Enhancing Node Cooperation in Mobile Ad Hoc Network

S. Kami Makki
Department of Computer Science, Lamar University, Beaumont, Texas, USA
Email: kami.makki@lamar.edu

Keenan B. Bonds
Department of Computer Science, Norfolk State University, Norfolk, Virginia, USA
Email: k.b.bonds@spartans.nsu.edu

Abstract—Mobile Ad Hoc Networks (MANET) have been a research interest over the past few years, yet, node cooperation has continually been a recognized issue for researchers. Because of their lack of infrastructure, MANETS depend on the cooperation of intermediate nodes in order to forward or send packets of their own to other nodes in the network. Therefore, nodes located in the central area of the network are used more frequently than the nodes located on the outer boundary. The inner nodes have to forward the packets of other nodes and if there is no payoff for forwarding the packets, the nodes may start to refrain from forwarding the packets of others to save their energy. The Community Enforcement Mechanism has been proposed to force the cooperation of among the nodes and reduce their misbehavior. Although, it provides cooperation among the nodes, it does not essentially increase the network life. In this paper, we present an efficient algorithm to improve the longevity of a MANET based upon more structured nodes cooperation.

Index Terms—Mobile Ad Hoc Network, Selfish, Cooperation, Misbehavior

I. INTRODUCTION

A Mobile Ad Hoc Network is a self-configurable wireless network made up of mobile devices, such as laptops, PDAs, smart phones etc., MANETS lack infrastructure, meaning that they have no central coordinator [13]. Given that they have no infrastructure, they require multi-hop relays. This means that a packet may be forwarded by more than one node until it reaches its destination. If a node desires to send a packet to a further node that is not in its communication range, it must depend on the intermediate nodes cooperation, which can forward its packets. Nodes adjacent to each other can communicate directly if they are in communication range. With the absence of a fixed infrastructure, the cooperation of nodes is more difficult to be managed since the nodes can move freely around the network, causing constant link breakage. Therefore, it is not guaranteed that the packets will arrive to their destination.

There are numerous applications that Mobile Ad Hoc Networks are associated with, including military communication, emergency services, commercial products, etc [1]. In order for the applications to operate correctly, each node is required to forward packets for the others. A node can refuse to forward a packet if it does not get any valuable gain from doing so. Therefore, to maintain a functional network, a node must not behave in this manner.

While dealing with Mobile Ad hoc Networks, one faces a plethora of challenges. These include constant link breakage, insufficient node battery power, low bandwidth, and security [13]. As stated earlier, each node is in constant motion, so it is quite common for the links between the nodes to be broken. If a node does not have sufficient battery power, it will fail to forward packets to other nodes. Along with these challenges, the node can refrain from cooperation. Therefore, in MANETS uncooperative nodes are classified into three types: malicious nodes, faulty nodes, and selfish nodes [13]. Malicious nodes can enter networks without authenticating themselves, launching a range of attacks on other nodes such as “blackhole” or “denial of service” (DoS) [2, 12]. Faulty nodes because of hardware malfunctions or software errors will withdraw from the participation of the network. Selfish nodes do not forward packets since they want to save their own resources, such as battery power, for their own sending of packets.

In this paper, the main objective is to label the issues of selfish nodes and introduce an algorithm which encourages nodes to evenly distribute their resources, leading to a functional network. In section II we will introduce related work and different cooperation schemes used in MANETS. Section III will present our proposed algorithm. In Section IV, we will explain the simulation environment and the results from our experiments, followed by our conclusion in Section V.

II. RELATED WORK

A large amount of research has been done to tackle the issue of node cooperation in multi-hop networks. The
cooperation of nodes in MANETS can be classified and divided into two schemes: Virtual currency based schemes and Reputation based schemes. Virtual currency based schemes offer an incentive to the nodes that help to cooperate and forward packets in the network [5, 6, 8]. A node will not receive any help if it does not have any incentives to offer. Reputation based scheme uses the behavior or reputation of the node to mitigate the selfish behavior [3, 4, 7, 9]. In order to attain the reputation of a node, reputation messages can be exchanged with other nodes or it can be obtained through direct observation. We describe these schemes in more detail below.

A. Virtual Currency based Schemes

When forwarding a packet for another node, a node has to use own resources, such as battery life. Without any gain, a node will be hesitant to cooperate to relay packets. To address this problem, uncooperative nodes will be given an incentive in order to forward packets for the other nodes. The incentives are also used to motivate nodes for future cooperation. To reward a node for its services, the virtual currency system [5, 6, 8] uses credit or micro payments. Therefore, virtual currency is used to charge the nodes sending or receiving a message and to reward the nodes that forward these messages. Two examples of the virtual currency system which are discussed below in more detail are Nuglets [6, 8] and Sprite [5].

A.1. Nuglets

The virtual currency, Nuglets, developed by Buttyan and Hubaux, is used to reward/charge for the forwarding of packets [6, 8]. There are two types of models that use Nuglets: Packet purse model, where the source of the packet is charged, and the Packet trade model, in which the destination of packet is charged. In the packet purse model, the source loads the packet with Nuglets equal to the number of hops that it takes to reach the destination. Each intermediate node takes the same amount of Nuglets and forwards the packet until it reaches the destination. In the packet trade model, the intermediate nodes use the “trading system”. An intermediate node purchases the packet from the prior node and then trades it to the next node for more Nuglets. This continues until it reaches the destination, where it purchases the packet for the total cost. This system would only function if each node has a valid counter so that the increments/decrements of the Nuglets can be managed equally.

A.2. Sprite

Sprite uses a credit-based system to offer incentives for nodes in MANETS to cooperate. This credit-based system manages the rewards and credit charges for each node participating in the forwarding of packets [5]. A receipt is given to a node that receives a message, later reporting it to the credit system when they are in contact. A node that attempts to forward a packet is benefited, but the amount of credit received depends on if the forwarding was successful. It is only considered successful if the node at the next hop has a receipt to give the credit system. This method is assumed to have a centralized server to manage these rewards and credit charges.

B. Reputation Based Schemes

Reputation based schemes are based on the performance of nodes in a network. A node is monitored by neighboring nodes, calculating its reputation value. There have several models developed that focuses on a nodes reputation. Three examples that we will discuss are: CONFIDANT [3], CORE [4, 7], and OCEAN [9].

B.1. CONFIDANT

In Dynamic Ad-hoc Network, the assessment of nodes’ reputation makes misbehavior of the nodes unattractive [3]. This reputation model focuses on finding the uncooperative nodes and isolating them from the network. Doing this makes the node unattractive, making other nodes want to participate in the network; since no node wants to be established as unattractive. Every node watches the activity of their neighbors, reporting any unusual behavior to a reputation system. If an unusual event occurred too many times, the reputation system deletes all of the routes containing the misbehaving node and notifies its neighbors.

B.2. CORE

In Collaborative Reputation, or CORE, a node receives its reputation based on their contribution to the network [4]. A node that has good reputation, always forwarding other nodes’ packets, it can use the resources of the other nodes. However, nodes that don’t cooperate in the forwarding of packets often get excluded from the network, not being able to send their own packets. There are the watchdog mechanism. Every node sends a watchdog mechanism that watches the next node’s transmission to verify that it forwards the packet, and a reputation table that contains the information based from the watchdog mechanism.

B.3. OCEAN

In OCEAN [9], which stands for Observation-based Cooperation Enforcement in Ad-hoc Networks, every node starts off with the rating of zero (0). Every time a node cooperates in the network, their rating increments by plus one (+1). If a node does not forward a packet, their rating goes down by minus two (-2). The node is added to a misbehaving list if it reaches the set threshold, which is negative forty (-40). Traffic that comes from any misbehaving nodes will be rejected; however, they do receive a second-chance after a certain timeout period. The rating does not change, so that if it continues to misbehave, it can be quickly added back onto the faulty list.

B.4. Classes

Reputation based schemes can be broken into three classes: Trust vs. Reputation, Direct vs. Indirect Trust, and Global vs. Local Reputation.

- Trust vs. Reputation [10]: Trust represents the honesty of a node, deciding if it is trustworthy to
forward packets, while reputation represents the behavior of a node, deciding if it is misbehaving or cooperative.

- **Direct vs. Indirect Reputation**: Direct reputation occurs when a node observes the behavior of a direct neighboring node (one-hop), while indirect reputation occurs when a node receives information about a nodes’ reputation from the other nodes in the network.

- **Global vs. Local Reputation** [11]: In global reputation, the reputation of a node is shared throughout the network indirectly, while in local reputation, the reputation information of a node is solely based on the direct observation of a neighboring node. CONFIDANT and CORE are examples of both local and global reputation, while OCEAN is an example of only local reputation.

C. Community Enforcement Mechanism

A functional network should include full cooperation of nodes within the network. To achieve this, a few characteristics should be present in the network, which are [12]:

- **Autonomy**: Every node in the network has the option to cooperate.
- **Local Views**: Every node should observe the behavior of its neighboring nodes regularly and directly [11].
- **Decentralization**: The network is self-configured, meaning that it does not include a central authority.
- **Mobility**: The nodes in the network can be either mobile or static.

Based on the characteristics presented above, a community enforcement mechanism would be best suitable for a MANET. With this mechanism, the nodes are able to monitor each other’s behavior. A node is considered to be misbehaving if it does not participate in the forwarding of a packet. In one strategy discussed about the community enforcement mechanism [12], there are two types of nodes that exist in a network, a node type “c”, which is not a defected node, and a node type “d”, which is a defected node. Every node uses a contagious strategy, where if a node with no defection (type c) identifies a defected node (type d) in its transmission range, it will also become a defected node. This ultimately leads to a complete network failure, since every node will become defected. This forces nodes to cooperate because a node does not want to be the one that starts the process of contaminating the network, ruining their future gain.

III. PROPOSED ALGORITHM

The community enforcement mechanism considers a node to be misbehaving if it does not cooperate. A node does not cooperate because of early depletion of their battery life for forwarding other’s packets. We propose an algorithm called the Measured Defection strategy, which is developed with the attention to the level of depletion of the battery of each node. It is considered that each node has an initial battery power X, and the energy that it spends to forward one packet is Y. Hence, a node is able to forward or send in its life time X/Y packets.

The longevity for a network can be achieved by preventing the quick depletion of an individual node. Therefore, a threshold value, Z, is set for the battery power of a node. In which, a node can send/forward packets as long as the battery power of a node does not fall below the threshold value Z. If the battery power of a node falls below the threshold Z, then the node does not need to cooperate with the other nodes for forwarding their packets. Therefore, this node will not be considered defective/selfish by the other nodes. The neighbor nodes instead will find an alternative route for forwarding their packets. This agreement among the nodes helps to maintain the battery life of a node and therefore such nodes do not affect the life of entire network. Eventually, after a certain point of time, most of the nodes energy level will be depleted, causing the network to break down. Although the network eventually fails, its lifetime will effectively become extended.

The Measured Defection strategy proved to be a beneficial progression of the Community Enforcement Mechanism; however, the internal nodes of the network are used much more frequently than nodes lying on the outer boundary. Consequently, they may not be able to send packets of their own since their battery life is being diminished faster than the other nodes. In order for a network to have the best performance, each node’s battery power should be used discreetly.

IV. SIMULATION

This experiment was performed in the Network Simulator-3 (NS-3), applying the AODV routing protocol to the node cooperation.

A. Network Simulator-3

NS-3 is a discrete event simulator that is written in C++. It was developed for educational purposes, primarily used for research involving networking. It is a refurbished version of the NS-2 simulator. The NS-2 is written using two languages, C++ and Object Oriented Tool Command Language (OTcl). The combination of two languages makes it difficult to correct possible errors that lie within the coding, and therefore it is not appropriate for large simulations, as it does not scale well. The NS-3 simulator was best suited for the development of the experiment. However, it is not backward compatible with its predecessor, the NS2 simulator. It contains a Random Waypoint Mobility model that allows nodes to move throughout the network freely, demonstrating a realistic effect of Mobile Ad Hoc Networks. The NS3 supports the use of a pyviz python simulator for viewing the actual flow of packets in a network. The NS3 also provides support for the global routing protocols and OLSR; various types of internet stacks such as Ipv4, Ipv6 and transport protocols like TCP, UDP implementations along with the support for implementation of some NetDevice such as WIFI, CSMA, etc. Some of the key features supported by the NS3 are: Virtualization, Tracing Model
and Statistical Framework, Object Aggregation Model, Attribute System, Updated Models.

B. AODV Routing Protocol

AODV or Ad Hoc On-Demand Distance Vector is a common routing protocol used for mobile and wireless ad hoc networks. This routing protocol finds a route from the source to destination as needed (on demand). A broadcast system is used in order to establish a route between the source node and the destination node (route discovery function). A source node broadcasts RREQ messages to its neighbors [13]. This process repeats with the intermediate nodes. If the node is the intended destination, it replies sending back a RREP message. Once the RREP message follows the path back to source node, the route has been established and the packet can be sent. Since wireless networks are susceptible to link breakages, the responsibility of the route maintenance function is to discover if any link between nodes is broken while a RREQ or RREP is in process, a RERR message is sent to the source node if a timeout occurs and another route has to be established.

C. Environment

The simulation area was set to be 800m x 800m on a plain surface. 80 nodes were randomly placed all over the simulation area. As explained above, the Random Waypoint Mobility model was applied to the simulation. Therefore, the nodes were able to move through the area at different speeds of 1, 2, 5, 10, and 20 m/sec. When a node moves toward the boundary line, it bounces off and continues to move throughout the area. The communication range between the nodes was set to 128m, meaning they are considered neighboring nodes if they are 128m or less next to each other. However, if they are more than 128m, a multi-hop is required. A simulation lasts for 1000 seconds. In each simulation run, 20 random nodes were selected as the source nodes, and each of these nodes sends 5 packets to a destination. For each source node, five destination nodes were selected randomly within the network. So a total of 100 connections were established with both source and destination chosen randomly.

D. Results and Analysis

The Measured Defection strategy is assumed to elongate the network when compared to the community enforcement mechanism. The Measured Defection strategy focuses on the battery life of nodes, and with the community enforcement mechanism, a node is turned defected if a defected node is in its vicinity. Figure 1 is a comparison between the community enforcement mechanism and the Measured Defection strategy. It clearly shows that within the time span of 15 seconds, the community enforcement mechanism has all 80 nodes defected, however only 24 nodes are defected in Measured Defection strategy. As Figure 1 shows, our proposed algorithm has increased the life of the network.

Figure 2 also shows the minimum, average and maximum time for defection of nodes from the whole network from the time that the first node was defected.

V. CONCLUSION

In this paper, we analyzed and tested node cooperation pertaining mobile ad-hoc networks using the NS-3 simulator. We studied the network behavior containing the community enforcement mechanism and proposed an algorithm, which can prolong the life of an ad-hoc network. Based on the results from our simulation, this algorithm shows fewer nodes defection in comparison with the community enforcement mechanism in the same amount of time, hence increasing the lifetime of the network.

ACKNOWLEDGMENT

The authors wish to thank Divya Rani Bhat and Yogesh Sanghi Lamar Students for the implementation of the algorithm.

REFERENCES


