Wireless Personal Communications manuscript No.
(will be inserted by the editor)

Stimulating Node Cooperation in Mobile Ad hoc Networks

Jamal N. Al-Karaki[1], Ahmed E. Kamal[2]

[1] Dept. of Electrical and Computer Engineering
The Hashemite University
Zarka 13115, JORDAN
Email: jkaraki@hu.edu.jo
[2] Dept. of Electrical and Computer Engineering
Iowa State University
Ames, IA 50010, USA

Abstract Mobile Ad hoc Networks (MANETs) rely on the cooperation of nodes for packet routing and forwarding. However, much of the existing work in MANETs assume that mobile nodes (possibly owned by selfish users) will follow prescribed protocols without deviation. However, a user may misbehave due to several advantages resulting from noncooperation, the most obvious being power saving. As such, the network availability is severely endangered. Hence, enforcing the cooperation among nodes becomes a very important issue. Several different approaches have been developed to detect non-cooperative nodes or deal with the non-cooperative behavior of mobile nodes in MANETs. These protocols are first surveyed in details in this paper. It is found that the proposed approaches have several concerns that prevent them from really enforcing the node cooperation in MANETs. Thus, a new scheme that can stimulate and also enforce nodes to cooperate in a selfish ad hoc environment is presented. We also present a mechanism to detect and exclude potential threats of selfish mobile nodes. The simulation results indicate that by using the proposed scheme, MANETs can be robust against nodes' misbehaving and the performance of the network is enhanced many folds when compared to other existing schemes.

1 Introduction

Mobile Ad hoc Networks (MANETs) are distinguished from other communication networks by many features. First, mobile nodes in MANETs may move freely in the absence of a fixed infrastructure. Therefore, frequent changes in routes may happen due to unpredictable topology changes and link disconnections. Second, nodes in MANETs have limited resources such as energy, bandwidth, and computational power. Finally, MANETs have no trusted centralized authority. Hence, security mechanisms should be distributed and should not cause unwanted resource consumption. Nevertheless, MANETs have a wide array of military and commercial applications [1]. MANETs are typically self-organized networks and intermediate nodes should carry the end-to-end communication. To achieve this, each node relies on its neighbor to forward the packet to the destination. In fact, most of previous studies on MANETs has implicitly assumed that nodes are cooperative As such, the issue of *node cooperation* becomes very important in MANETs. However, cooperation may be harder to enforce in MANETs than in infrastructure-based networks due to many reasons. First, nodes can arbitrarily join or leave the network. Second, detection of misbehavior and subsequent isolation of a misbehaved node has to work in a distributed fashion due to lack of centralized control. Finally, user specific requirements or attitude should not be ignored. Some users view their energy resource as being limited by battery life, and hence they may not feel inclined to relay traffic for other users. As such, user's behavior will impact the system performance driven by his application needs or physical constraints.

In general, uncooperative nodes in MANETs may be classified into two classes: malicious nodes and selfish nodes. The term *malicious* refers to the group of nodes that intentionally try to attack the system or break the network. On the other hand, the term *selfish* refers to the nodes that try to gain help from the network without willing to pay back the help received. Both malicious and selfish nodes are considered as misbehaving nodes [15].

A reasonable question then is how to motivate or enforce nodes to cooperate or how to discover misbehaving nodes and isolate them from the network.

In this paper, we address the problem of enforcing cooperation in MANETs. In particular, a scheme that can stimulate or enforce the cooperation among nodes in MANETs is presented. There are basically two methods to enforce a desirable strategy in MANETs: either punishing misbehaving nodes or encouraging those who adopt it. We believe that in the case of MANETs, it is better to consider punishment based techniques for two reasons. First, rewarding strategies are not easily extensible to scenarios in which malicious entities are active. Second, malicious nodes may not be interested in rewards. Instead, when they are susceptible to punishments (i.e., being excluded from network), they tend to behave well. Moreover, the boundary between a punishment and a reward is extremely unclear, i.e., a reward could be a punishment, or a punishment can be a reward [15].

The proposed scheme can provide fairness as well as a high level of cooperation among nodes based on a combined approach. Although the proposed scheme has some similarities with previous schemes, but it differs in the way it works and the qualities it provides. Moreover, the proposed scheme addresses the concerns raised from previous schemes that will be studied in the next section. We study the performance of our scheme through simulations and compare it to other schemes. Our results show that the scheme can enforce the cooperation among selfish users on two levels: local and global. In this context, local refers to a group of nodes in neighborhood, while global refers to the group of whole nodes in the network.

The rest of this paper is organized as follows. In the next section, we present a detailed overview of previous studies on the issue of cooperation in MANETs. The section can serve as a comprehensive survey on this topic by itself. In section 3, the network model used in this paper is presented. The local cooperation scheme is presented in section 4. A hybrid scheme that enforces global cooperation in MANETs is presented in section 5. In section 6, the simulation results are presented. We conclude the paper with final conclusions in section 7.

2 Related Work

The problem of non-cooperative mobile nodes in MANETs has been addressed in few works [12–16,6–8]. In [13], non-cooperative nodes are viewed as malicious, and methods to identify misbehaving users and to avoid routing through these nodes are presented. In [12,14,15], simple rules are used to determine on a packet-by-packet basis whether a user should forward other nodes traffic or not. In particular, in [12,14], the authors introduce a virtual currency called nuglets. Every network node has an initial stock of nuglets. Either the source or the destination of each traffic connection use nuglets to pay the relay nodes for forwarding data traffic. Packets sent by or destined to nodes that do not have a sufficient amount of nuglets are discarded. Notice that by allowing source or destination to charge can under- or over-estimate the packet price. Moreover, both proposals require a tamper-proof hardware at each node so that the correct amount of credit is added or deducted from the node. As a result of this requirement, both proposals may not find wide-spread acceptance. In [15], source nodes pay as many battery units as the estimated number of nodes on the path to the destination, and makes relay nodes earn as many battery units as the number of forwarded packets. In [16], a game-theoretic approach for routing in MANETs, called Ad hoc Vickrey, Clarke, and Groves (Ad hoc-VCG) that consists of greedy and selfish agents was considered. Those agents accept payments for forwarding data for other agents if the payments cover their individual costs incurred by forwarding data. Ad hoc-VCG is a reactive routing protocol, which may guarantee that routing is done along the most cost-efficient path by paying to the intermediate nodes a premium over their actual costs for forwarding data packets.

In general, mechanisms that try to mitigate and stimulate the misbehaved or uncooperative node can be classified into two classes [4]: (a) *Virtual Currency based schemes*, which uses some incentive to motivate nodes to cooperate. That is, the node will get some incentive if it serves the network and pays back some price when it gains help from the network, and (b) *Reputation based schemes*, which uses the node's reputation or behavior to mitigate the selfish node behavior. The nodes' reputation can be obtained using direct observation or from reputation messages from other nodes in the network. In the following, we illustrate these two classes in more details.

2.1 Virtual Currency based schemes

The idea of virtual currency stems from the fact that mobile nodes in MANETs have a limited battery, so when a node forwards a packet to another node, a price must be paid. Hence, a virtual currency is used to charge/reward

the packet forwarding service. The virtual currency system must compensate a node that cooperate in order to motivate this node for future cooperation. The system uses a credit or micro payments to reward for the services that a node does. Therefore, a node receives a virtual payment for forwarding the message of another node, and this payment is deducted from the sender or from the destination. We illustrate two examples of protocols based on the virtual currency concept, namely, Nuglets and Sprite.

2.1.1 Nuglets The virtual currency in this protocol is called nuglets [6] which is used to charge/reward the packet forwarding service. A credit counter can implement the Nuglets. When a node want to send a packet it must have a counter value that is at least equal to the route hop count. When the node is the source the counter value is decreased by the hop count. When an intermediate node forwards a packets, its counter value is incremented by one. There are two models in Nuglets [6,14]: (i) Packet Purse Model: in which the credit payment is deducted from the source. (ii) Packet Trade Model: in which the credit payment is deducted from the destination. The major problem in Nuglets is that it needs a tamper-proof hardware to manage the increments and decrements of the credit counter for each node. That is, each node has legitimate counter value.

2.1.2 Sprite This protocol [8] uses a Credit Clearance Service (CCS) that manages the rewards and the credit payments for each node. A node that tries to forward a message is compensated, but the credit that 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 [4,8]. The problem with this approach is that it needs a centralized server to manage the rewards and the credit payment for each node in the network, and this requirement does not meet many practical ad hoc scenario.

2.2 Reputation based Schemes

Reputation mechanisms are based on the behavior of a node in the network. Each node has a reputation value that reflects its behavior. This value is stored and calculated by other nodes that watch its behavior. As such, the mechanisms needed should take care of [4]: the calculation and update of reputation values, the detection of misbehavior, and the reaction to uncooperative behavior. Some of the key points that need to be addressed under this class are:

- Trust vs. Reputation: *Reputation rating* represents how well a node behaves, and is used to decide whether the node is cooperative or misbehaving. On the other hand, *trust rating* represents how honest a node is, used to decide whether the node is trustworthy or not, thus the indirect reputation message from the node is accepted or not.
- Direct vs. Indirect Trust (Reputation): *Direct Reputation (First Hand Information)* is obtained by direct observation. A node monitors the behavior of other nodes usually in one-hop to see if it works well. On the other hand, *Indirect Reputation (Second Hand Information)* obtains reputation information about a node from other nodes in the network. The acceptance or rejection of this information is based on the trust level of the sender node.
- Global vs. Local Reputation: *Global reputation* refers to the case where every node knows the reputation of every other node in the network. This is achieved by exchanging indirect reputation messages among the network. In *local reputation*, however, information is based only on direct observations of one-hop neighbors. Any second-hand reputation exchanges are disallowed.

In general reputation mechanisms can be classified into two classes: (i) the reputation value is updated based on both local and global reputation information such as the case in CONFIDANT and the CORE protocols that will be discussed in the next subsection, (ii). In the second class, updates of the reputation value is based on local reputation information only such as the case of OCEAN and LARS discussed in section 2.4.

2.3 Global Reputation Protocols

Each node in this class updates reputation value using two information: direct observation and valid reputation message from other nodes.

1. CONFIDANT-Cooperation Of Nodes and Fairness In Dynamic Ad-hoc Network: CONFIDANT [6] aims at detecting and isolating uncooperative nodes, thus making it unattractive to deny cooperation. Each node in the CONFIDANT contains several components that is responsible for monitoring and rating the behavior of the neighbor nodes (one-hop) by listening to the transmission of the neighbor node. Each node monitors the behavior of its neighbor. If it detects unusual behavior, it reports it and sends it to a reputation system. The Reputation System check to see whether the event has occurred more often than a predefined threshold. If a certain threshold is exceeded, the reputation system updates the rating of the node that caused the event. If the rating turns out to be intolerable, the information is relayed to the path manager component, which proceeds to delete all routes containing the misbehaving node from the path cache and tell the trust manager component to send ALARM message to its friend list.
2. CORE - COllaborative REputation: CORE [7] is a generic mechanism that can be integrated with many network functions, e.g., packet forwarding. CORE stimulates node cooperation by using a collaborative monitoring technique and a reputation mechanism. In this mechanism, reputation is a measure of someone's contribution to network operations. Each network entity in CORE keeps track of other entities' collaboration using a technique called reputation. An interesting feature of the CORE mechanism is that a denial of service attacks based on malicious broadcasting of negative ratings for legitimate nodes is prevented. Furthermore, each node in the network has two components: (a) Watchdog: which monitors the behavior of the next node. It works in the promiscuously mode to see if the neighbor or next node correctly execute the function that assign to it such as forwarding a packet. There are two entities, the requester that apply the watchdog and the provider that execute the function. The requester updates the reputation value of the provider based on his behavior, and (b) The Reputation Table: a data structure stored in each node. Each row of the table consists of four entries: the unique identifier of the entity, a collection of recent subjective observations made on that entity's behavior, a list of the recent indirect reputation values provided by other entities and the value of the reputation evaluated for a predefined function. The reputation table updates its entries based on the result generated by the Watchdog.

2.4 Local Reputation Protocols

In this class, only local reputation information is used to update the reputation value for the neighbor node. The protocols assume that each node knows its neighbors and maintains a reputation value for each one.

1. OCEAN-Observation-based Cooperation Enforcement in Ad hoc Networks: OCEAN [9] is a layer that resides between the network and MAC layers of the protocol stack, and it helps nodes make intelligent routing and forwarding decisions. Its design on top of DSR and contains the following components: (a) Neighbor-Watch: monitor the behavior of the neighbor node. Whenever misbehavior is detected, Neighbor-Watch reports to the Route-Ranker, which maintains ratings of the neighbor nodes, (b) Route Ranker: maintains a rating for each of its neighboring nodes. Each node is initialized to Neutral (0), every positive behavior resulting in an increment (+1) of the rating, and every negative behavior resulting in a decrement (-2) of the rating. Once the rating of a node reaches below a certain misbehaved threshold (-40), the node is added to a misbehaved list, (c) Ranked-Based Routing: keeps track of the rating value resulting from the Route-Ranker to a void some route contains misbehaved nodes, (d) Malicious Traffic Rejection: rejects any traffic from misbehaved nodes, and (e) Second Chance Mechanism: gives another chance to the node that misbehaved to return to become cooperative node. A timeout approach is used where a misbehaving node is removed from the faulty list after a fixed period of inactivity. Even though the node is removed from the faulty list, its rating is not increased, so that it can quickly be added back to the faulty list if it continues the misbehavior.
2. LARS-Locally Aware Reputation System: The protocols in LARS [4] define three level of trustiness, T, which is based on a reputation value called R, and as follows:

$$T = \begin{cases} 1 & , \quad R_t < R < R_{max} (\text{trustworthy node}) \\ -1 & , \quad R_{min} < R < R_u (\text{untrustworthy node}) \\ 0 & , \quad R_u < R < R_t (\text{undecided node}) \end{cases}$$

where R_t and R_u are the trusted and untrusted reputation values, respectively, while R_{min} and R_{max} are the boundaries of R. In fact, LARS is based on direct observation. That is, if the reputation value of a neighbor node, called M, with respect to a certain node X drops below the untrustworthy threshold R_u, then M is considered as a misbehaving node by node X. After that, node X will notify its neighbors about M's misbehavior by

initiating a WARNING message. To trust the WARNING message it should be signed by m nodes before it can be broadcasted to the k-hop neighborhood, where k is the number of nodes in neighborhood to node X and m is a subset of k. LARS uses different weights when updating the reputation value. When a node forwards a packet, it's reputation value increases by μ, while if it discards the packet, its value is decreased by s where $s > \mu$.

3. Probability Route - Toward Reliable Forward: In this approach [10], a node is responsible not only for forwarding a packet, but it shall forward it on the route that maximizes its success probability. The framework based on reliability indices. Every node has a dynamically updated reliability table containing a value for every outgoing link to a neighbor (one-hop). This value indicates the reliability of a route path starting from this neighbor node. Every time the node sends a packet on a path, it updates the reliability value associated to the neighbor through whom the packet has passed: the updating is positive whenever source node receives an acknowledgment from destination, negative otherwise. The reliability value is unique for all paths rooted on that neighbor. If source node observes that the reliability index of that subtree decreases, then it should immediately reduce the traffic sent through that neighbor, by preferring routes passing through a neighbor with a higher reliability index.

Overall, the two reputation-based approaches, namely, the global and local schemes can be contrasted along several issues:

1. *Table Size:* Global Reputations: each node maintains reputation values of every other node, so the size is $O(N)$. Local Reputation: each node maintains reputation values of the neighbor node that is located in one-hop.
2. *Network Traffic:* Disseminate reputation information greatly increases the volume of network traffic. In global reputations, the reputation message distributed during each reputation disseminate period is $O(N^2)$. In Local reputation, there is no second hand information. Therefore, local reputation has better performance due to less network traffic.
3. *Overhead Computation:* Global reputation needs an additional computational overhead to decide whether to accept or reject a warning message and to update the reputation table.
4. *False Accusations:* Local reputations are less vulnerable to false accusations than global reputations because it uses direct observation.
5. *Reliability:* global reputation message traverse across the network so it could be delayed, modified, replayed or accidentally lost during transmission.
6. *Mobility:* Global reputation has better performance with respect to the mobility issue, because every node knows the behavior of other node in the network so possibility to cheat is less.

In general, local reputation has less cost, more reliable and more efficient than global reputation [4].

2.5 Performance Concerns of Related work

It is trustworthy to note that most of the approaches discussed in the previous section share the following critical concerns, which still need to be solved:

1. The packet-by-packet paying system imply a significant communication overhead and implementation complexity.
2. They all assume the users are of the same behavior or of the same class.
3. They did not pay attention to the fairness issue in routing when some nodes do not get any reward due to some reasons, e.g., location,
4. None of these approaches dealt with cluster-based networks where the role of clusterhead is critical for the operation of the network as a whole.
5. Some proposals require tamper-proof hardware at nodes, while others assume mobile nodes have secure access to a trusted third party through Internet, for example.

The protocol developed in this paper and described in section 5 will try to address the above concerns in one way or another.

3 Network Model

In this section, the network model used in this paper is presented. We consider a MANET with N mobile nodes randomly distributed in a two-dimensional network area A in square meters. All nodes are equipped with batteries with possibly different levels of energy density. We assume that each node has a unique, persistent, and distinct identity. Each node knows its one-hop neighbors. We also assume that links between nodes are bidirectional. The MAC layer is the IEEE 802.11 standard. For the sake of simplicity and robustness, a virtual backbone can be imposed on top of the physical backbone of the network. An efficient approach for building such a virtual backbone was presented in [2]. The virtual backbone, called Virtual Grid Architecture (VGA), will be used in this paper as the underlying backbone for performing the cooperative routing in MANETs. VGA consists of a collection of nodes, called clusterheads (CHs), that are elected based on an efficient eligibility criteria detailed in [2]. As the network area is divided into fixed-size square zones, each CH is a representative of the group of nodes inside its square zone. In fact, VGA was shown to perform well when compared to optimal clustering [3]. Figure 1 shows how VGA is constructed in both homogeneous (e.g., identical transmission ranges) as well as heterogeneous (e.g., variable transmission ranges) MANETs for the case when the number of zones is 16. In Figure 1(a), all nodes are identical and hence the zoning process and CH election will be simple. This is not the case for heterogeneous nodes, where Figure 1(b) shows the case of zoning and CH election for only two types of nodes, namely, short and long range transmission nodes. The interested reader is encouraged to read [2] for more details about VGA clustering approach.

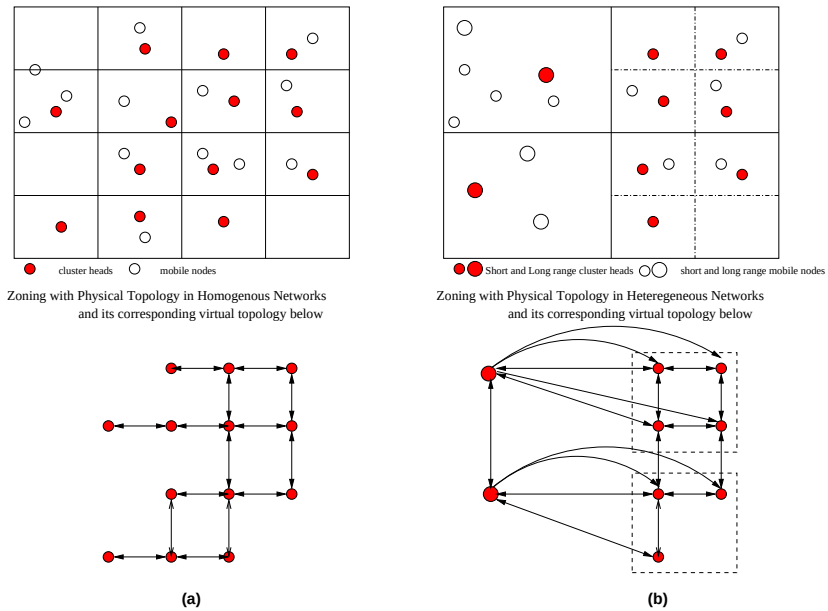

Fig. 1 An example of fixed zoning process in VGA: (a) homogeneous networks and (b) heterogeneous networks.

In [2], the CH election process of VGA was based on the assumption of cooperative environment, in which all mobile nodes participate toward the CH election process. When the cooperative assumption is relaxed, such as in the case of selfish users, a conflict emerges between local energy conservation and network operation. In the next section, we modify the election algorithm to take into consideration the selfishness of nodes toward the CH election.

4 Local Cooperation: A stimulating Mechanism for VGA CH Election

A rational mobile node may prefer to avoid network participation, i.e., act as a CH, in order to better satisfy its energy performance goals. If all mobile nodes in a zone employ this reasoning, then the CH election protocol might fail. Moreover, if no user cooperates in the CH election in a certain zone, it will result in a loss in connectivity of this zone to the rest of the network. The question is how can we stimulate the user or enforce him to cooperate.

The proposed scheme is a reputation-based one that stimulates and enforces nodes to cooperate at two levels: intra-zone (CH election process) and inter-zone (relaying traffic to other zones). In both cases, the decision of a mobile node to cooperate depends on many parameters, e.g., node's location, node's energy constraints, node's mobility pattern, and its particular needs. Therefore, we associate with each node a parameter, called *willingness*, that refers to the node's willingness to cooperate. The willingness of node i to act as CH at period p, denoted by, w_i^p, takes the values [0,1], where a value of 0 refers to a completely selfish node while a value of 1 refers to a completely cooperative node. Note that the value of w_i^p can be a constant or it can be dynamically set by each node at the beginning of each election period or at the beginning of a new routing session. The value of w_i^p may depend on the status of that node at the time of decision, e.g., its current energy budget and its attitude.

Our objective is to study the impact of the behavior of selfish nodes on the system performance reflected by the CH election process and the traffic relaying process. In the CH election process, we want to stimulate the most eligible node to always accept the role of CH if it wishes to maximize its throughput. Let Z be the set of resulting zones. Let $N(z)$ be the number of nodes in zone z, $z \in Z$. Let E_i^p be the power spent per byte by node i in transmitting packets for other nodes in its zone during period p. We assume that each node has a memory of whether it has been helped earlier by each node in the zone. Hence, a node will always remember the favor done to it by other nodes in the zone, as well as the selfishness of other nodes. To do that, we associate with each node two more parameters: *take* and *give*. The parameter *take* intuitively reflects the amount of help that a node has received from other nodes in the zone relaying its messages. On the other hand, the parameter *give* reflects the amount of help that the node has rendered in relaying messages for other nodes in the zone while being a CH. The proposed algorithm attempts to balance the amount of takes and gives at each node in the network. Given a value of w_i^p, a node i is more willing to act as CH if it has received more help than it has given. Conversely, if a node has been generous in the past without receiving a commensurate amount of assistance from other nodes, then it is inclined to reject the CH role. As the value of w_i^p decreases, nodes tend to behave more selfishly.

4.0.1 Intra-Zone Cooperation (CH election) The objective of intra-zone cooperation process is to ensure that the most eligible node will always accept the CH role. When a node serves as a CH, it will maintain a table with entries for all nodes that have received help (node IDs), the amount of the received help (take), and the amount of the rendered help in previous periods (give). At the end of the current election period, the CH will broadcast an INFO message that contains the information in this table to all nodes in the zone. The values in this table will be used to calculate the eligibility factor for the next period as follows. At the beginning of the election process, each node in the zone will receive the EF values of all other nodes. Before doing the comparison, the value of the EF received from the generic ith node will be modified in accordance with the difference between the amount of help the node received and the amount of help the node rendered in the network, respectively, during the previous period(s). Let $take(i,p)$ and $give(i,p)$ be the amount of take/give parameters of the generic i-th node at the beginning of period p. Then, the new eligibility factor will be calculated as,

$$EF_j = EF_j * (take(j,p) - give(j,p)) \quad j \neq i, j \in N(z) \tag{1}$$

The values of take and give parameters are received in each INFO message. The second term will increase the value of EF for those nodes who received more help than what they give, and hence need to help in turn. In order to ensure that these nodes who have been acting selfishly will not receive more help, a CH node will forward traffic for non-CH node i in its zone if the following condition is met:

$$give(i,p) > (1 - w_i^p).\{take(i,p) + Msg\_size * E_i^p\} \tag{2}$$

where Msg_size is the size of message to be transmitted in bytes. The above condition forces nodes to always try to provide help (i.e., have credit) in order for its packets to be forwarded in turn. This credit will be obtained if the node has served as a CH in the previous periods.

4.0.2 Inter-Zone Cooperation (Relay traffic): The objective of inter-zone cooperation process is to ensure that intermediate CH nodes will always help in relaying traffic for other CH nodes in the network. Assume that a session is generated between a source-destination CH pair (s-d) and that the set of available routes between this pair is $\mathcal{R}$. Consider the minimum energy cost path $r \in \mathcal{R}$. Let $e(k,r)$ be the amount of energy per byte spent by node k on the route r in relaying traffic to the next node on r. Let $t(k,r)$ and $g(k,r)$ be the amount of take/give parameters of the generic k-th node in route r. Let $N(r)$ be the set of CH nodes on the route r, respectively. When a source node s initiates a request and the request is accepted by the relay nodes on r, the node can forward its traffic for

the whole session. For each session accepted for any source node s on the route r, the ith relay node $i \in N(r)$ will store the amount of help it gives to node s as follows:

$$give(i,r) = give(i,r) + \{Msg\_size * e(i,r)\},\ \forall i \in N(r) \tag{3}$$

At the end of a session, each relay node will attach to the last ACK message, that is sent back to the source node, the value of the amount of help the node rendered in forwarding the traffic of this session. Accordingly, the source CH node will update its take parameter as follows:

$$take(s,r) = take(s,r) + \sum_{i \in N(r)} give(i,r) \tag{4}$$

A relay CH node i will accept forwarding traffic for a source CH node s if the node received help from the network more or equal to what it gives, i.e., the following condition should be met:

$$take(i,r) - give(i,r) >= (1 - w_i^r).\{Msg\_size * e(i,r)\} \tag{5}$$

Hence, each node will be trying to repay the network whatever it takes from the network (debt). As a result, nodes are stimulated or enforced to cooperate for their own best interest.

5 Global Cooperation: A Hybrid Approach for Stimulating Node Cooperation in MANETs

In this section, we present a novel scheme that enforces node cooperation in MANETs. Recall that virtual currency schemes provide more fairness than the reputation-based schemes. However, the cooperation between nodes in reputation-based schemes is better. The model we present in this section tries to combine both features. When a node enters a new zone of VGA, it broadcasts a message to identify itself to the node in that zone and to know the neighboring nodes. Hence, a node can identify its neighbors.

The model is built on top of the well-known Dynamic Source Routing (DSR) which in turn operate on top of VGA. Further, we assume that each node fit in one of four categories of reputation values (R). These are:

- *New node:* when a new node joins the network for the first time, it is assigned an initial value that is equal to 0.
- *Normal node:* a cooperative node that has a reputation value R where $0 < R < T_s$
- *Super node:* a cooperative node that has a reputation value R where $R > T_s$.
- *Misbehaving node:* a selfish node that have a reputation value R where $R < T_m$

where T_s and T_m are edge values for super cooperative and misbehaving nodes, respectively. Note that the super node is needed to indicate those nodes that will play much more active role than others because of their location with respect to other nodes, for example. In the next section, we present an overview of the functionality of the proposed protocol.

5.1 Protocol Overview

The protocol is distributed in the sense that all nodes are aware of the behavior of all other nodes. Each node monitors the behavior of its neighbor using a direct observation and a valid reputation message. Each node maintains a table that describes the behavior of each of its neighbors. Each record in the table describes one neighbor and contains information about this neighbor such as its *Total Help* (TH) that was provided to other nodes in the network. The records of the tables are updated as follows. When a source node A initiates its session through a route of intermediate nodes[1], the first intermediate node forwards the packet and all nodes that are neighbors to this node increment the TH field related to it by a value of one and at the same time decrement the TH for node A based on the acknowledgment (Ack) packet return. If the intermediate node drops the packet, its TH is decremented by a value of two. This process will apply to all nodes participating in the forwarding function. The accumulation of the reputation values continues to build as nodes participate or refuse to participate in the network activities.

If the behavior of a node falls below certain threshold, namely, T_m, the neighbor node becomes a *Misbehaving* node and all nodes will inform each other about this node so that nodes will not cooperate with it in the future. In the following, we discuss the protocol in details. Each node in the system has the following components (shown in Figure 2):

[1] The route is selected through a path selector component in the node such that the probability of discarding the packet is a minimum.

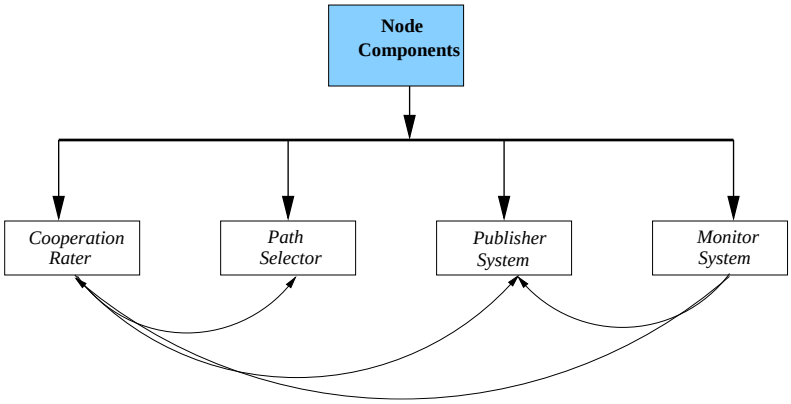

Fig. 2 The node components in the proposed protocol.

Node−ID	My Count	Node Total Help	N or F

Fig. 3 The structure of a table of the cooperation rater.

1. **Cooperation Rater:** each node maintains a data structure or table that describes the behavior of its neighbors and some foreign nodes (see Figure 3). Each record in the table contains the following information:
   Node ID: Unique identifier of the node.
   My Count: An integer which represents (the amount of help given - help received) by this node.
   Node Total Help: Total help a node gave to its neighbors.
   N or F: refers to Neighbor (one-hop)/ Foreign node, respectively.
2. **Path Selector:** If there are multiple routes between the source and the destination, then the Path Selector will select the path that has less probability for discarding packets enroute. The selection is based on the rating table formed by the cooperation rater. The Path Selector will choose the route that begins with a neighbor that has a minimum Total Help and at the same time gained the maximum help from the source. The Path Selector will also make load balancing between the outgoing paths based on several parameters such as hop count, Total Help of the neighbor, and the help a neighbor gained from the source. Path Selector uses Round Robin algorithm and gives a different weight for each route to maximize throughput. Load balancing applies only when a huge data is to be sent to a certain destination. Note that the Path Selector executes only at the source node.
3. **Publisher System:** The Publisher System monitors the rating of the neighboring nodes. If the rate of a neighbor node X falls below a certain threshold, namely, T_m, the Publisher System at any node that is a one-hop from X and has this observation, will broadcast a message to tell other nodes that node X is a Misbehaving node. If the rate of a neighbor node X less than T_s for a certain time, then the Publisher System at any node that is a one-hop from X will broadcast a message to tell other nodes that node X is a Normal node (used when a node changes status from misbehaving node to a normal node). If the rate of a neighbor node X exceeds T_s for a certain time then the Publisher System at any node that is at one-hop from X broadcasts message to tell other nodes that node X is a super node. The message contains the ID of the informing node as well as the list of the nodes in neighborhood of X.
   The Publisher System also processes all incoming reputation message in a certain node. Recall that the received massage may carry information about a node either being a neighbor or a foreign node. If the node is local, i.e., in the neighborhood, then the above procedure apply. While if the information is about a foreign node, the system performs the following operations:
   (a) If the message is coming from a node with low reputation value, i.e., a misbehaving node, the system will reject the message directly.
   (b) If there is no record for the foreign node, the system creates a new record.
   (c) To update the record of the new node, the system must receive at least three messages from nodes that are neighbors to this node. This rule applies to both types of nodes, namely, neighbor and foreign nodes.

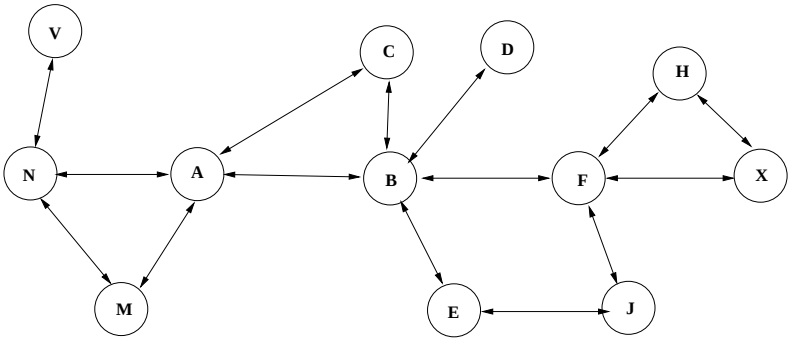

Fig. 4 Network structure used in the example.

(d) The reputation value of the foreign node is set based on the information contained in the message. If the message informs about a misbehaving node, the reputation value is set to T_m. If the message informs about a normal node, the reputation value is set to 0, i.e., a new node. Finally, if the message informs about super node, the reputation value is set to $T_s/2$. These values were chosen based on experimentation.
(e) If the message received is always being from a misbehaving node, then give an incentive to this node to be cooperative node and set T_m for that node to be T_m+5.

4. **Monitor System:** each node has a Monitor System that does the following:
   (a) Monitors the behavior of its neighbors and reports to the Cooperation Rater any usual or unusual behavior.
   (b) Forwards any incoming Reputation Message to the Publisher System.
   (c) Asks the Cooperation Rater to decide whether to forward the packet or not.

The set of node components described above make a complete system for monitoring, rating, and publishing the reputation of any node. The output of each component is used by another to make efficient and useful routing in MANETs. Indeed, the proposed approach can enforce nodes to cooperate in a fully distributed manner. The extra overhead of maintaining the reputation rating of nodes up-to-date and recurrent is justified by the ability of the protocol to detect, stimulate, and enforce nodes to cooperate as shown in the simulation results later in this paper. In the next section, we present a detailed example that shows how the proposed approach works.

5.2 An Example

Let us assume that we have the network shown in Figure 4. Suppose that a source node A sends a packet to a destination node X and the first route hop is node B. The update procedure of the protocol needs to handle two cases, namely, either B will cooperate by forwarding the packet or misbehave by refusing to forward the packet, i.e., drops the packet. For each case, we show how the protocol works as follows:

– If B forwards the packet, the following will occur:
  1. All B's neighbors, i.e., nodes (A, C, D, E, F) increment the Total Help related to B by a value of one. This rule applies to all intermediate nodes.
  2. All neighbors that can hear A & B decrements the Total Help related to A based on the return result. If the destination X acknowledges the packet, the Total Help for A is decremented by the hop count of the route. If an intermediate node reports a route break or a misbehaving node in the downstream of the route, the Total Help for A will be decremented only by the hop count to reach that reporting node. This rule applies only to the source node.
  3. All neighbors that hear A only decrement the Total Help related to A based on the return result. If the destination X acknowledges the packet, the Total Help for A is decremented by the hop count; otherwise (the source send the previous packet) the Total Help decrement by total hop to reach the destination divided by 2. This rule applies only for the source node.
  4. A decrements its Count based on item 2 above.
– If B drops the packet, the following occurs:
  1. A or the sender and all neighbors that can hear both A and B (e.g,. C) will decrement the Total Help with respect to B by a value of two. This rule applies to all intermediate nodes.
  2. The Total Help of node A (the sender) with respect to its neighbors (C, N, M) doesn't change.

5.3 Protocol Optimizations

The protocol presented in this paper applies to all types of ad hoc networks. However, different types of ad hoc networks have different characteristics. In fact, we can make use of these differences as well as some key observations to make some optimizations on the proposed protocol to achieve better performance:

A. *Low mobility:* If the network has low mobility or no mobility at all such as the case of Wireless Sensor Networks, we can limit the reputation message to be forwarded only to a certain number of hops (e.g., a hop count limit of 3). This modification have the potential of reducing the network traffic because the message is dropped after a certain hop.

B. *Fast route recovery:* In the example of section 5.2, assume that node A sends a packet to node X and the route goes through nodes B, F and then X. Now, suppose that node B forwards the packet to node F but node F discards the packet. In the original protocol, node B informs the source node A about the link failure and then A can choose another route or initiate a rout discovery if needed. A more logical solution is to allow node B to recover by finding a new route locally or if it knows another route to the destination X, then that route can be used instead. In other words, instead of informing the source about the broken link, node B uses its own way to forward the message to the destination. This optimization can reduce the network traffic and can save more power.

C. *More fairness:* When an intermediate node discards a packet, all its neighbors will decrement its Total Help by a value of 2. A more fair approach is to decrement only from the Total Help of the misbehaving node, i.e., decrement from the misbehaving node the number of hops from the source to that selfish node. In the example above, suppose that node N sends a message to node X through the route N, V, C, D, H, X and node H discards the packet. In this case, all neighbors of node H will decrement the Total Help field of H by an amount of 4, which is exactly the hop count to reach node H. By doing this, node H will soon be declared uncooperative node and finds itself isolated.

6 Performance Evaluation

The proposed scheme was studied through simulations. All simulations were carried out using the NS 2.26 simulator [19]. A MANET in a fixed area of size (2000m×2000m) with a variable number of mobile nodes was simulated. Unless mentioned specifically, the number of mobile nodes in the simulations are 500. Mobile nodes were initially placed randomly within the fixed-size network area. The transmission distance is set to 250 meters. Links have a transmission rate that is equiprobably selected from a pre-defined set of transmission rates (1, 2, 5.5, and 11 Mbps) as in IEEE 802.11 standard. To simulate good and bad behavior, neighborhood, observation mistakes, movement, and trust updates, we used a grid of nodes, namely, VGA as the underlying structure. Nodes observe the behavior of their neighborhood. Depending on its position in the grid, a node has up to 8 neighbors. A node can only directly observe neighbors, i.e., node i at row j and column k, denoted as i_{jk}, can observe any neighboring node n in its row $n_{j,<k+1|k-1>}$ and in its column $n_{<j+1|j-1>,k}$ or diagonally one hop away $n_{<j+1|j-1>,<k+1|k-1>}$.

Connections are established between randomly selected pairs of source/destination nodes. For the communication pattern, we used bursty traffic with ON-OFF periods. The ON and OFF periods of each flow are exponentially distributed with mean values being simulation parameters. During ON periods, traffic is generated at a variable rate that is a simulation parameter. During OFF period, the source does not generate packets and remains idle. New session requests arrive according to a Poisson arrival process with mean arrival rate of 0.1. The length of each flow session is exponentially distributed with the mean value being a simulation parameter. We vary the session duration, the ON period length of each session, the packet generation rate during the ON period, and the mean OFF period length to obtain different levels of the offered load. Unless otherwise specified, the default offered traffic load was set to 0.8. The packet sizes are fixed at 512 bytes. Periodically, nodes move around. We emulate this with a realistic mobility model, called Probabilistic mobility model (PMM), that was implemented and used in the simulations. The details of this model can be found in [18].

In order to determine the performance of the proposed protocols, we focus on the following important performance metrics:

(a) *Packet Delivery Ratio (PDR):* This metric is evaluated based on the following basic formula:

$$TP = \frac{R}{G} \tag{6}$$

Where R and G are the number of packets received successfully and the number of packets generated, respectively. In fact, PDR is affected by packet loss and packet retransmissions. Packet loss can occur for many reasons. However,

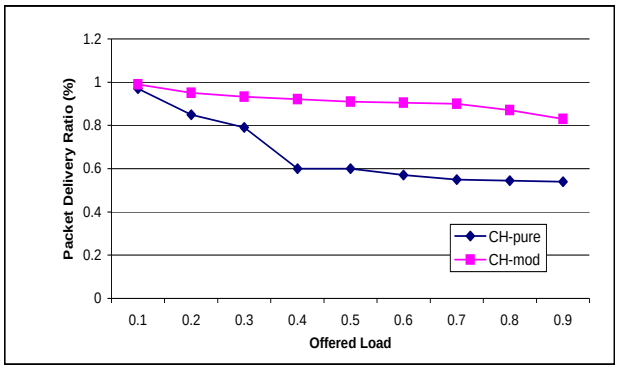

Fig. 5 Effect of selfish behavior on packet delivery ratio.

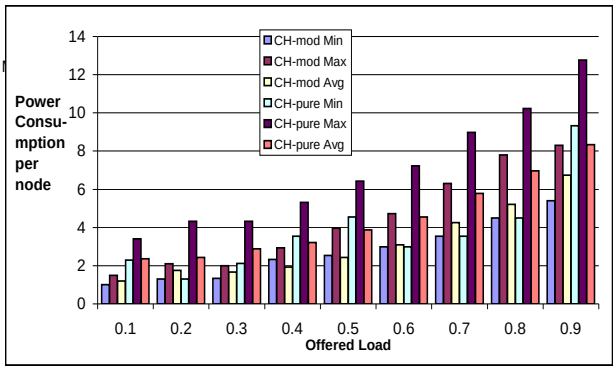

Fig. 6 Effect of selfish behavior on node power consumption.

we focus on the packets loss due to the situation when an intermediate node intentionally drops received packets instead of forwarding them. This is only a form of packet loss that is directly attributable to malicious behavior. Therefore, we use the number of intentionally dropped packets as a metric, both in absolute numbers and relative to the number of packets originated by source nodes.

(b) *Information Exchange:* By this we mean with whom should information be exchanged, i.e., with neighbors or remote nodes? And, what is the effect of mobility on this? A key question that is also addressed here is how long does it take until a misbehaved node is detected?

(d)*Fairness and Profit:* When nodes feel that their interest is respected, then they intend to cooperate. This would happen when the nodes feel that they receive fair treatment. In this regard, fairness means to protect well-behaved flows from aggressive and ill-behaved flows. Recall that the proposed protocol uses reputation information from the network, and allocates resources to flows such that fairness is achieved. The profit is calculated as the ratio between the total help that a node gave to its neighbors and the total help the node gained from the network. This ratio should be as close to 1 as possible, i.e., the amount of help a node received is nearly equivalent to the amount of help exerted by the node to others. This can stimulate nodes to always cooperate.

(c) *End-to-end Delay:* The end-to-end delay is defined as the time needed to send a packet successfully from the source node to the destination node. This is important because packet dropping can adversely affect the packet delivery ratio especially when delay guarantees are required. The highest, lowest and average end-to-end delay will be considered [1].

The set of simulation results is divided into two sets. The first set is obtained for the case when the set of nodes in a zone are uncooperative in the CH election algorithm. In the second set, we extend this to the whole network and study the effect of the hybrid approach with comparison of some existing schemes.

(i) Enforcing node cooperation in CH election: Our interest is to study the effect of user selfishness on the CH election process in terms of the per node packet delivery ratio and the per node average energy consumption in the network. We are also interested in studying the effect of varying the number of selfish users on the call acceptance rate in the network. We modified the CH election algorithm to take into consideration the effect of selfish behavior as per equations (1) and (2). The value of w_i^p of a node i was selected based on the fraction of available energy at node i, E_i, at the beginning of the election period, i.e., $w_i^p = E_i$. First, we measure the average energy consumption per node in the intra-zone cooperation. We fix the number of nodes to 350. Our CH election is the periodic strategy with fixed period set to 3 sec. Figure 5 shows the packet delivery ratio when the offered load changes when both the original CH election algorithm (CH-pure) and the CH election algorithm with the effect of selfish behavior is added (CH-mod). It is noted from the figure that the scheme is able to enhance the packet delivery ratio by enforcing nodes to cooperate for the benefit of all. We also measured the node power consumption variance for both cases as shown in Figure 6. The addition of the scheme shows that average node power consumption decreased, which helps prolong the network operational lifetime.

Then, we study the effect of users behavior on the blocking probability, that is the number of rejected calls, for the inter-zone cooperation. We assume that two types of users exist. The first type is cooperative users who apply the same strategy but with different values of willingness, while the second type is selfish users with low willingness values. In particular, we consider a group of cooperative users with willingness randomly generated between 0.8 and 1.0 and a group of selfish users with willingness equal to 0.5. This scenario may correspond to the case where users

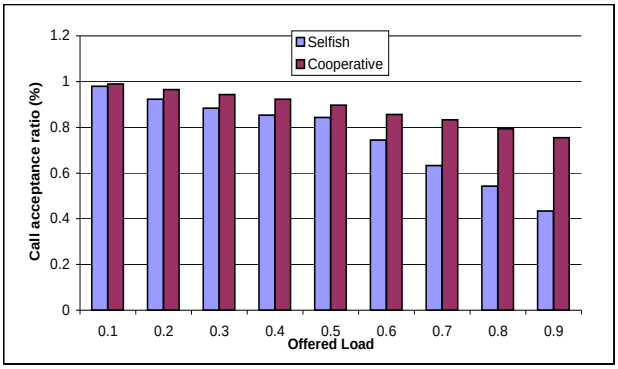

Fig. 7 Effect of selfish behavior on call acceptance ratios for variable offered load

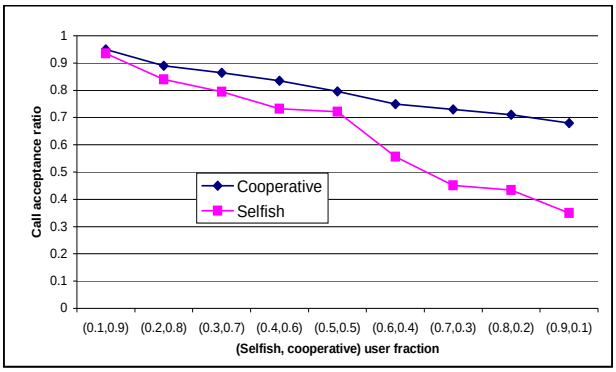

Fig. 8 Effect of selfish behavior on the call acceptance rate for various populations.

have different energy constraints or the case where some users are acting selfishly to maximize their own benefit. We fixed the total number of users to 100. We used the TC-based routing scheme. We measured the percentage of accepted calls for each user type when the offered load is varied. Figure 7 shows the call acceptance ratio for various values of offered load for both selfish and cooperative users when the ratio of selfish to cooperative users is (0.3,0.7). At low loads, the system is able to support both users requests. As the load increases, the system is able to differentiate between the two user types and favor the cooperative users.

We then varied the number of users in each type such that the user ratios range from (0.1,0.9) to (0.9,0.1) corresponding to the fraction of selfish and cooperative users in the network, respectively, and measured the call acceptance rate in the system for both the original routing scheme and the modified one. In Figure 8, the curves labeled by "cooperative" and "selfish" represent the performance of cooperative and selfish users, respectively under the original routing scheme and the modified routing with the inter-zone cooperations scheme. As the number of selfish users increases, the number of calls accepted for them decreases, while cooperative users managed to get over 70% of their calls through. This, reflects that the system is able to prevent selfish users from making advantage of cooperative users.

(ii) Enforcing node cooperation in the network:

Our first objective here is to be able to detect and isolate misbehaving nodes. Once isolated, these nodes will indirectly be enforced to cooperate. Mobile nodes exchanged reputation information with their neighbors periodically. The misbehaving nodes will reverse the reputation information before giving it to the neighbors. This process was iterated until all of the malicious nodes were classified as "detected" by all of the nodes in the network. The value of T_m was set to 0.75. As a rehabilitation mechanism, the nodes periodically review their reputation opinions and reverse their opinion from "Misbehaving" to "normal" when the reputation was substantially better than T_m. For this part, we also vary the number of misbehaving nodes and the threshold for detecting those nodes (default: 0.75).

Figure 9 shows the maximum detection time, i.e., the time in the simulation when the last node detected a particular malicious node versus how many of the malicious nodes were detected by all nodes at that time. This figure is a representative of the results obtained by the simulation since the type of a node is drawn from probability distribution and not explicitly specified, thus the portion of malicious nodes varies. As the number of observations made increases, i.e. using first hand as well as second hand and so on, the possibility of detecting malicious nodes approaches 1. This means that all malicious nodes are detected and isolated. Figure 10 shows the minimum, average, and maximum end-to-end delay of the system with and without the proposed enforcing mechanism. As the figure shows, the enforcing mechanism is able to reduce the end-to-end delay by forwarding more packets in less time. Finally, Figure 11 shows a comparison between the proposed scheme with two reputation-based schemes, namely, CONFIDANT and CORE in terms of the packet delivery ratio. Our scheme can outperform the other two schemes especially at higher loads on the network.

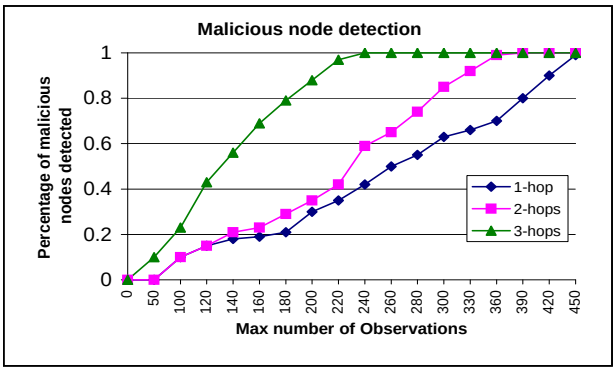

Fig. 9 Percentage of malicious nodes detected vs. maximum detection time.

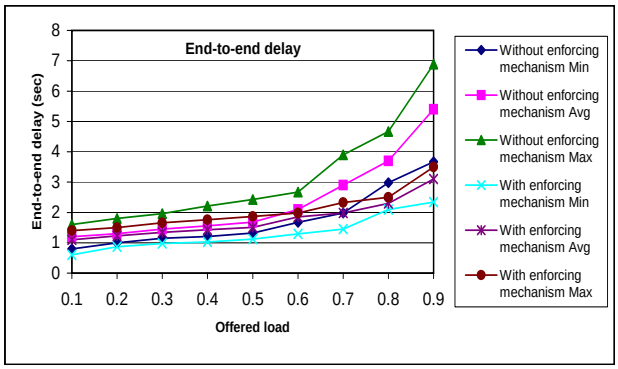

Fig. 10 Effect of selfish behavior on the call acceptance rate for various populations.

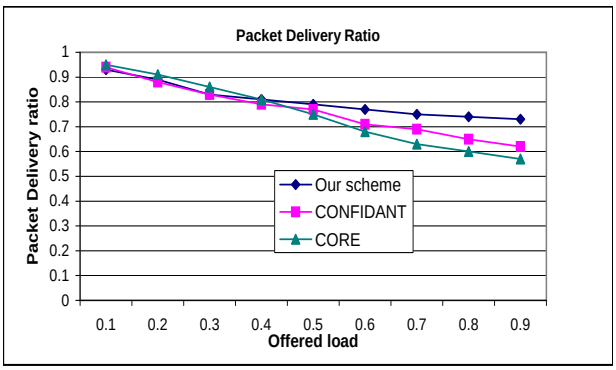

Fig. 11 Comparison between our scheme vs. other schemes.

7 Conclusions

Cooperation is very important in Mobile ad hoc networks because most of the network operations, e.g., packet forwarding rely totally on cooperation among nodes. In this paper, we presented a new model that achieves fairness and improves cooperation among nodes on the expense of a little extra overhead. In particular, we presented two reputation-based incentive mechanisms. The proposed mechanism is protected from abuse by its fully distributed nature and hence there is no need implement it in a tamper resistant hardware module. To design a protocol that achieves cooperation between nodes we must take into account important issues such fairness. It was shown in this paper that by combining reputation-based with virtual currency based schemes, better performance can be achieved in MANETs. For future work, we will try to optimize some of the critical parameters such as the misbehave threshold T_m and the super node threshold T_s that maximize other performance metrics.

References

1. X. Bangnan, S. Hischke, B. Walke, "The role of ad hoc networking in future wireless communications", International Conference on Communication Technology (ICCT 2003), **Vol: 2**, Pages: 1353-1358.
2. J. N. Al-Karaki, A. E. Kamal, "End-to-End Support for Statistical Quality of Service in Heterogeneous Mobile Ad hoc Networks", Computer Communications, **Vol. 28**, No. 18, pp. 2119–2132.
3. J.N. Al-Karaki, A.E. Kamal, "On the Optimal Clustering in Mobile Ad hoc Networks", Proceedings of IEEE Consumer Communications and Networking, Las Vegas, Nevada USA / January 5-8, 2004.
4. J. Hu, "Cooperation in Mobile Ad Hoc Networks", Technical report, Computer Science Department, Florida State University, January 11, 2005.
5. L. Buttyan, J.-P. Hubaux, "Nuglets: a Virtual Currency to Stimulate Cooperation in Self Organized Mobile Ad Hoc Networks"', Technical report No. DSC/2001, Swiss Federal Institute of Technology, Lausanne, August 2001.
6. S. Buchegger and J.-Y. Le Boudec,"Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes: Fairness In Dynamic Ad-hoc Networks", In Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), Lausanne, CH, June, 2002.

7. P. Michiardi, R. Molva, “Core: A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks”’, IFIP-Communicatin and Multimedia Securtiy Conference, 2002.
8. S. Zhong, J. Chen, and Y. Richard Yang, “Sprite: A simple, Cheatproof, Credit-based System for Mobile Ad hoc Networks”, in Proceedings of IEEE INFOCOM ’03, San Francisco, CA, April 2003.
9. S. Bansal and M. Baker, ”Observation-based Cooperation Enforcement in Ad Hoc Networks”, http://arxiv.org/pdf/cs.NI/0307012, July 2003.
10. M. Conti, E. Gregori, and G. Maselli,“Towards Reliable Forwarding for Ad Hoc Networks”’, Technical report, IIT Institute-CNR, Italy, 2003.
11. D. Senn,“Reputation and Trust Management in Ad Hoc Networks with Misbehaving Nodes”’, Diploma Thesis, July 2003.
12. L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J. P. Hubaux, J. Y. Le Boudec, “Self-Organization in Mobile Ad-Hoc Networks: the Approach of Terminodes”, IEEE Communications Magazine, **Vol. 39**, No. 6, June 2001.
13. S. Marti, T. J. Giuli, K. Lai, M. Baker, “Mitigating routing misbehavior in mobile ad hoc networks”, proceedings of MOBICOM 00 pages 255-265, August 2000.
14. N. Ben Salem, L. Buttyan, J. P. Hubaux, and M. Jakobsson,“A charging and rewarding scheme for packet forwarding in multihop cellular networks”, proceedings of MobiHoc’03, 2003.
15. L. Buttyan, J. P. Hubaux, “Stimulating cooperation in self organizing mobile ad hoc networks”, in ACM/Kluwer Mobile Networks and Applications (MONET), **Vol. 8**, No. 5, Oct. 2003.
16. L. Anderegg, S. Eidenbenz, “Ad hoc-VCG: A truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents”, International Conference on Mobile Computing and Networking, Pages: 245-259, 2003.
17. L. Buttyan, J. P. Hubaux, “Enforcing Service Availability in Mobile Ad-Hoc WANs”, Proceedings of the IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC), Boston, MA, USA, August 2000
18. Jamal N. Al-Karaki, ”Infrastructureless wireless networks: Cluster-based architectures and protocols”, http://archives.ece.iastate.edu/archive/00000062/, Ph.D. thesis, Iowa State University, 2004.
19. “The ns-2 simulator”, http://www.isi.edu/nsnam/ns.