ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 1, Issue 2, November 2012

A System for MANET to Detect Selfish Nodes Using NS2

Sagar D. Padiya, Rakesh Pandit, Sachin Patel

***Abstract*— A Mobile ad hoc network system (MANET), contain a network area with nodes. In a mobile ad hoc network system, each node has to rely on others to relay its data packets. Since most mobile nodes are typically constrained by power and computing resources, so some nodes may choose, not to cooperative by refusing to do so while still using the network to forward their packets. Most previous works focus on data forwarding. However, dropping control packets is a better strategy for the selfish nodes to avoid themselves from being asked to forward data packets and hence could conserve resources for their own use. In this paper [1], we present a new system to detect those selfish nodes and simulate result using NS2 tool. 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. In this paper we only present the review and propose system for selfish nodes detection using a NS2 tools. Currently we are working with following keyword for practical implementation of this paper.**

***Index Terms*— Network Simulator 2; Network area; Nodes; Selfish Nodes; Mobile Ad-Hoc Networks (MANET).**

I. INTRODUCTION

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 [1]. Routing protocol such as Dynamic Source Routing [DSR] [2], AODV [3] have been designed to handle such environment. 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 [4]. However, since there is no centralized administration, the performance of a MANET greatly depends on the cooperation of all nodes in the network. In this paper, we propose a 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 approaches for node misbehavior detection that have been proposed and studied in the literature. In Section V, we introduce and present our new system to detect selfish nodes. In Section VI, we review our system for simulation environment. In section VII, we conclude the work.

II. DYNAMIC SOURCE ROUTING (DSR)

Dynamic Source Routing is one of routing protocols proposed within the 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 which operate entirely on-demand. A source node which wishes to form communication with a destination node, will first search its own route cache table.

1) 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, adds its address to the RREQ and then rebroadcast the modified RREQ. If the destination node receives the RREQ, it construct a RREP (Route Reply) packet and sends the RREP back to the source node using the exact 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.
2) Route maintenance on the other hand is used to handle link break. If a node detect 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

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 1, Issue 2, November 2012

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.

III. NODE MISBEHAVIORS IN MANETS

All other and above discussed routing protocols designed for MANET naively assume that all the nodes in the network are cooperative in performing the networking tasks like the DSR. 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. Providing network services such as forwarding packets and detecting routes consumes network bandwidth, local CPU time, memory and battery power which are limited in MANET nodes [5]. For example, simulation studies by Buttyan and Hubaux [6] show that when the average numbers of hops from a source to a destination is around 5, then almost 80% of the transmission energy will be devoted to packet forwarding. By denying services for others, a node could reserve its resources for its own use and stay longer in the network. So there is a strong motivation for the nodes not to cooperate and misbehaving.In general, there are two types of node misbehaving:

I) MISLEADING :

A misleading node is selective in choosing which packet it wants to respond. It behaves like an 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" [7].

II) SELFISH :

Selfish node 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. Simulation done by Babakkhouya et al. [8] shows that the percentage of misleading nodes can decrease the number of packets that are successfully delivered in the network. When 50% of the nodes of the network become misleading, the packet delivery ratio (PDR) degrades by 55%. Selfish nodes on the other hand, 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%. In this paper, we present a system to detect selfish nodes in a MANET.

IV. RELATED WORK

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

1) CREDIT-BASED SYSTEM:

Credit based systems [6], [9] are designed to provide incentives for forwarding packets in the form of virtual money (specifically called as Credit). Nodes earn Credit by providing forwarding services to others and have to pay to get services from other nodes. However, to protect the Credit value from attacks and modification, some costly security modules independent of nodes have to be used. In addition, colluding nodes can agree to forward their own flows to accumulate credits while dropping all other flows. Moreover, a well-behaved node that is not asked to route enough packets could not earn credits and will be unable to send its own packet.

2) REPUTATION BASED SYSTEM:

Reputation-based systems 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. [10] to detect data packet non forwarding by overhearing the transmission of the next node. [11], [12], [13] use similar monitoring scheme but then propagate collected information to nearby nodes and

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 1, Issue 2, November 2012

are susceptible to false praise and false accusation attacks. Mr. Bansal and Mr. Baker proposed a system called OCEAN [14] 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.

3) ACKNOWLEDGEMENT-BASED SYSTEM:

The last category is acknowledgment-based systems which rely on the reception of an acknowledgment to verify that a packet has been forwarded. Liu et al. [15] proposed the 2ACK system where nodes explicitly send acknowledgment two hops upstream to verify cooperation. 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.

There are a few systems that have been proposed to detect selfish nodes in a MANET. One example is Context Aware Scheme [16] introduced by Mr. Paul and Mr. Westhoff. This system uses un-keyed hash chains and a promiscuous mode to detect the misbehavior during route discovery phase. The observers of misbehavior independently communicate their accusation to the source. To convict a culprit, more than three accusations are needed. If there is only one accusing node, the accusing node itself will be considered to be an attacker. The drawback of this system is that it is more beneficial for a node not to send the alarm message to avoid the risk being the only accuser and regarded as attacker. In [17], Djenouri et al. propose two different techniques to detect two different types of control packet droppers. They suggest the use of two-hop ACK approach for monitoring directed packets (RREP, RRER) and promiscuous-based overhearing technique for monitoring broadcast packets (RREQ). Huang et al. [18] suggest that the monitoring node simply compares the ratio of
relay RREQ number between its neighbor and itself. If the ratio is smaller than a threshold, the neighbor is regarded as selfish and its packet is dropped as the punishment.

V. OUR PROPOSAL

We argue 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 systems used to detect misleading nodes (monitoring data forwarding) are not effective for detecting selfish nodes. The main reason is the 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.

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 [19] 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.

1) last_action
2) last_request
3) Status

1) last_action is the time the neighboring node is last seen contributing or providing services to the network.

2) last_request on the other hand is the time recorded the neighboring 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. 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. We have identified type of actions that are considered as contributing, utilizing or neither of that. These are illustrated in Figure 1.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 1, Issue 2, November 2012

Whenever a monitoring node hears a request from its neighboring node to forward a data packet, it will first check the time difference between last_request and last_action of the requestor. If it is still within a threshold, this is shown in Figure 2.

3) Status for the node is set to behave. We call this threshold as ActionHoldoffTime. 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. In this testing, a fake RREQ packet will be broadcasted into the network. 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).

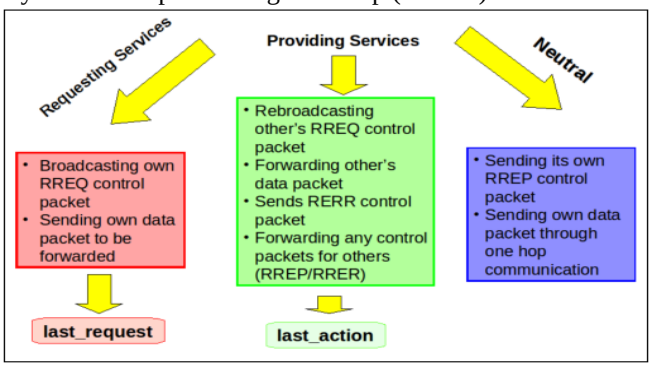

Fig 1: Type of Action

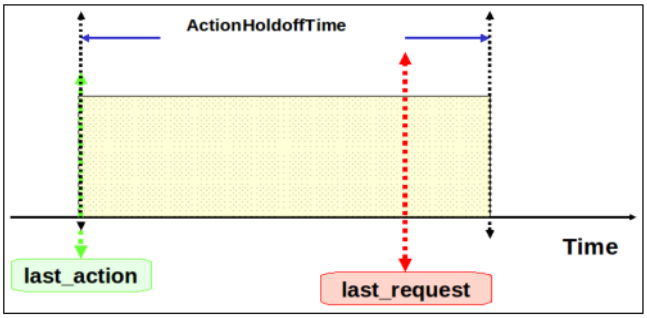

Fig 2: Action Hold off Time

All monitoring nodes in the neighborhood that detect this potential misbehavior 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 last_action will be updated. If it discards the packet and does not respond, the monitoring nodes will label the suspicious node as selfish. In our system, each monitoring node will only consider its own personal discovery and will not share this observation to other nodes. This eliminates most trust management complexity and avoids any false accusation and false praise attacks.

VI. SIMULATIONS

We are using Network Simulator 2 (NS2) tool with some nodes containing some of them as selfish nodes of MANET to simulate our proposal.

A) Simulation Review :

Table I shows the parameters of the NS2 simulations. We will simulate a network with a field size of 500m x 500m and 15 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 0.5 second. The communication patterns which will use will have constant bit rate (CBR) connection with a data rate of 3 packets per second. 15 connections will establish at random so that each node would chance to connect to every other node. We will simulate our system using four configurations of selfish nodes in the network:

i) no selfish node,
ii) 5 selfish nodes,

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 1, Issue 2, November 2012

iii) 10 selfish nodes, and
iv) 15 selfish nodes.

For each configuration, we will evaluates the system by changing the node's speed (0ms - 4ms) and the window size of ActionHoldoffTime. We also have to test our system in scenarios where the selfish nodes employ different rate of selfishness (0% - 100%). For example, 50% selfish rate means that the selfish nodes drop half of the forwarding request they receive and forward the other half. In order to evaluate the detection capability of our system in a relatively small period, we only set the simulation time to 100 seconds.

Parameters	Value
Simulation Time	100 seconds
Number of Nodes	15
Packet Size	512 bytes
Network Area	500m x 500m
Mobility Model	Random WayPoint
Transmission Range	250 meters
Routing Protocol	Dynamic Source Routing (DSR)
Data Rate	3 packets/second
Transport Protocol	UDP

B) SIMULATION METRICS :

For measuring our proposal, we consider the following metrics:

1) True Detection Rate: True detection rate (TDR), or true positives, represents the average of true detection computed as follows:

$$TDR = \sum_{i=1, mi \neq 0}^{n} \left(\frac{tdi/mi}{k} \right)$$

tdi : number of selfish nodes monitored and detect by node i
m : the number of selfish nodes monitored by node i
n : the number of nodes
k : the number of nodes that have monitored selfish nodes (whose mi?=0)

2) False Detection Rate: False detection rate, or false positives, represents the average rate of false detection using the following formula :

$$FDR = \sum_{i=1, m'i \neq 0}^{n} \left(\frac{fdi/m'i}{k'} \right)$$

fdi : number of well-behaving nodes monitored and wrongly detected by node i
m' : the number of well-behaving nodes monitored by node i
n : the number of nodes
k' : the number of nodes that have monitored well-behaving nodes (whose m i0)
A good detection scheme is where the TDR is high and FDR is low.

VII. CONCLUSION

In this paper, we propose a new low-cost system to detect selfish nodes. Selfish nodes are nodes which 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. In this paper, we proposed a new system to detect selfish node. Review results shows that this system will performs well to detect selfish nodes in a MANET. For our future work, we plan to investigate the effect of data packet rate to our detection system. We are working with this selfish nodes detection system in MANET using NS2. We are working for actual practical simulation result.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 1, Issue 2, November 2012

REFERENCES

[1] Khairul Azmi Abu Bakar and James Irvine "A Scheme for Detecting Selfish Nodes in MANETs using OMNET++"2010, Sixth International Conference on Wireless and Mobile Communications.

[2] J. Broch, D. B. Johnson, and D. A. Maltz, "The dynamic source routing protocol for mobile ad hoc network," in IETF, February 2003, internet Draft Version 08.

[3] C. E. Perkins, E. M. Royer, and S. R. Das, "Ad hoc on-demand distance vector (AODV) routing (rfc3561)," in The Internet Society, 2003, memo RFC 3561.

[4] E. M. Royer and C.-K. Toh, "A review of current routing protocols for ad-hoc mobile wireless network" IEEE Personal Communications, vol. 6, pp. 46–55, Apr 1999.

[5] H. Yang, H. Y. Luo, F. Ye, S. W. Lu, and L. Zhang, "Security in mobile ad hoc networks: Challenges and solutions," IEEE Wireless Communications, vol. 11, pp. 38–47, 2004.

[6] L. Buttyan and J. Hubaux, "Stimulating cooperation in self-organizing mobile ad hoc networks," in Mobile Networks and Applications, vol. 8, no. 5, October 2003, pp. 579–592.

[7] G. Xiapeng and C. Wei, "A novel gray hole attack detection scheme for mobile ad-hoc networks," in IFIP International Conference on Network and Parallel Computing, September 2007, pp. 209–214.

[8] A. Babakhouya, Y. Challal, and A. Buouabdallah, "A simulation analysis of routing misbehavior in mobile ad hoc networks," in The Second International Conference on Next Generation Mobile Applications, Services and Technologies NGMAST'08, 2008, pp. 592–597.

[9] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," in INFOCOM 2003, 2003.

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

[11] Q. He, D. Wu, and P. Khosla, "Sori: A secure and objective reputation based incentive scheme for ad-hoc networks," in WCNC 2004, 2004.

[12] 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 Mobile Ad Hoc Networking and Computing (MobiHoc'02), June 2002, pp. 226–336.

AUTHOR BIOGRAPHY

	**Sagar D. Padiya**, received the BE degree in Information Technology from Sipna College of Engineering and Technology, Amravati, India in 2009. Now he is appearing M.Tech in Information Technology from Patel College of Science and Technology, Indore, India. He is currently working on a dissertation of final year. His current research interest includes MANET using NS2 tool. He published one research paper in international journal and presented one research paper in national conference.
	**Prof. Rakesh Pandit** received the MSc in Physics degree from DAVV, Indore, India in 1992, and the M.Tech in Information Technology from SHIATS (AAI) Deemed University, Allahabad in 2005. From 1997 to 2011, he was with the Christian Eminent College, Indore as a programmer. He joined the department of Information Technology, Patel College of science and Technology, Indore in 2010. He has 17 years of academic experience. He published 10 research papers in various journals. His early research interest includes Cloud Computing, IT Applications, Information Security and Computer Graphics.
	**Prof. Sachin Patel** received the BE from JIT Borava in 2003, the M.Tech in computer science from DAVV, Indore in 2009. He is PhD apperaing. From 2004 to 2008, he was with JIT, Borava, and Indore as a lecturer. He joined as a Head of the department at the department of information technology, Patel College of science and Technology, Indore in 2010. He has 9.5 years of academic experience. He published 10 research papers in various national as well as 5 research papers in various international journals.