Improved Selfish Node Detection Algorithm for Mobile Ad Hoc Network

Ahmed. A. Hadi
Dept. of Computer Science
Faculty of Information Science and Technology
National University of Malaysia
Bangi, Selengor, Malaysia

Zulkarnain Md. Ali
Dept. of Computer Science
Faculty of Information Science and Technology
National University of Malaysia
Bangi, Selengor, Malaysia

Yazan Aljeroudi
Dept. of Mechanical Engineering
Faculty of Engineering
International Islamic University Malaysia
Jalan Gombak, Kuala Lumpur, Malaysia

***Abstract*—Mobile Ad hoc network (MANET) suffers from different security issues. Ideally, not all nodes in MANET cooperate in forwarding packets because of non-malicious intention. This node is called selfish node and it behaves so due to its internal state such as limited energy concerns. Selfish nodes drop packets and that harms the process of routes establishment and relaying packets. Therefore, it is very important to detect these nodes and avoid them, which guarantees improving the performance of the overall network. Here, an improved scheme has been developed for detecting selfish node in Ad Hoc On-demand Distance Vector routing protocol (AODV) based wireless routing network. Two algorithms were integrated for assuring least fault positive decision of selfish nodes detection; first one is to avoid false positive of detection of selfish nodes in forwarding Route Request (RREQ) and second one is to avoid false positive of detection of selfish node in forwarding data packets. This scheme guarantees improvement in performance of packet forwarding in terms of Packet Delivery Ratio (PDR) and End-to-End delay (E2E delay).**

Keywords*—Selfish nodes detection; AODV routing; routing protocols; MANET

I. INTRODUCTION

Ad hoc networks are defined as networks that lack a fixed infrastructure and hence are flexible and adaptive in nature. Ad hoc networks consist of individual devices, also known as nodes that communicate with each other wirelessly without a central access point. The devices, hence, do not rely on a base station to coordinate the flow of messages.[1] Instead, the individual network nodes pass packets to each other within the network. Ad hoc networks can be used in multiple applications such as creation of communication networks at times of emergency when the existing communication is damaged due to natural disasters, creating conferencing networks for office use that do not rely on the internet, home networking and personal area networks, especially with Bluetooth devices associated with a single person. A mobile ad hoc network (MANET) is defined as an ad hoc network that uses mobile nodes that are arbitrarily located. The nodes in a MANET are highly dynamic and may join and leave the system frequently. Since the nodes are highly mobile, the topology of the network changes rapidly. MANET systems have found use in many applications such as military communication networks through mobile radio transmitters and receivers, rescue missions without adequate wireless coverage. Fig. 1. Shows an example of a MANET.

Fig. 1. Mobile Ad hoc Network

MANETs are highly efficient in establishing an impromptu mobile network. However, there are certain issues that these networks face. Since MANET networks are based on mobile nodes, these nodes usually operate within limited resources in terms of power and bandwidth availability, and quality of the node hardware. MANET networks may also face a problem of route optimization. Apart from these constraints, MANET may also face security concerns such as passive and active attacks for information extraction. Furthermore, MANET may contain malicious nodes that drop all or selective packets. This work focuses on malicious nodes that act selfishly in the network such that they drop packets to conserve resources while using the network to broadcast their own packets. Selfishness within nodes has imminent disastrous effect on MANET as it reduces the performance of overall network and can paralyze the network when the number of selfish nodes increases in the network. Therefore, there is high motivational aspect to address this problem from research perspective.

In the literature, numerous algorithms have been developed to detect selfish nodes in MANET. In the work of [2], a fuzzy reputation system has been proposed to discipline selfish behavior of nodes and motivate packet forwarding. [3] implemented secure and objective reputation-based algorithm in detecting nodes that are selfish in nature where every node

is liable to keep track of other nodes or acquire the reputation from a centralized node. [4] suggested a credit-payment scheme. The objective of this scheme is to encourage nodes to forward packets by earning credits which they need in order to transmit their own packets. [5] proposed activity-based overhearing, iterative probing and unambiguous probing to detect multitude of selfish nodes in MANET. In another paper, [6] employed a fuzzy-based analyzer to differentiate trusted and selfish behavior in nodes. The method incorporated the concept of trust and certificate authority to combat selfishness. [7] proposed a collaborative watchdog to improve selfish node detection. [8] proposed two network-layer acknowledgement-based scheme to detect misbehaving nodes and then inform the routing protocol to avoid these nodes in the future. [9] used game theory to study behavior of nodes and apply reputation as a tool to encourage cooperation in nodes.

Above papers discuss about detecting selfish behavior of nodes in MANET and propose ways to encourage cooperative behavior. However, these papers overlook the aspect of false decision in detecting the selfish nodes. False detection of selfishness may degrade network performances, as normal nodes are susceptible to be falsely identified as selfish, resulting in elimination of participation in packet forwarding. Therefore, an improved scheme based on AODV routing protocol has been proposed in this paper. The work presented in this paper builds upon [10] and identifies key problems in it. Also, it provides a robust solution to minimize false detection of selfish nodes. The organization of this article is as follows. In Section II a problem statement is presented followed by methodology in Section III. Section IV presented the results obtained from tests and discussion on the results and this paper is ended with conclusion and future work in Section V.

II. Problem Statement

Selfishness within nodes has imminent disastrous effect on MANET as it reduces the performance of overall network. Therefore, detecting and eliminating these nodes is a vital step in ensuring a working system. [10] presented a method of detecting selfish nodes and tested the proposed method on an AODV routing protocol. Although the method looks promising and efficient, it suffers from several limitations. After implementing the protocol and running the method on MATLAB, it has been found that their proposed selfish nodes detection mechanism has two prominent drawbacks.

Firstly, the method failed to notice the problem pertaining to the first type of selfish node; dropping RREQ packets. Due to the dynamic structure of MANET, a node may receive RREQ more than once and from difference source nodes. This method suggests that when a node receives RREQ from a node with the same ID it has previously received, it will drop the packet, hence considered as potentially malicious node but it does not necessarily mean it is selfish.

Secondly, the method could not address an issue related to the second type of selfish node; dropping data packet. When a node receives a data packet, it forwards the packet to the neighboring node following the established route until the packet reaches the destination node. Problem arises when neighboring node may be out of coverage zone of the forwarding node. After sending a data packet, sending node does not know if the receiving node has successfully received the packet, and it might identify the receiving node falsely as malicious in case no forwarding action has been performed by the receiving node.

In both cases, the paper failed to see these problems and therefore could see the next node of a particular forwarding node as selfish in nature. This false decision will lead to lower performance as normal nodes could be terminated from the network while in fact, the nodes can participate in forwarding packets. This paper proposes a method to minimize false detection of selfish nodes.

III. Methodology

This work uses the (AODV) presented in [10] and builds upon the said protocol to reduce false detection of selfish nodes more efficiently.

This paper focuses on two behaviors that were used to identify the selfish nodes namely (*A*) Not forwarding RREQ messages and (*B*) Not forwarding data messages. The drawbacks of the above methods have been mentioned in the previous section. This section will present the potential methods to deal with the stated problems.

A. Not forwarding the RREQ message

As mentioned in the problem statement, the work presented earlier is not robust in its identification of the selfish node. It ends up classifying a normal node as selfish if the node has broadcasted an RREQ message previously but does not broadcast the same message again. This paper introduces a new type of packet known as the Route Request Confirmation Packet (RRC). The main purpose of this packet is to confirm to the other nodes that the current node has previously rebroadcasted the message and is not a selfish node. Hence, upon receiving the RREQ message, a normal node rebroadcasts the message in case it has not received it earlier. Otherwise, it sends an RRC packet to the sending node. The aim is to let the sending node know that it has forwarded the same message once and there is no need to do so again. Algorithm.1.presents the algorithm used in the modified protocol.

Algorithm.1. False detection of selfish node for not-forwarding RREQ

1: **Start.**
2: Source node sends RREQ to all of its one hop neighbors
3: Each normal neighbor node either rebroadcasts the RREQ to its neighbor nodes or sends an RRC packet to the sender node if it has already rebroadcasted the same RREQ before.
4: After waiting for a prefixed period of time, the source node checks its routing table and examines the behavior of its neighbors
5: **IF** the source node receives back the RREQ packet OR receives an RRC packet from its neighbor,
THEN this neighbor node is characterized as normal node.
ELSE the neighbor node is marked as potential selfish node.
6: Flooding of the RREQ continues. Each intermediate node receiving an RREQ must rebroadcast the message or send an RRC if it has rebroadcasted the same message before.
7: For each intermediate node, repeat Step2 to Step 4 and sender intermediate node is considered as the source node.
8: Process continues until destination node is reached.
9: **End**

B. Not forwarding the data packet

As mentioned in the earlier section, since the nodes of the network are mobile, a node may not be able to forward a data packet not because it is a selfish node, but because it never received the data packet in the first place, due to a break in the route. The existing work fails to recognize this possibility and hence may mark a normal node as selfish. In order to avoid false decisions about this type of selfish node, each node must update its routing table before it sends any data packet. To guarantee getting the most updated information about neighbors, the process of updating routing table has been transformed to be event-based instead of being periodic. Hence, each node before sending a data packet broadcasts a hello message and updates the routing table to confirm its connection to the next node. If the next node is not available, the node drops the packet and tells the previous node with a Path Break (PB) message that the next node is not available and it cannot send the data packet. Algorithm.2. Presents the algorithm used where (SN) stands for sender node and (RN) stands for receiver node.

Algorithm.2. False detection of selfish node for non-forwarding data packets.

1: **Start.**
2: Initially SN is source node and RN is the 2nd node of the transmission path.
3: SN sends a Hello message to RN to confirm that RN is still present in the transmission route and updates its routing table.
4: **IF** SN does not receive back the hello message from RN,
THEN RN is considered to be out of the transmission route and another route is established.
ELSE RN is in the transmission route.
5: Data packet is sent from SN to RN.
SN and RN are modified whenever data packet reaches a new intermediate node of the transmission path. Whereby the previous RN becomes new SN.
6: Step 3 is repeated with new SN and RN nodes.
7: **IF** new RN is out of transmission path,
THEN SN sends a PB message to the previous node indicating a break in transmission path.
ELSE SN broadcasts data packet to RN.
8: **IF** new SN does not broadcast any data packet,
THEN SN is considered potential selfish node.
ELSE SN is a normal node.
Process continues.
9: **IF** SN = RN,
THEN the data packet has reached the destination successfully.
ELSE data packet has not reached the destination.
11: **End**
12: **End**

In order to validate the proposed method, MATLAB environment has been used for simulation. A MANET of 49 nodes has been established where each node has coverage zone equal to 250. The average size of packet is 80 bit. The mean velocity of nodes is 10 unit/sec. The timeout time after sending a route request packet is 1.5 sec and the route request buffer size is 1000 packet.

The size of route reply packet and route request packet is 100 packets. Data packet lifetime is 10 sec. To generate data packets, two Poisson random variables have been used where

one is for generating random times with mean equal to 6 second and the other is for generating a random number of packets with mean equal to 2 packets. For the proposed method route request selfishness threshold and data packet selfishness threshold has been chosen to be 150 and 10 respectively whereas, in the benchmark case, the route request selfishness threshold has been chosen to be 1500 in order to decrease the number of false decisions during the execution. The results from the simulation have been discussed in detail in the next section.

IV. Results and Discussion

This section discusses the results from the conducted experiments in detail. The proposed modified algorithm was tested with respect to a benchmark in, Packet Delivery Ratio (PDR), End-to-End delay (E2E), overhead and energy consumption measures. The results are hereby published.

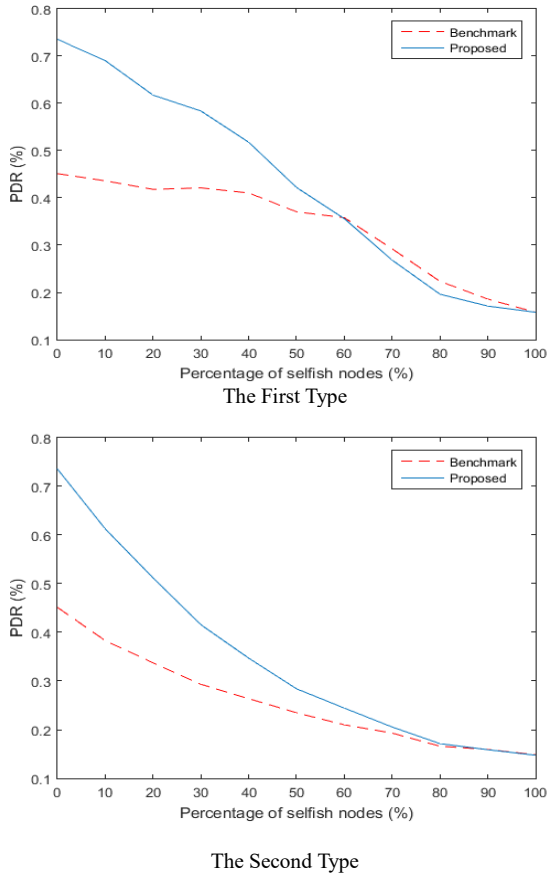


Fig. 2. PDR percentage vs selfish node percentage for first and second types of misbehavior

Fig.2. PDR percentage vs selfish node percentage for first and second types of misbehavior. Shows the comparison of the proposed algorithms with respect to the benchmark in PDR performance. The x-axis represents the percentage of selfish nodes in the network and the y-axis represents the PDR percentage. A false decision in marking a normal node as a selfish node decreases the PDR. It can be seen from Fig.2. that the proposed method achieves greater PDR performance in each of the two types of algorithms as compared to the benchmark. The results thus clearly indicate a decrease in the false decisions regarding the selfish nodes. In the first type however, the PDR performance drops slightly lower than the benchmark when the number of selfish nodes is more than 60%. This is because most of the nodes in the network are real selfish nodes. In the second type of algorithm, the PDR remains higher than the benchmark until the number of selfish nodes reaches 78% and is equal to the benchmark from then on. From Fig.2.By taking the reading at 0% of selfish node for first type and second type of misbehavior, the data points show similar values. Therefore, the improvements on PDR for both cases are calculated as follows.

$$Improvement\ PDR = \frac{0.74-0.45}{0.74} x100 = 39.2 \quad (1)$$

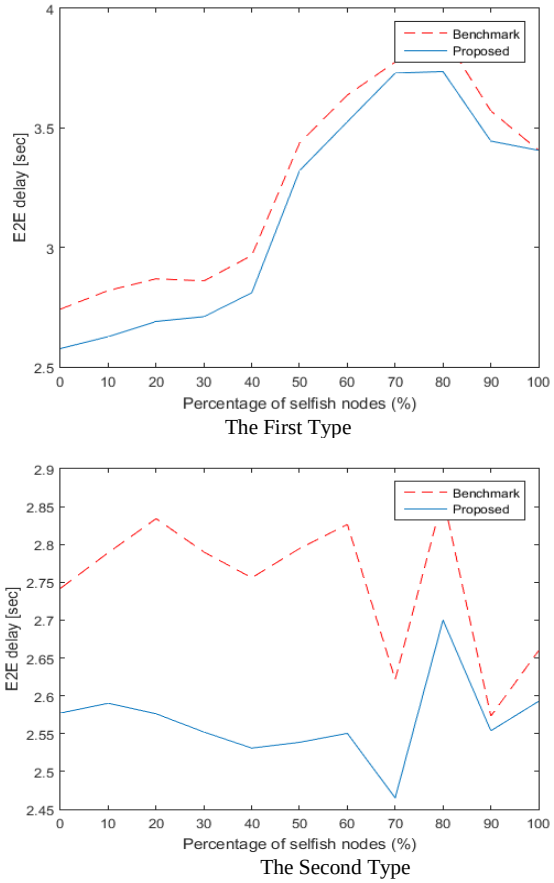
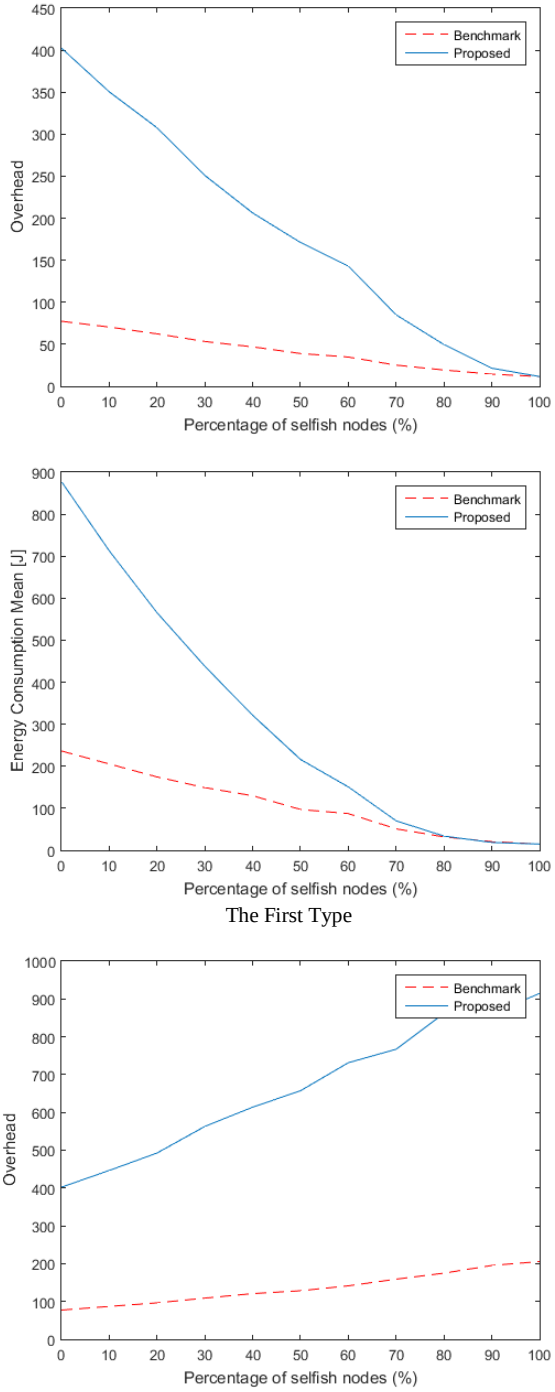


Fig. 3. E2E delay vs selfish node percentage for first and second types of misbehavior

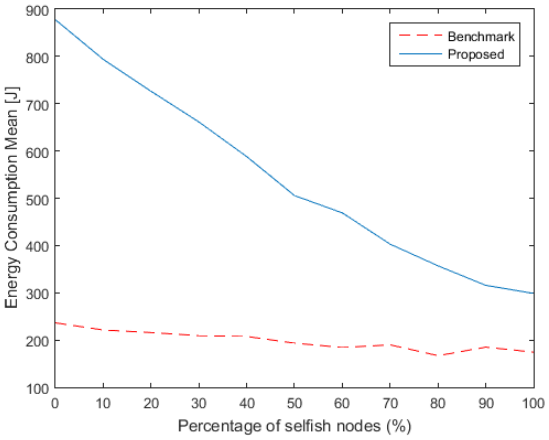
Fig.3.Presents the comparison of the E2E delay of the proposed algorithms and the benchmark. The x-axis represents the percentage of selfish nodes whereas the y-axis represents the E2E delay. It can be seen from Fig.3.That the proposed algorithms reduce the E2E delay in both the algorithm. This is due to the reason that the proposed algorithm only identifies the real selfish nodes in the network hence reducing the time needed to establish the routes also more nodes relay packets from other nodes, hence reducing the E2E delay. By taking values at 0% of selfish node, the improvements on E2E delay for first type and second type of misbehavior are calculated as follows.

$$ImprovementE2Edelaytype1 = \frac{2.73-2.58}{2.73} x100 = 5.5 \quad (2)$$

$$ImprovementE2Edelaytype2 = \frac{2.74-2.58}{2.74} x100 = 5.8 \quad (3)$$



The First Type



The Second Type

Fig. 4. Overhead vs selfish node percentage for first and second types of misbehavior on upper half of figure and mean energy consumption vs selfish node percentage in lower half

Fig.4. shows the performance of the proposed algorithm in comparison to the benchmark with respect to the overhead and the energy consumption measures. The upper half of the figure represents the overhead comparison with the y-axis representing the overhead value. The lower part of the figure shows the comparison of energy consumption. The energy consumption value is represented on the y-axis. The x-axis in both cases shows the percentage of selfish nodes. From the Fig.4. It can be seen that the overhead and energy consumption of the proposed algorithm is higher than the benchmark in both types of algorithms. This is due to the fact that a new packet i.e. Route Request Confirmation Packet (RRC) was introduced thereby increasing both, the overhead as well as the energy consumption. Furthermore, it should be noted that some of the selfish nodes in the benchmark algorithm are not identified properly, hence further reducing the overhead and energy consumption.

V. Conclusion and Future Work

Presence of selfish nodes in any network especially MANET may impact the performance of the whole communication system. As much as it is important to detect selfish nodes to improve network performance, false detection may also affect the network because of the result of avoidance of collaboration with selfish node. Based on[10], two main drawbacks are found. They are failure of a node to rebroadcast same packet more than once and hence it drops the packet,

which results in being falsely identified as a selfish node. The second drawback is the possibility of RN being out of SN's coverage area. In this paper, an improved scheme has been proposed. This proposed scheme detects selfish nodes effectively in a MANET and at the same time works to minimize false detection of selfish nodes to ensure data packets are transmitted from the source node to destination node more efficiently. First type of selfish node issue has been handled by introducing a new packet type namely Route Request Confirmation Packet (RRC) which lets other nodes know that a node has forwarded a packet previously and it is not selfish. The second type of selfish node problem has been solved by updating routing table and making it event-based instead of periodic-based. Experiments and tests are conducted in MATLAB environment and the results show improvement with 40% of PDR and 5.5 – 5.8% of E2E delay. Future work is to validate the proposed approach is real world scenario. Another future aspect of work is to develop approaches of re-introducing the node to the network in case of changing the selfish behavior to normal.

REFERENCES

[1] Ahamad T. Detection and Defense Against Packet Drop Attack in MANET.International Journal of Advanced Computer Science and Applications (IJACSA) 2016;7.

[2] Jalali M, Aghaee NG. A fuzzy reputation system in vehicular ad hoc networks. Procedia Computer Science 2011;5:951-6.

[3] He S, Prempain E, Wu Q. An improved particle swarm optimizer for mechanical design optimization problems. Engineering Optimization 2004;36:585-605.

[4] Yoo Y, Ahn S, Agrawal DP. A credit-payment scheme for packet forwarding fairness in mobile ad hoc networks. Communications, 2005 ICC 2005 2005 IEEE International Conference on: IEEE; 2005. p. 3005-9.

[5] Kargl F, Klenk A, Schlott S, Weber M. Advanced detection of selfish or malicious nodes in ad hoc networks. European Workshop on Security in Ad-hoc and Sensor Networks: Springer; 2004. p. 152-65.

[6] Manoj V, Aaqib M, Raghavendiran N, Vijayan R. A novel security framework using trust and fuzzy logic in MANET. International Journal of Distributed and Parallel Systems 2012;3:284.

[7] Hernandez-Orallo E, Serrat MD, Cano J-C, Calafate CT, Manzoni P. Improving selfish node detection in MANETs using a collaborative watchdog. IEEE Communications letters 2012;16:642-5.

[8] Balakrishnan K, Deng J, Varshney V. TWOACK: preventing selfishness in mobile ad hoc networks. Wireless communications and networking conference, 2005 IEEE: IEEE; 2005. p. 2137-42.

[9] Gupta R, Somani AK. Game theory as a tool to strategize as well as predict nodes' behavior in peer-to-peer networks. Parallel and Distributed Systems, 2005 Proceedings 11th International Conference on: IEEE; 2005. p. 244-9.

[10] Das D, Majumder K, Dasgupta A. Selfish node detection and low cost data transmission in MANET using game theory. Procedia Computer Science 2015;54:92-101.