ARM: An Account-based Hierarchical Reputation Management System for Wireless Ad Hoc Networks

Haiying Shen and Ze Li
Department of Computer Science and Computer Engineering
University of Arkansas, Fayetteville, AR 72701
{hshen, zxl008}@uark.edu

Abstract

Mobile ad hoc networking works properly only if mobile nodes cooperate in routing. However, since wireless mobile nodes are usually constrained by limited power and computation resources, some selfish nodes may refuse to forward packets which are not of their direct interest. Most previous efforts counter this behavior by having each node keep a reputation table and exchanging the information with each other, and refusing to forward the packets of selfish nodes. Maintaining and exchanging information among individual mobile nodes in a dynamic environment consumes significant resources, and such a punishment method is not effective enough. This paper presents a novel Account-based hierarchical Reputation Management system (ARM) to avoid selfish nodes and encourage node cooperation. In order to save the resource consumption of information exchange, ARM builds a hierarchical structure with low mobility nodes in the high level. These nodes constitute a locality-aware DHT for efficient reputation value collection and exchange. Furthermore, ARM provides a novel account management model to encourage node cooperation in the network. The account management model intelligently integrates global reputation management reputation system and pricing-based model for effective selfish node punishment. Theoretical analysis and simulation results show that ARM can greatly improve the performance of a defenseless network by effectively deterring selfish nodes and encouraging node cooperation at a significantly low resource consumption.

Keywords: Wireless ad hoc network, Reputation systems, Pricing-based model, Peer-to-peer, Distributed hash table.

I. Introduction

A mobile ad hoc network (MANET) uses a self-configuring network that is formed automatically by a collection of mobile nodes without the help of a fixed infrastructure or centralized management. In an ad hoc network, the transmission range of a mobile node is limited due to the power constraint. Hence, communication between two nodes beyond the transmission range relies on intermediate nodes to forward the packets. However, since each of these devices belongs to different authorities or individuals, and these mobile nodes are typically constrained by power and computing resources, a selfish node may not be willing to forward packets that are not directly beneficial to it. The presence of only a few misbehaving nodes can dramatically degrade the performance of an entire system.[1]

Numerous methods have been proposed to cope with this cooperation problem. They can be divided into two main categories: reputation-based schemes and pricing-based schemes. Reputation-based schemes [1–5] let each node hold a reputation table recording the reputation of other nodes, and exchanges information with neighbor nodes. A node selects routing path according to node's reputation value. However, the existing reputation-based schemes suffer from lack of effective mechanisms to measure and propagate reputation information, which consume storage resources, enormous computing resources, and limit the efficiency of the propagation of the reputation values especially in a highly dramatic network. Moreover, most methods set up a reputation threshold. Nodes whose reputation value are higher than the threshold are regarded as unselfish nodes, while nodes whose reputation value are lower than the threshold are regarded as selfish nodes. Nodes provide services to high-reputed nodes, and refuse to provide services to low-reputed nodes. Therefore, as long as a node has a RV a little higher than the threshold, it can always be served. This is not fair to high-reputed nodes with different reputation levels. Reputation-based schemes need to have a complement method to help them wisely punish selfish nodes, and reward altruistic nodes.

Pricing-based model [6–9] treat packet forwarding as a service that can be priced, and introduce some form of virtual currency to regulate packet forwarding relationships among different nodes. However, traditional methods that include "virtual currency" in the transmitting packets requires a fair amount of computation and storage resources. In addition, they fail to provide a way to know the service quality of a node. Moreover, the implementation of "virtual currency" and "virtual bank" make them more complex with high requirements on overhead, security and topology.

In this paper, we address the problem of selfish nodes by an Account-based Reputation Management system (ARM). The main purpose of this system is to detect and eliminate selfish behaviors and encourage node cooperation in an effective and efficient way without increasing packet complexity. The novel features of the ARM scheme include:

(1) Unlike the traditional reputation method that let individual node keep reputation table, ARM selects low mobility nodes as reputation management nodes (RMN), and builds a hierarchical structure with RMNs in the high level and normal wireless nodes in the low level. Letting RMNs be responsible for managing RVs saves computing resources for information exchange in a dynamic network. In addition, it releases the reputation management load from the mobile nodes, so that they can have more resources for the data transmission.

(2) Distributed Hash Table (DHT) is characterized by reliability, scalability and efficiency. ARM constitutes RMN into a locality-aware DHT structure for efficient reputation information collection and exchange.

(3) Unlike the traditional reputation-based system, ARM implements an pricing-based model, i.e. account management model incorporating reputation system. In ARM all services are priced based on node reputation. Such method can prevent the selfish nodes keeping their RV at a low level that just above reputation threshold to avoid punishment.

(4) Unlike traditional pricing-based module, ARM does not require "currency" circulated in the system, which destroy traditional IP packet structure. Furthermore, ARM can effectively encourage the selfish behavior without increasing system complexity.

The remainder of this paper is organized as follows. Section II provides related works for encouraging nodes cooperation in MANET. In section III, we specify assumptions made in this paper and introduce the ARM system. section IV presents simulation results to demonstrate the effectiveness of the ARM scheme. Section V concludes the paper.

II. Related Work

The approaches for fostering node cooperation can be classified into two categories. One category of approaches builds a reputation system to get a RV for each node's trustworthiness based on the evaluation from others about its performance [10, 2–4]. Marti [10] proposed two techniques, *watchdog* and *pathrater*. The *watchdog* in a node promiscuously listens to the transmission of the next node in the path for detecting misbehavior. The *pathrater* in a node keeps the rating of other nodes to avoid uncooperative nodes. Core [2] uses the *watchdog* technique and weighs heavily towards past reputation to avoid mistaking cooperative nodes with low battery condition as misbehaving nodes. CONFIDANT [3] detects misbehavior nodes and sends alarm messages to other nodes to isolate misbehaving nodes. Wu and Khosla [4] defined the first-hand reputation as the ratio of the number of packets to be forwarded to the number of packets that have been forwarded, and proposed to update RVs when necessary.

Another category of approaches is pricing-based model that provides incentives by using credit, virtual currency or micro payment [6–9]. Buttyan and Hubaux [7] proposed two payment models: packet purse model and packet trade model. In the former, a source node pays relay nodes by storing virtual cashes in the packet-head. In the latter, a relay node buys packets from the previous node and sell them to the next node in the path for more virtual cashes. The credit-based system in [9] uses credit clearance service and message receipts. When a node receives a message, the node keeps a receipt of the message and uploads it to the credit clearance service for credits.

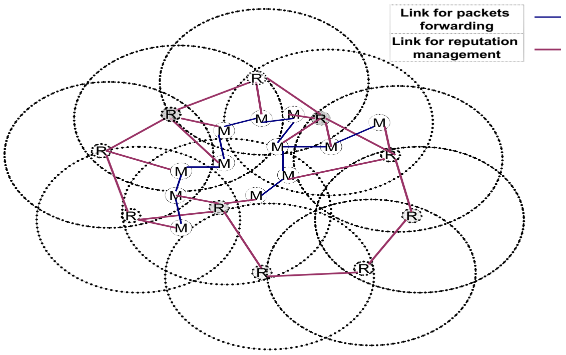


Fig. 1. The ARM hierarchical structure

ARM builds a hierarchical structure to efficiently manage the RVs of all nodes, and release the reputation management load from individual high mobility nodes. This enables low-overhead and fast global reputation information accesses. Furthermore, ARM intelligently integrates reputation system into pricing-based model to avoid selfish nodes. Rather than using threshold to detect selfish nodes, which treats equally to the reputed nodes with different RVs, ARM takes reputation into account when determining service prices, which effectively distinguishes reputed nodes in different levels. In addition, supported by the account management model, ARM does not require the "currency" transmitting with packets.

III. ARM: Account-based hierarchical Reputation Management System

A. Assumptions

In this paper, we make the following assumptions.

(1) Selfish nodes in the ad hoc network under our consideration are interest-driven. That is, the behaviors of these nodes in the system are only for the best benefit.

(2) We don't consider malicious and conspiracy nodes.

(3) Every node in the system has dual mode interface, e.g. the IEEE 802.11 and cellular interface.

(4) The RMNs are static or low mobility nodes. They will not serve as relay nodes for packet forwarding.

(5) A wireless interface of each node supports promiscuous mode operation: a node always listens to every transmission within its one-hop neighborhood although it doesn't involve in the transmissions.

(6) An antenna used on each node is omni-directional which enables its transmissions to be monitored by its one-hop neighbors.

B. Overview of ARM

Figure 1 shows the hierarchical structure of ARM where M represents wireless mobile node, and R denotes RMN which forms the Global Reputation Management System (GRMS) to manage the account value (AV) and RV of mobile nodes. These RMNs constitute a locality-aware DHT structure for efficient operation. GRMS consists of two functions: reputation management and account management. Reputation management function is used to manage the RV of the individual mobile node. Each mobile node has a watchdog [10] to calculate the RV of its neighbor nodes and report RVs to GRMS periodically at every time interval of T. These RVs are merged in the GRMS to determine a new RV of individual node. The account management function is used to encourage the cooperation of all mobile nodes in the system. The AV of node N will be deducted by a certain value by GRMS according to the number of packets generated by the node, while the AV will be increased according to the number of packets forwarded by the node. If node N has higher reputation, it need pay lower price to GRMS. Equipped with dual mode interfaces, mobile nodes in the systems can either use low transmission power interface (i.e. IEEE 802.11) for the packet forwarding or use high power interface (i.e. cellular radio) for the reputation data inquiring or updating from the RMNs.

ARM specifies that in a packet transmission, only the source node needs to pay for the transmission to the next hop node. Forwarding nodes do not need to be charged for the transmission, and will be awarded for the packet forwarding effort.

The accounts are managed by GRMS, and there is no actual "currency" circulated in the networks. The updated RV reported by misbehavior node (RV below the threshold RV) or nodes whose AV lower than zero, will be ignored by GRMS.

More specifically, when node $N1$ receives a request from $N2$ to forward a certain number of packets, $N1$ will contact a close RMN (e.g. R1) to request the RV and AV of $N2$ using high power interface. If $R1$ does not has the information of $N2$, $R1$ can inquire other RMNs according to the DHT routing algorithm. If the RV of $N2$ below a threshold value Th or the AV of $N2$ is less than zero, $N1$ refuse to forward the packets from $N2$. If the RV of $N2$ is higher than Th, $N1$ charge the forwarding from $N2$ based on its RV. All nodes in the system merge their new collected RV of other node in the GRMS at every interval T.

C. Reputation Management System

In the reputation management system, two operations occur frequently: reputation update and reputation information query.

1) Neighbor Monitoring and Reputation Update: In ARM, neighbor monitoring is used to collect information about the packet-forwarding behavior of the neighbors. With the promiscuous mode, a node is capable of overhearing the transmissions of its neighbors. A node maintains a current neighbor node list (NNL), which contains all of its current neighbor node IDs and their information including RV and AV. Therefore, it does not need to query GRMS all the time. The NNL uses a Time To Live (TTL) function to keep the freshness of the information of neighbor nodes. When a neighbor node leaves the transmission range of the node, the information of that neighbor will be deleted from NNL. Each node will periodically reports the collected RVs at every time interval T to GRMS. If the topology of the wireless ad hoc network changes dramatically, the T can be assigned a small value, while if the changes is slow, T can be assigned to a large value.

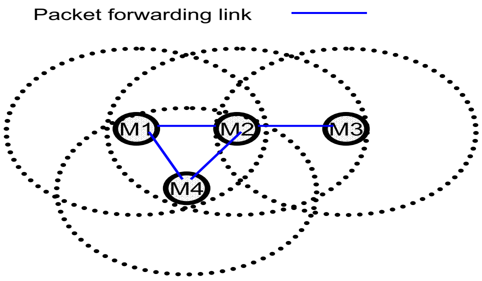


Fig. 2. The watchdog function

In addition, node uses a watchdog mechanism [10] to monitor the packet-forwarding behavior for each of its neighbors by keeping track of two counters. One is for counting total number of packets that node i has transmitted to j, called"Request-for-Forwarding", denoted by $RF_i(j)$. Another is for counting the total number of packets that have been forwarded by j and noticed by i, called "Has-Forwarded" denoted by $HF_i(j)$. Figure 2 shows how watchdog works. In the figure, when $M1$ transmits packets to $M2$, $M4$ also receive packets. When $M2$ forwards packets to $M3$, $M1$ and $M4$ also overhear the transmissions. Therefore, whenever node $M2$ receives a packet which should be forwarded, either from node $M1$ or from node $M4$, $M4$ and $M1$ will overhear the transmission and store the packet in their buffer, set a timeout and overhear the transmission of $M2$, and increase the RF counter by one. If node $M2$ forwards the packet, the packet is removed from the buffer and the HF counter is increased by one. Otherwise, the packet is removed when time reaches the specified timeout.

These two counters for each neighbor are counted over T. Periodically, the monitoring module creates a value called Local Value (LV) based on this two counters for all overhearing neighbor nodes and reports LV to the most close RMN at each time $t_0 + nT$, where t_0 is the time when mobile node join the system, and $n \in (1, 2, 3...)$. $LV_i(j) = \frac{HF_i(j)}{RF_i(j)}$. The value of $RF_i(j)$ and $HF_i(j)$ will also be reported to the GRMS for the account value calculation. After updating, these counter will be reset to zero.

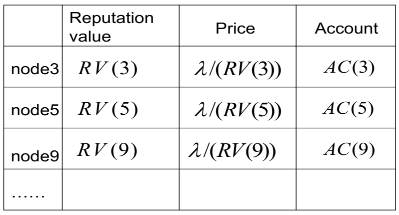
2) Reputation Management Structure: We leverage DHT network architecture [11] for the infrastructure of GRMS for scalable communication and the DHT structure of GRMS is formed by a number of low mobility nodes. Before presenting the details of the reputation management structure, we introduce the DHT network at first.

DHT network is a class of decentralized systems that partition ownership of a set of objects among participating nodes, and it achieves a time complexity of $O(\log n)$ per lookup request by using $O(\log n)$ neighbors per node, where n is the number of nodes. Each object or node is assigned an ID (or key) that is the hashed value of the object or node IP address using consistent hash function. An object is stored in a node whose ID equals to or immediately succeeds to the object's ID. The overlay network provides two main functions: `Insert(key,object)` and `Lookup(key)` to store an object to a node responsible for the key, and to retrieve the object. The message for the two functions is forwarded based on the DHT routing algorithm. Each node maintains a routing table recording its neighbors in the overlay network.

We propose to construct locality-aware DHT-based infrastructure network where logical proximity abstraction derived from the RMNs match the physical proximity information in reality. Thus, a RMN always physically close RMN, leading to high efficiency. We use a landmarking method to represent node closeness by closeness by indices [12]. Each RMN measures its reputation management node measures its physical distances to the distances $< d_1, d_2, \ldots, d_m >$ as its coordinate in Cartesian space. Two physically close RMNs will have similar landmark physically close RMNs will have similar landmark vectors. We use Hilbert curve [12], to map m-dimensional landmark vectors to real numbers, such that the closeness relationship among the points is preserved. We call this number the *Hilbert number* of a RMN. physical closeness of base stations on the network.

To build a locality-aware DHT-based reputation management network for the GRMS, we directly use a reputation management node's Hilbert number as its DHT ID, and assign an ID to the reputation information of a node by hashing the node's IP address using consistent hash function. As a result, GRMS constitute a locality-aware DHT structure where physically close neighbors are neighbors due to the feature of Hilbert number. Based on DHT key assignment policy, each node's RV will be stored in its owner. Therefore, if node i wants to query for node j's RV, it asks its closed RMNs. If these these nodes do not have the reputation information, node i sends `Lookup(key)` request with the hashed value of node j's IP address as the key. The request will be forwarded to the node that has the reputation information of node j using DHT routing algorithm. In the case that a source node is not in the range of a RMN, it can either use recently queried information. Thus, a node can always access the reputation of another node efficiently. another node efficiently.

In ARM, each mobile node keeps a list of its current neighbor nodes to facilitate the frequently link establishment. If one of node N's neighbors N_i requests node N to forward a packet, node N will query the RV and AV of N_i in the close $R1$, if there is no information about N_i in its NNL. If $R1$ does not have the RV that node N required, $R1$ will inquire the RV from other $RMNs$ based on the DHT query algorithm. After getting the RV of N_i, node N will keep it, until N_i move out of the range of N.



	Reputation value	Price	Account
node3	$RV(3)$	$\lambda/(RV(3))$	$AC(3)$
node5	$RV(5)$	$\lambda/(RV(5))$	$AC(5)$
node9	$RV(9)$	$\lambda/(RV(9))$	$AC(9)$
......			

Fig. 3. Reputation table structure in the reputation management node

3) Reputation Management: Recalled that each mobile node reports its observed LV to GRMS periodically. However, during each time period T, many nodes observe the behavior of a certain node and report them to the GRMS. Therefore, we should give more weight to the higher reputed node to calculate the current RV about node N. The formula for calculation is:

$$RV_{Current}(N) = \frac{\sum_{i \in N \cup (LV_i(N) > Th)} LV_i(N) \cdot RV_{table}(i)}{\sum_{i=1}^{n} LV_i(N)}$$

where Th denotes the reputation threshold, below which the node will be regarded as misbehavior node, and RV_{table} denotes the past RV. When emerging those RVs, the GRMS ignores the RV reported by the selfish node. This is to prevent them from accusing other nodes of refusing to forward their packets.

However, note that in the wireless networks, packets are sometimes lost because of the noise interfere. Therefore, it is unfair to the node being regarded as doing selfish behaviors in this situation. Therefore, the old RV recorded in the reputation table should also be taken into account when calculating the new RV, that is

$$RV_{new}(N) = \alpha RV_{table}(N) + (1 - \alpha) RV_{Current}(N).$$

Therefore, according to different environments, we can adjust the weight on past or current behavior by changing the value of α to build the new RV.

Proposition 3.1: In ARM, the effect of the individual mistaking reputation report is small on global reputation evaluation, and the global RVs in GRMS can accurately reflect nodes' behavior.

Proof: Unlike the traditional reputation management scheme, ARM collects all the first hand observed RV together into the GRMS periodically. According to the statistical theory, the effect of extreme events is negligible if a large samples are provided. The weak law of large numbers shows, $lim_{n \to \infty} P(|X_n - \mu| < \varepsilon) = 1$; that is, given a sample of independent and identically distributed random variables with a finite expected value, the average of these observations will eventually approach and stay close to the expected value. ■

D. Account Management Function

ARM uses an account management function to avoid equal treatment to high-reputed nodes in different levels, providing incentive of cooperation between nodes, and sequentially deter selfish behaviors.

In the account management function, the pricing policy is generally based on RV in the reputation table:

$$P(N) = \frac{\gamma}{RV_{table}(N)} \ (1)$$

where γ is a constant weigh value and P is the transmission price per packet. The higher RV a node has, the lower price it need to pay for the transmission. Moreover, the GRMS also maintains a virtual cash account for each node in the system. Figure 3 shows the structure of a reputation table in GRMS. The GRMS initially assigns each new joining node a certain amount of virtual cashes in the account: Sum. Every time when a node N generates some packets in the system to other nodes, its account value will be deducted a certain amount of virtual cashes by $P_i(N) \cdot RFS_i(N)$, where $RFS_i(N)$ denotes the number of packets that source node N sends to its neighbor node in T. On the other hand, if node N helps others forward packets, node N's AV will be increased by $\lambda \cdot HF_i$ where λ is a constant reward for per packets node N forwarded and HF_i denotes the number of packets the nodes has forwarded in the time period T. Therefore, the total AV of node N is calculated as

$$AV = Sum - \sum_{i=t_0}^{t_0+mT} (P_i(N) \cdot RF_i(N) + \lambda \cdot HF_i). \ (2)$$

The forwarding node don't need to pay for the packets forwarding to the next hop.

AV of each node is allowed to be negative. However, only nodes with positive AVs are allowed to send packets.

Proposition 3.2: In ARM, a higher cooperative node will have large AV whereas a low cooperative node may lead to a negative AV.

Proof: Recalled in Section III-C3 that a higher cooperative node has higher RV in the GRMS. Moreover, formula (1) and (2) show a higher RV in the system leads to a low transmission price. Collaboratively forwarding packets can help node earn AV into the account. Therefore, highly cooperative nodes get higher AVs. Moreover, nodes can not get any benefit by maintaining RVs only above a threshold value. Formula (1)(2) show that a low RV results in a high transmission price leading to a fast AV decreasing. With a negative AV, a node cannot send any packet out. It takes a long time and consumes more resources to make the AV turn back to a positive value by forwarding packets. ■

As the proposition 3.2 indicates, to be cooperative will bring more benefit to selfish node. Therefore they will always be cooperative with each other.

IV. Performance Evaluations

This section demonstrates the distinguishing properties of ARM through simulation built on NS-2 [13]. we integrated ARM as an extension to the Dynamic Source Routing protocol (DSR). The results are compared against the regular DSR network without ARM. The simulated network consist of 50 wireless nodes randomly deployed in a field of 1000×1000 square meters. We use the Distributed Coordination Function (DCF) of IEEE 802.11 as the medium access control layer protocol. The radio transmission range for each node is 250 meter, and the raw physical link bandwidth is 2Mbits/s. The physical layer model is the two-ray propagation model. The height of both the transmit antenna and the receive antenna is 1.5 meter. The constant bit rate (CBR) is selected as our traffic mode with a rate of 2 packets per second.

The random way-point mobility model is used to generate the moving direction, the speed and the pause duration of each node. Each node is first randomly placed in the field, waiting for a pause time randomly chosen from $1 - 5$s, then moves to another random position with a speed chosen between 1 to 10m/s. 10 new source and destination nodes are randomly chosen every 40s. The results presented here are the average from the 10 simulations. Each simulation lasts $200s$. we set the reputation threshold as 0.4. The RV of each nodes is range from $0 - 1$, and we assume that the possibility of each node to forward the packets is equal to the RV.

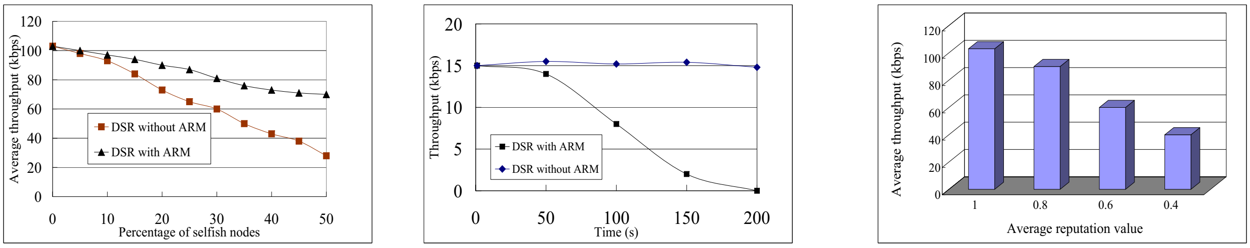
Figure 4 (a) plots the average throughput in the system versus the percentage of selfish nodes. The figure shows that the throughput of both scenarios decreases as the number of selfish nodes increases. This is because selfish nodes drop most of received packets. DSR with ARM can detect and avoid selfish nodes in routing, therefore its throughput is much higher. We also observer that the throughput of DSR with ARM also decrease as the number of selfish nodes increases. It is because the optimal routing without selfish transmission has longer path length, which incurs more transmission interference and leads to decreasing throughput.

In order to verify the effectiveness of ARM on punishing selfish nodes by refusing their packet transmissions, we tested the throughput of packets initialed by selfish nodes during a time interval. We assumed that there were 10 selfish nodes in the system. Figure 4 (b) show the experiment results. The figure shows that without ARM, selfish nodes keep constant throughput at about 15 kpbs. The throughput of ARM deceases sharply as time goes on. With the time passing by, ARM detects selfish nodes and let other nodes refuse to forward their packets leading to deceasing throughput of selfish nodes.

Figure 4 (c) shows the average throughput of the system versus the average RV of all nodes. Because a selfish node with a lower RV has a higher possibility of packet dropping, the average system throughput decreases as the average RV decreases. This result implies that it is important to keep a high RV of each node and to deter the behavior of keeping RV a little above reputation threshold. ARM is effective in encouraging not only the cooperation behavior of nodes but also achieving higher RV.

V. Conclusions

In this paper, we propose an Account-based hierarchical Reputation Management system (ARM) for detecting and eliminating selfish nodes in mobile ad hoc networks. Unlike traditional reputation management models and pricing-based



(a) Average system throughput (b) Throughput initiated by selfish nodes (c) Reputation value versus throughput

Fig. 4. Evaluation of the performance of ARM

models, in which the reputation value is maintained in each node, ARM builds a Global Reputation Management System (GRMS) including a reputation management function and account management function. GRMS consists of low mobility nodes which forms a locality-aware DHT structure for efficient reputation value management. Therefore, ARM can reduce the storage and computing burden of the mobile nodes for reputation management and increase the scalability of the reputation distribution. On the other hand, since the selfish node can manipulate their reputation to avoid the punishment, based on the account management function, such selfish behaviors can be deterred with low overhead.

Since some specialized RMNs need to be deployed to form a GRMS, it is not applicable in some situation such as space exploration network. Therefore, it will be very interesting to investigate the implementation of ARM on cluster based wireless mobile ad hoc networks.

Acknowledgments: This research was supported in part by U.S. Acxiom Corporation.

References

[1] S. Buchegger and J. Y. L. Boudec. Mitigating routing misbehavior in mobile ad hoc networks. In *Proc. of MobiCom*, 2000.

[2] P. Michiardi and R. Molva. Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In *Proc. of CMS*, 2002.

[3] S. Buchegger and J.-Y. L. Boudec. Performance analysis of the confidant protocol: Cooperation of nodes - fairness in dynamic ad-hoc networks. In *Proc. of Mobihoc*, 2003.

[4] Q. He, D. Wu, and P. khosla. Sori: A secure and objective reputation-based incentive scheme for ad-hoc networks. In *Proc. of WCNC*, 2004.

[5] T. Anantvalee and J. Wu. Reputation-based system for encouraging the cooperation of nodes in mobile ad hoc networks. In *Proc. of ICC*, 2007.

[6] M. Jakobsson, J. Hubaux, and L. Buttyan. A micro-payment scheme encouraging collaboration in multi-hop cellular networks. In *Proc. of Financial*, 2003.

[7] L. Buttyan and J. Hubaux. Enforcing service availability in mobile ad-hoc wans. In *Proc. of MobiHoc*, 2000.

[8] L. Buttyan and J. P. Hubaux. Stimulating cooperation in self-organzing mobile ad hoc network. *ACM Journal for MONET*, 2002.

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

[10] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proc. of MobiCom*, 2000.

[11] I. Stoica, R. Morris, and et al. Chord: A scalable peer-to-peer lookup protocol for Internet applications. *TON*, 1(1):17–32, 2003.

[12] Z. Xu, M. Mahalingam, and M. Karlsson. Turning heterogeneity into an advantage in overlay routing. In *Proc. of INFOCOM*, 2003.

[13] The network simulator ns-2. http://www.isi.edu/nsnam/ns/.