Finding Selfish Nodes in Mobile Ad Hoc Networks
Miss. Pallavi Agrawal*1, Prof. Dinesh Datar*2

*1 PG Student of computer science Department, G.H. Raisoni Engineering College, Amaravati, India
*2 Prof. of computer science Department, G.H. Raisoni Engineering College, Amaravati, India
Pallavi.a.agrawal123@gmail.com*1, chetanagrawal13@gmail.com*2

Abstract
Securing the mobile ad hoc networks (MANETs) in an untrustworthy open environment is always a challenging problem. In recent years, mobile ad hoc networks have become a very popular research topic. MANETs are attractive technology for many applications such as rescue operations, tactical operations, environmental monitoring, conferences, and the like. However, performing network functions consumes energy and other resources. To save its energy a node may behave selfishly and uses the forwarding service of other nodes without correctly can severely degrade the performance at the routing layer. Specifically, nodes may participate in the route discovery and maintenance process but refuse to forward data packets. In this survey various methods for detecting selfish nodes are discussed with their key advantages. Moreover one of the most important aspects is to propose specific behavior pattern creation that would let to evaluate neighbor behavior; I surveyed the key algorithms for constructing behavior pattern for the neighboring nodes in MANETs. In the literature there are many methods which deal with the selfish behaviour of the nodes. This paper compares different methods available for reducing the effect of selfish nodes in mobile ad hoc networks.

Keywords: Mobile Ad Hoc Networks, Routing misbehavior, Selfishness, Network security.

I. INTRODUCTION
Mobile ad-hoc networks (MANETs) allow for wireless devices to form a network without the need for central infrastructure [1]. While the lack of need for infrastructure allows the network to be very flexible, it also makes routing a critical concern in the network. The data collection component is responsible for collection and preprocessing data tasks: transferring data to a common format, data storage and sending data to the detection module. In ad-hoc wireless networks each computer with a wireless interface can communicate directly with participating nodes. These nodes can self-organize without central management and special infrastructure[2][3]. The network is established using (limited range) radio communication where each node acts as both data terminal and data transfer equipment. Moreover, nodes can move freely resulting in changes to the network topology and updated routing in order to forward the packets. The topology change depends on different factors such as mobility model, node speed etc. Due to the infrastructure less nature of MANETs packets sent between distant nodes are expected to be relayed by intermediate ones [3], which act as routers and provide the forwarding service. The forwarding service is closely related to the routing. It consists in correctly relaying the received packets from node to node until reaching their final destination, following routes selected and maintained by the routing protocol [3]. These services (routing and data forwarding) together are at the core of the network layer. The nature of MANET makes cooperation among nodes essential for the system to be operational. In some MANET’s applications where all nodes belong to a single authority (in the application layer point of view) and have a common goal, e.g.-soldiers in a military unit during a battlefield or rescuers in a rescue team during a rescue operation, nodes are cooperative by nature[2][3]. However, in many civilian applications, such as networks of cars and provision of communication facilities in remote areas, nodes typically do not belong to a single authority and do not pursue a common goal. In such networks, forwarding packets for other nodes is not in the direct interest of anyone, so there is no good reason to trust nodes and assume that they always cooperate.

II. 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:

**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].

**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.

## III.RELATED WORK

### A.Credit Based Methods

Credit based methods are also called as incentive based methods. In these methods selfish nodes are not punished instead unselfish nodes are rewarded for helping other nodes. This stimulates the cooperation of nodes in the network. This section discusses some of the credit based systems in the literature.

### B. Secure Incentive Protocol

This approach assumes that each mobile node (MN) has a tamper-proof security module such as SIM cards in GSM networks, which deals with security related functions and each intermediate node (IN) puts non-forged stamps on the forwarded packets as a proof of forwarding[2]. Secure Incentive Protocol, (SIP) uses “credits” as the incentives to stimulate packet forwarding. For this purpose, each smartcard has a credit counter (CC) which is pre-charged with a certain amount of credits before shipped out[2][3]. The charging and rewarding on a node is done by decreasing or increasing the CC in that node and the CC will retain its value even when the MN is power off. When the MN is power-on again, it could still reuse the credits in the CC even in another SIP-enabled ad hoc network. To guarantee the security of SIP, each smartcard contains a private number and a public number (keys). The nodes have no knowledge about the keys stored in the smartcard and could not change CC in an unauthorized way either. SIP is session-based and mainly consists of three phases. During the first Session initialization phase, a session initiator (SI) negotiates session keys and other information with a session responder (SR) and INs between them. And each IN puts a non-forged stamp on each data packet forwarded and SI/SR collect those stamps for later rewarding use in the next Data forwarding phase[2]. The final phase is Rewarding phase, in which each IN is awarded a certain number of credits based on
the number of forwarded packets. Advantages of this method are 1. SIP is routing-independent in the sense that it could coexist with any on-demand unicast routing protocol such as DSR and AODV. 2. SIP is session-based rather than packet-based. 3. Security module is tamper-proof and hence unauthorized access is not allowed. But the problem with this approach is, it needs every node to possess the hardware module and SIP is implemented in the hardware module. Hardware module will not be available in the already existing mobile nodes.

C. Sprite

The basic idea of their scheme is as follows: a Credit Clearance Service (CCS) is introduced to determine the charge and credit to each node involved in the transmission of a message [5]. When a node receives a message, the node keeps a receipt of the message and later reports it to the CCS when the node has a fast connection with the CCS. Payments and charges are determined from a game theory perspective. The sender instead of the destination is charged in order to prevent denial-of-service attack in the destination by sending it a large amount of traffic [5][6]. Any node who has ever tried to forwarding a message is compensated, but the credit a node receives depends on whether or not its forwarding action is successful – forwarding is considered successful if and only if the next node on the path reports a valid receipt to the CCS.

Three selfish actions and the corresponding countermeasures are discussed in the paper:

1. After receiving a message, a selfish node may save a receipt but does not forward the message. To prevent this, the CCS should give more credit to a node who forwards a message than to a node that does not forward a message to motivate a selfish node to forward others’ message. To achieve this objective, if the destination does not submit a receipt, the CCS first determines the last node on the path that has ever received the message. Then the CCS pays this last node less than it pays each of the predecessors of the last node [5].

2. A node received a message may not report the receipt. This is possible if the sender colludes with the intermediate nodes, so that the sender can pay the node a behind-the-scene compensation, which is little bit more than the CCS will pay, and the sender still get a net gain.

In order to prevent this cheating action, the CCS charges the sender an extra amount of credit if the destination does not report the receipt so that colluding group get no benefit.

3. Since reporting a receipt to the CCS is sufficient for getting credit, a group of colluding nodes may forward only the receipt of a message, instead of forwarding the whole message, to its successor.

Two cases are considered: 1) the destination colludes with the intermediate nodes; 2) the destination does not collude with the intermediate nodes. In the first case, since the message is for the destination and if the destination really submits the receipt, then the intermediate nodes and the destination should be paid as if no cheating had happened. In the second case, if the destination does not report a receipt of a message, the credit paid to each node should be multiply by a fraction, \( r \), where \( r \leq 1 \).

Modeling the submissions of receipts regarding a given message as a one-round game, the authors proved the correctness of the receipt-submission system using game theory. Although the main purpose of the system is for message-forwarding in unicast, it can be extended to route discovery and multicast as well. This scheme, however, may have several issues:

1. Receipts of each node along a path maybe submitted to the CCS at different times, making it difficult for the CCS to determine the actual payment to each node [5].

2. The scheme[6] is based on DSR, which includes the path in the forwarding message. A malicious node not on the path can collude with nodes on the path to forge a receipt and spoof the CCS.

IV. A FRAMEWORK FOR DETECTION OF SELFISHNESS

This Paper describes a new framework based on Dempster-Shafer theory-based selfishness detection framework (DST-SDF) with some mathematical back-ground and simulation analysis. The DST-SDF is dedicated for
MANETs based on standard routing like dynamic source routing (DSR) [12]. The main concept relies on end-to-end packet acknowledgments in the following way: every time a source node sends a packet to a destination node, it waits for a certain predefined time for an acknowledgement of the packet. If one arrives within the predefined time, the source node has reason to claim that all nodes on the path are cooperative (none is selfish). Otherwise if there are no other indications of faultiness on the path (e.g., RERR messages), the source node knows that there are selfish nodes on the path. Whenever an acknowledgment does or does not arrive in time, a special recommendation message is sent out to inform the other nodes about the detected situation (selfish or cooperative behavior on the path, respectively). Every node in the network is equipped with a dedicated component executing a DST-based algorithm that uses received recommendation messages to evaluate the selfishness of each node. The resulting values can be used as routing metrics while selecting packets’ routes in the near future.

V. CLASSIFICATION OF DETECTION TECHNIQUES

Based on the architectural differences the detection techniques are classified among four major categories: Stand-alone IDS, Distributed and cooperative IDS, Hierarchical IDS, and Mobile Agent based IDS.

A. Standalone IDS

Here in this architecture IDS is executed independently for each node and there is no data interchange for taking the decisions. This architecture is more suitable for flat network infrastructure.

B. Distributed and cooperative IDS

An intrusion detection system in MANET which is both distributed and dependent on nodes cooperation. Each node cooperates in intrusion detection and an action is performed by IDS agent on it. Each IDS agent is responsible for detection, data collection and local events in order to detect intrusions and generate an independent response.

C. Hierarchical IDS

Here is the architecture such that the network is divided in clusters. Each cluster has a cluster head which performs control functions like routing packets. Each IDS agent is performed on every member node and it is locally responsible for its node.

D. Mobile agent for IDS

In this architecture the mobile agents move in the network and helps in the distributed intrusion detection. This architecture reduces the energy consumption and is more fault tolerant. Also there are various methods proposed in the literature to detect selfish nodes which are classified as either credit based, reputation based or game theory based methods.

Reputation based method [2] [1] detects selfish node based on the reputation it gets from the other neighboring nodes depending on its participation.

In credit based method [6], node collects incentives by forwarding the messages traveling through it. A node is required to maintain credit threshold so as to send its own packets.

In game theoretic scheme the detection system compares node’s performance against other nodes based on a repeated game theory [12].

Fig. Selfish Node Detection Approaches
V. CONCLUSION

This paper discussed several approaches for dealing with selfish nodes. Selfish nodes are a real problem for ad hoc networks since they affect the network throughput. Many approaches are available in the literature. But no approach provides a solid solution to the selfish nodes problem. The Credit based approach provides incentives to the well behaving nodes and just by passes the selfish nodes in selecting a route to the destination. But selfish node still enjoys services without cooperating with others. The detection and isolation mechanism isolates the selfish nodes so that they don’t receive any services from the network. Thus penalizing the selfish nodes. But what happens if many nodes become selfish? Network communication itself will become impossible. Thus we cannot eliminate all the selfish nodes from the network. A new method to reduce the effect of selfishness and stimulating the nodes to cooperate in the network services should be developed. But the overhead in achieving this should also be less. Because we should remember that after all we are dealing with battery operated devices

REFERENCES


