
- ➢ A package of tools that simulates behavior of network
  - Create Network Topologies
  - Log events that happen under any load
  - Analyze events to understand the network behavior.

1) Simulation Tool :

Network Simulator-2 (Version 2.32), widely known as NS2, is simply an event driven simulation tool that has proved useful in studying the dynamic nature of communication networks. NS2 consists of two key languages: C++ and Object-oriented Tool Command Language (OTcl). While the C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events (i.e., a frontend).

2) Simulation Review:

Table shows the parameters of the NS2 simulations. We simulated a network with a field size of 500m x 500m and 20 nodes. The nodes will move within the network space according to the random waypoint mobility model [20]. In random waypoint mobility model, each node will moves to a random location within the specified network area. Once the node arrives at the target location, it will remains in the position for a time (pause time) before moving to another random location. In our simulation, the pause time will set to 0.3 second. The communication patterns which will use will have constant bit rate (CBR) connection with a data rate of 3 packets per second. 20 connections will establish at random so that each node would chance to connect to every other node. We will simulate our system using two configurations of selfish nodes in the network:

i) No selfish node,

ii) ii) 4 selfish nodes.

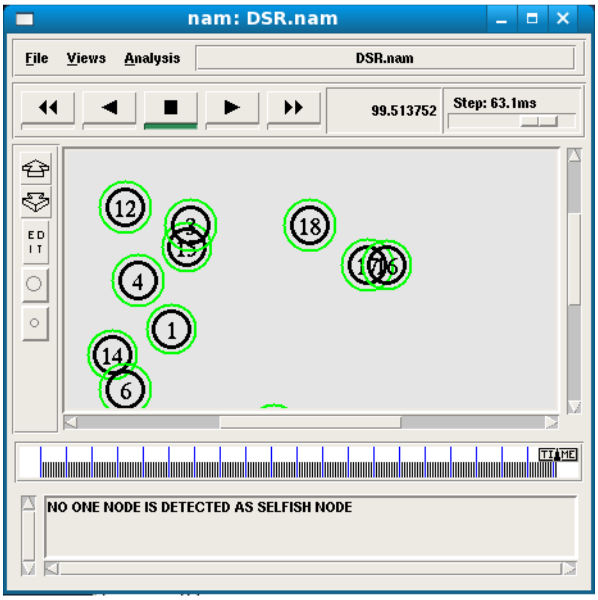
For each configuration, we will evaluates the system by changing the node's speed (0ms - 4ms) and the window size of Threshold_Time. We also have to test our system in scenarios where the selfish nodes employ different rate of selfishness (0% - 100%). In order to evaluate the detection capability of our system in a relatively small period, we only set the simulation time to 100 seconds.

Simulator	NS-2(version 2.32
Simulation Time	100 (s)
Number of Mobile Nodes	20
Topology	500 * 500 (m)
Routing Protocol	DSR
Traffic	Constant Bit Rate (CBR)
Pause Time	03(m/s)
Max Speed	20 (m/s)

V. SIMULATION RESULT:

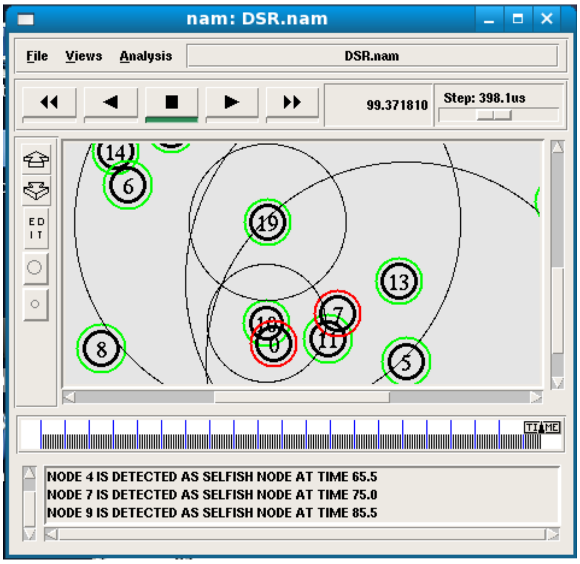
After performing proposed system using network simulation-2 with required tools and environment, we can run animator. The animator is a screen which performs all completed work. Where we simulate our system using two configurations of selfish nodes in the network:i] No selfish node,

ii] 4 selfish nodes.



NO ONE NODE IS DETECTED AS SELFISH NODE

Animator: Normal DSR (No selfish node)



NODE 4 IS DETECTED AS SELFISH NODE AT TIME 65.5
NODE 7 IS DETECTED AS SELFISH NODE AT TIME 75.0
NODE 9 IS DETECTED AS SELFISH NODE AT TIME 85.5

NS Animator: Selfish DSR (4 Selfish node)

1) Simulation Performance: The performance metrics chosen for the evaluation of black hole attack are packet delivery ratio, packet end-to-end delay and network load.

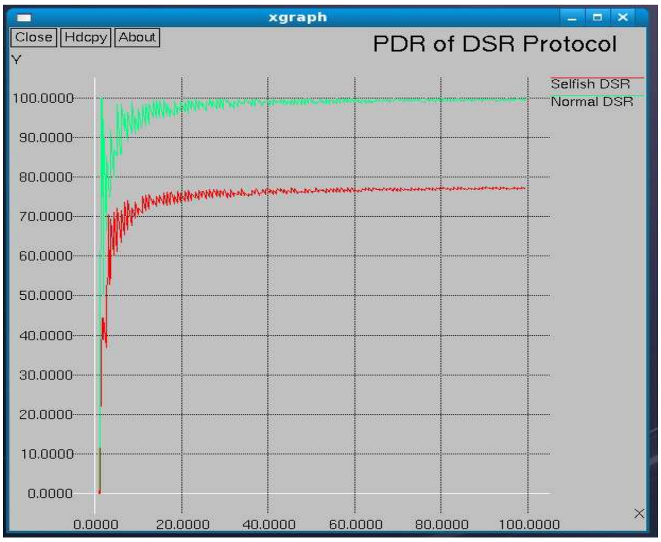
Packet Delivery Ratio: The ratio between the number of packets originated by the “application layer” CBR sources and the number of packets received by the CBR sink at the final destination.

Packet end-to-end delay: It is the average time in order to traverse the packet inside the network. This includes the time from generating the packet from sender up till the reception of the packet by receiver or destination and expressed in seconds. This includes the overall delay of networks including buffer queues, transmission time and induced delay due to routing activities. Different application needs different packet delay level. Voice and video transmission require lesser delay and show little tolerance to the delay level.

Network load, it is the total traffic received by the entire network from higher layer of MAC which is accepted and queued for transmission. It indicates the quantity of traffic in entire network. It represents the total data traffic in bits per seconds received by the entire network from higher layer accepted and queued for transmission. It does not include any higher layer data traffic rejected without queuing due to large data packet size.

2) Simulation Results:

The results were collected as comparison of Packet Delivery Ratio (PDR) for Selfish DSR (four nodes acts as selfish) and Normal DSR (No selfish node). Graph for Normal DSR shows that there is no any loss of data packets (100% PDR) ie. No one node act as a selfish node. While graph for Selfish DSR shows that there is loss of some packets, all packets are not delivered. ie. Some nodes act as selfish nodes which are find out by using our neighboring node based system for MANET to detect selfish node.



Graph for PDR of Normal & Selfish DSR

VI. CONCLUSIONS

In this paper, we propose and simulate a new neighboring node based system for MANET to detect selfish nodes. Selfish nodes refuse to carry out networking tasks for others while still using the services provided by others in the network. They ignore all data and control packets that are not destined to them, reserving resources for their own to the maximum. Simulation results show that this system performs well to detect selfish nodes in a MANET.

VII. REFERENCE

1. Parker J, Undercoffer J, Pinkston J, Joshi A. (2004). “On intrusion Detection and Response for Mobil Ad HocNetworks” in

Proceeding IEEE International Conference on Performance Computer and Communications Workshop on Information Assurance, pp 747-52.

2. KhairulAzmi Abu Bakar and James Irvine "Contribution Time-based Selfish Nodes Detection Scheme", ISBN: 978-1-902560-24-3 © 2010 PGNet

3. K. Liu, J. Deng, P. K. Varshney, & K. Balakrishnan, "An acknowledgment-based approach for the detection of routing misbehavior in manets," in IEEE Transactions on Mobile Computing, 2006.

4. K. Paul and D. Westhoff, "Context aware detection of selfish nodes in dsr based ad-hoc networks," in proceedings of IEEE Vehicular Technology Conference.

5. D. Djenouri, O. Mahmoudi, M. Bouamama, D. Llewellyn-Jones, and M. Merabti, "On securing manet routing protocol against control packet dropping," in The IEEE (ICPS'2007), Istanbul, July

6. P. Michiardi and R. Molva, "Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in (CMS'02), September 2002.

7. S. Padiya, R. Pandit and S. Patel, "Survey of Innovated Techniques to detect selfish nodes in MANET" in IJCNWMC, ISSN: 2250-1568, Vol. 3, Issue 1, Mar 2013, 221-230 © TJPRC Pvt. Ltd.

8. S. Marti, T. Giuli, K. Lai, and M. Bakar, "Mitigating routing misbehaviour in mobile ad hoc networks," in Proc. 6th Annual Int. Conf. on Mobile Computing and Networking (MobiCom'00), August 2000, pp. 255–265.

9. Q. He, D. Wu, and P. Khosla, "Sori: A secure and objective reputationbased incentive scheme for ad-hoc networks," in WCNC 2004, 2004.

10. S. Buchegger and J. L. Boudec, "Performance analysis of the confidant protocol: (cooperative of nodes - fairness in dynamic ad hoc networks)," in Proc. IEEE/ACM Workshop on (MobiHoc'02), June 2002, pp. 226–336.

11. S. Bansal and M. Baker, "Observation-based cooperation enforcement in ad hoc networks" Stanford University, 2003.

12. Dipali Koshti, Supriya Kamoji "Comparative study of Techniques used for Detection of Selfish Nodes in Mobile Ad hoc

Networks", (IJSCE) ISSN: 2231-2307, Volume-1, Issue-4, September 2011.