A Reputation based scheme to enforce Cooperation in Wireless Ad-Hoc Networks

Carryoles Maria #[1], Mani Prashanth Varma Manthena #[2], Liang Huo #[3]

# *Electrical Engineering Department, Delft University of Technology*
Mekelweg 4 2628 CD Delft
[1] C.R.A.Maria@student.tudelft.nl
[2] M.P.V.Manthena@student.tudelft.nl [3] L.Huo@student.tudelft.nl

***Abstract*— In this paper we describe an enforcing technique that improves the throughput in a wireless adhoc network inhabited by nodes that agree to relay packets but fail to do so. To diminish this issue, we propose a scheme that computes the reputation of nodes participating in the wireless adhoc network. Sometimes indirect or non-neighbor nodes may forward untrustworthy reputation information, we also propose a solution to mitigate this problem. Through our simulation, selfish nodes are effectively isolated, thus better performance of network is achieved.**

I. Introduction

A wireless ad-hoc network is a self-organized multi-hop type of wireless network. In [1], these wireless networks are described as ad hoc because they do not rely on any fixed infrastructure. Each node cooperates in routing by forwarding data from and to other nodes in the network. The nodes in an adhoc network are generally portable devices with constraint resources, such as power, computation ability and storage capacity. In an ad hoc network all devices have equal status on a network and are free to associate with any other ad hoc network device in link range. Very often, ad hoc network refers to a mode of operation of IEEE 802.11 wireless networks. An ad-hoc network is made up of multiple nodes connected by links. Links are influenced by the node's resources (e.g. transmitter power, computing power and memory) and by behavioral properties (e.g. reliability), as well as by link properties (e.g. length-of-link and signal loss, interference and noise). Since links can be connected or disconnected at any time, a functioning network must be able to cope with this dynamic restructuring, preferably in a way that is timely, efficient, reliable, robust and scalable. The network must allow any two nodes to communicate, by relaying the information via a path through other nodes.

Since there is no fixed infrastructure nor centralized infrastructure available, end-to-end communication sometimes may require routing via several intermediate nodes.

Routing protocols proposed, such as Dynamic Source Routing (DSR)[10] and Adhoc On-demand Distance Vector routing (AODV)[11], for wireless adhoc networks are based on the assumption that nodes joining and leaving a wireless adhoc network will cooperate. Cooperative behavior therefor cannot be taken for granted. In the algorithm we enforce cooperation between nodes in the network by isolating them either temporarily or permanently depending on repeated selfish behavior and thus inspiring good behavior. The algorithm takes liars into account and avoid their faulty information. Reputation agreement and propagation between the nodes are also taken into consideration. The rest of this paper is organized in the following way. In section II we review related work adressing the problem of non-cooperative nodes. Section III discusses our problem statement, assumptions and contraints. Section IV presents our algorithm, followed by a detailed analysis on the implementation. Lastly in section V we conclude our work and discuss future work in section VI.

II. Related Work

Schemes that stimulate cooperation and mitigate the harmful effect of non-cooperative in mobile ad-hoc networks can be classified into two basic schemes. i) Virtual currency based and ii) reputation based. In [4], virtual currency schemes are defined as schemes that use incentives to enforce nodes cooperation. Nodes get incentives when cooperating in packet forwarding. If a node doesnt have any incentives, packets will not be forwarded to nor will packets from these nodes be forwarded.

A. *Virtual Currency Schemes*

Since forwarding a message will incur a cost (of energy and other resources) to a node, an uncooperative node will need an incentive in order to forward messages of other nodes. Virtual currency systems [4, 12, 13, 14, 15, 16] use credit or micro payments to compensate for the service of a node. A node receives a virtual payment for forwarding the message of another node, and this payment is deducted from the sender (or the destination node). Two examples of such systems are: Nuglets [4, 12, 13, 14, 15,] and Sprite [4, 16].

1) Nuglets: Buttyan and Hubaux have introduced a virtual currency, called nuglets, and present a mechanism of charging/rewarding service usage/provision to stimulate cooperation in mobile ad hoc network [12, 13, 14, 15,]. Models presented for using the nuglets are: i) packet purse model: and ii)packet trade mode. In the packet purse model, the source of the packet is charged. When sending the packet, the source loads it with a number of nuglets sufficient to reach the destination. Intermediate nodes take some nuglets for the forwarding service.

In the packet trade model, the destination is charged. Packets are traded for nuglets by intermediate nodes. Here intermediary nodes buy" the packet from the previous node for some nuglets and sell" it to the next node for more nuglets. Therefore, every intermediate node gains nuglets for forwarding and the total cost of forwarding the packet is paid by the destination node.

To implement either the packet purse model or the packet trade model, tamper-proof hardware is required at each node to prevent the node from illegitimately increasing its own nuglets and to ensure that the correct amount of nuglets is deducted or credited at each node [4].

2) Sprite: S. Zhong et al. proposed Sprite [16], a cheat-proof, credit-based system for mobile ad hoc networks. Sprite uses credit to provide incentives for mobile nodes to cooperate and report honestly. Their basic idea of is as follows:

A Credit Clearance Service (CCS) is introduced which determines the charge and credit to each node involved in the transmission (source, destination and intermediary) of a message. When a node receives a message, it keeps a receipt of the message and later reports it to the CCS. Payments and charges are determined out of a game theory perspective.

The sender is charged, in order to prevent a denial-of-service attack to the destination by sending it a large amount of traffic. Forwarding is considered successful if and only if the next node on the path reports a valid receipt to the CCS. Modeling the submissions of receipts regarding a given message as a one-round game, the authors proved the correctness of the receipt submission system using game theory [4, 16, 17].

B. *Reputation based schemes*

Reputation schemes however use nodes' reputation to identify and isolate selfish behavior. Several reputation systems have been proposed to mitigate selfishness and stimulate cooperation in mobile ad hoc network, including CORE, CONFIDANT and OCEAN.

1) CONFIDANT: Buchegger et al. propose a reputation based protocol called CONFIDANT, for making misbehavior unattractive [3, 4, 5]. CONFIDANT stands for Cooperation Of Nodes: Fairness In Dynamic Ad-hoc Networks, it aims at detecting and isolating selfish nodes. CONFIDANT is an extension of a reactive source-routing protocol [3]. With CONFIDANT, each node has four components;

- The monitor, which works the same way as watchdog [2], where nodes locally monitor deviating behavior.
- The trust manager, which makes decisions about route information.
- A reputation system that manages a table consisting of entries for nodes and their rating.
- The trust manager which makes decisions concerning route information

Each node monitors the behavior of its neighbors assuming bidirectional communication symmetry on every link. If suspicious behavior is detected the information is given to the reputation system. And if the rating turns out to be intolerable the path manager proceeds to delete all routes containing the misbehaving node from the path cache. Buchegger et al. later improved the CONFIDANT protocol in [6] to cope with false reputation information.

2) CORE: P. Michiardi et al. proposed a mechanism called CORE (COllaborative REputation mechanism) to enforce node cooperation in mobile ad hoc networks [4,7]. This generic mechanism can be integrated with network functions such as packet forwarding, route discovery etc. It also stimulates node cooperation by using a collaborative monitoring technique and a reputation table. CORE defines three types of reputation [7, 9];

- Subjective reputation, this is a reputation value

which is locally calculated based on direct observation.
- Indirect reputation which is a second hand reputation information established by other nodes.
- Functional reputation related to a certain function, where each function is given a weight as to its importance.

Each node computes a reputation value for every neighbor again assuming bidirectional communication symmetry on every link using a reputation mechanism that differentiates between the above mentioned reputations. CORE consists of two basic components: a watchdog mechanism and a reputation table. ”The watchdog mechanism [2, 7] is used to detect mischievous nodes. And the reputation table is a data structure stored in each node.

3) OCEAN: S. Bansal et al. proposed an Observation-based Cooperation Enforcement in Ad hoc Networks (OCEAN) [8]. In contrast to CONFIDANT and CORE, OCEAN avoids indirect reputation information therefor using only one-hop observations of other nodes behavior. OCEAN has five components in each node to detect and extenuate misbehavior.

- Neighbor Watch, works in the same way as watchdog[2]. Where nodes observe the behavior of its neighbor.
- Route Ranker, maintains the rating for each of its single hop neighbors tha same way as pathrater[2].
- Rank Based Routing, this module applies the information from the NeighborWatch in the selection of routes.
- Malicious Traffic Rejection: rejects traffic from nodes it considers misleading/liars.
- Second Chance Mechanism,is intended to allow nodes previously considered misleading to become deisolated.

This approach is to disallow any second or third hop reputation exchanges. Routing decisions are made based solely on direct neighbor observations, eliminating most trust management complexity. The main focus of OCEAN as described in [4], is to focus on the robustness of packet forwarding and therefore maintaining the overall packet throughput of a mobile ad hoc network with the existence of misbehaving nodes at the routing layer.

III. Problem Statement

To understand the importance of cooperation in ad-hoc networks, we have to clarify at first about what kind of problem we are dealing with. In the scenario that we propose, there are some key points we need to emphasize. What we want to accomplish with our algorithm, is to avoid selfish behavior, detect false reputation and encouraged cooperation. In order to accomplish these goals, we first have to quantify a nodes reputation, which means that reputation values should be assigned to each node. Nodes reputations are collected by direct and indirect monitoring of nodes. This indicates that the information used to calculate a particular nodes reputation, not only comes from its neighbors (one-hop away nodes), but also from other nodes which are 2 or more hops away. This indirect reputation monitoring leads to another problem; the credibility of reputation values from indirect neighboring nodes. In indirect monitoring, malicious nodes may forward or transmit faulty reputation values, therefor decreasing the throughput of the whole system. These malicious nodes are considered as liars. A certain mechanism for preventing liars among nodes therefore is necessary. The reputation value from direct monitoring is generally considered to be more precise. For instance as shown in figure I, if a certain node A is a direct neighbor of another node B while a third node C is a direct neighbor to B and not A, the reputation of B can be calculated using $R(B) = a * R(A, B) + b * R(C, B)$ where $a + b = 1$ and $a > b$.

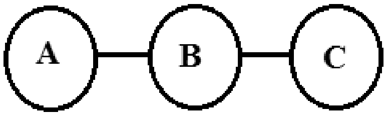

Figure I

The above part states the problem of avoiding selfishness and inspiring good behavior among nodes to improve the network performance. However, there are several assumptions we need to make for solving the problem. And also constraints exist in the scenario we proposed.

A. Assumptions

We have to make the following assumptions in order to reasonably analyze the problem. i) Nodes are well connected in the network and each node has at least one direct neighbor.
ii) Reputation values are between 0 to 1 and are set to be 1 at the beginning for each node.
iii) The value increases when the node cooperates, and decreases when it does not cooperate.
iv)Cooperation means relaying packets from other nodes.

Level of cooperation is defined only on the action of dropping packets, regardless of intentional or unintentional selfish behavior.
v)Every node has a reputation table that has the reputation values of all the nodes in the network which is updated and propagated through the network proactively.

B. Constraints

To simplify the analysis or due to the assumption we made, there are also some constraints in the problem we stated.
i) Due to the dynamic topology of ad-hoc network, nodes are coming into or going out of the network, which makes it difficult to keep track of their reputation values.
ii) Reputation propagation leads to network congestion and large overhead in the packets.
iii) The unified solution does not deal with the different QoS requirements (e.g. packet loss) of various kinds of services, i.e. message or video.
iv) Dropping packets can be due to other reasons like physical collisions, faulty nodes with no resources, etc.
v) Security issues of the network are not considered in our study.

IV. Our Approach

As mentioned previously we propose a simple reputation algorithm to compute global and local reputation of nodes participating in the network. We make a distinction between global and local reputation. Global reputation is achieved by the exchange of reputation information among the nodes in the network. And local reputation is computed based on direct monitoring single hop neighbor node behavior. Since indirect or non-neighbor nodes may forward untrustworthy reputation information, we also propose an in our algorithm a method to identify and punish false accusations. Every node in the network keeps a reputation table of every node in the network, as shown in Table I.

TABLE I
Reputation Table

Node ID	Reputation value
A	1
B	1
C	1
D	1
E	1

After reputation values have been calculated, we set a reputation threshold where selfish behavior can be detected. When a node is found to be selfish, we can isolate it from the network temporarily. If this node begins to cooperate after its isolation, we update its reputation value above the threshold again. However, if it continues to behave selfishly, after several times of temporarily isolation, we isolate the node from the network permanently, this is called punishment. So far, we have only addressed the problem of avoiding selfishness and stimulate good behavior among nodes to improve the network performance. This approach is further discussed below using our algorithm.

Algorithm

Step 1: Initialize the network
Step 2: Create a reputation table for every node. An initial reputation value of nodes inside the reputation table is one (1). For example, the reputation table of node A in figure II is as follows:

TABLE II
Reputation Table

Node ID	Reputation value
B	1
C	1
D	1
E	1

Step 3: Send packets by flooding, which have reputation tables in its header.
Step 4: Compute the reputation of direct neighbors by the following equation.

Reputation value of B computed by A is as follows:

$$R(A,B) = \frac{NumberOfPacketsForwardedByNodeB}{TotalNumberOfPacketsSentToNodeBFromA}$$

Step 5: Update the reputation values of the direct neighbors in the reputation table.
Step 6: Detect liar i.e. one who gives false reputation values of other nodes using liar test as shown below.

$$L = R(A,B) - R(C,B)$$

Step 7: if $L > 0.3$ (minimum deviation of reputation values), C is a liar, then rep. table is not updated using the values from rep. table of C. Go back to step 3. Else, the rep. value of a node in the table is updated by the following equation.

$$R(A,B) = \frac{R(A,B)+R(C,B)}{2}$$

Step 8: After a certain time t_1, If $R(A,B) \leq 0.4$ (threshold value). The node B is declared selfish and it is isolated temporarily from the network for time t_2 and a counter c is incremented by one. If $count \geq 3$ the node is permanently isolated from the network. Note: Counter c is initialized to a value Zero. Else i.e. $R(A,B) > 0.3$, Go back to step 3.

A. Results

We have simulated a random wireless ad-hoc network using Matlab as shown in Figure II and implemented reputation tables for all the nodes in the network. Reputation values of the nodes are updated in these tables by direct and indirect monitoring.

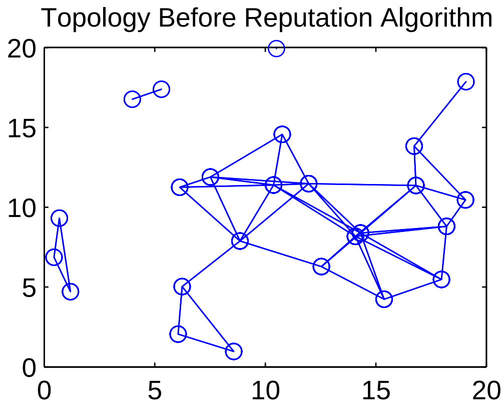

Figure II

We implemented a deviation test to detect liars in the network and prevented reputation values from such nodes to be propagated in the network during indirect monitoring. As the nodes agree on reputation values selfish nodes are identified and afterwards isolated temporarily or permanently as shown in figure III based on repeated selfish behavior.

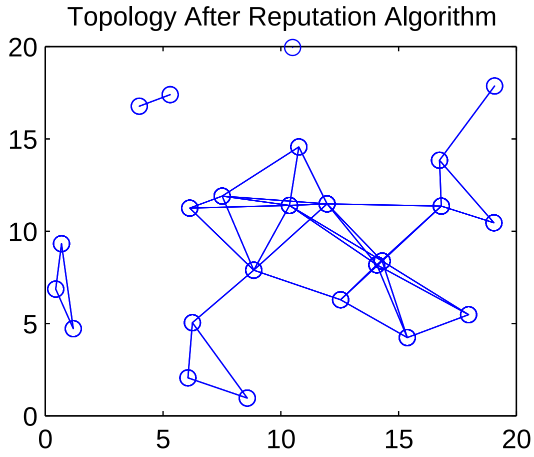

Figure III

Figure IV shows the reputation values of the nodes in the network after imlementing our algorithm. Figure V and figure VI shows how the reputation values may vary for the nodes in the network if we don't handel liars in the network i.e no liar test in the algorithm. In such a case there may exist two cases. Figure V shows the first case were the selfish node still finds a place in the network whereas figure VI shows the second case were even good cooperating nodes are isolated from the network due to liar not being handled in the algorithm. So such situation may degrade the system performance.

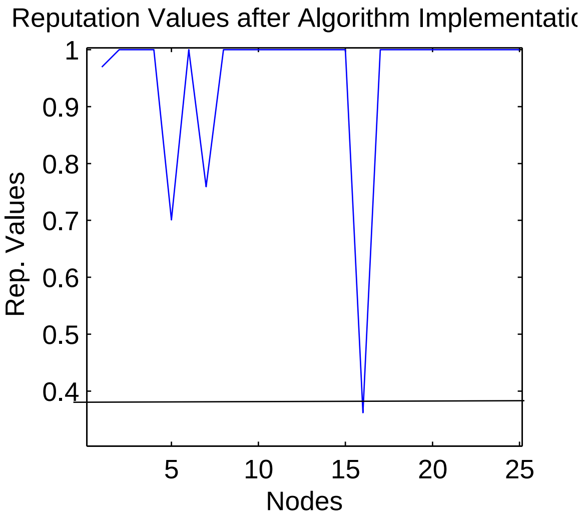

Figure IV

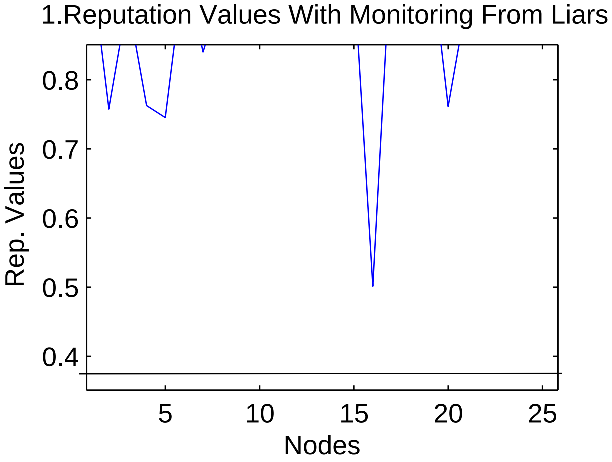

Figure V

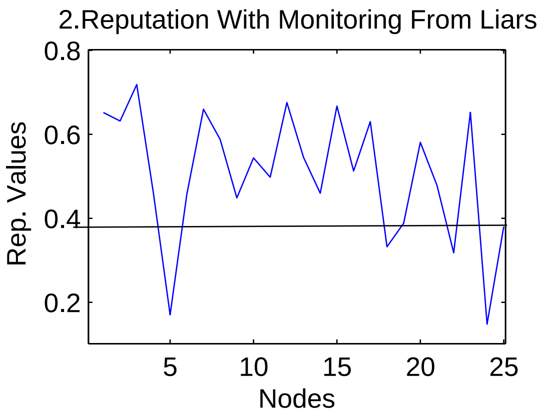

Figure VI

Figure VI shows the computed reputation values after the algortihm has identified the liars in the system and isolate them therefore ignoring their reputation information. As we can see lying behavior affects the overall system performance drastically.

V. Conclusion

Cooperation between nodes becomes trivial in self organized, wireless ad-hoc networks. So, all the nodes in the network should cooperate by relaying packets from others for better system performance i.e throughput. Often networks have nodes which drop packets and act selfishly. Such selfish behavior is controlled by using above algorithm and thus inspiring good behavior. Reputation values for the nodes are propagated untill all the nodes get to an agreement i.e. global reputation. Reputation values from liars are identified and not propagated through the network. Selfish nodes are isolated temporarily or permanently based on repeated selfish behavior. This approach promotes good behaviour between the nodes and increases the overall system performance.

VI. Future Work

The next step is to investigate our algorithm over time, considering transient removal and convergence to a reputation value and determine whether the performance converges to a threshold and if so when. In regards to the simulation, we have implemented our algorithm using MATLAB, but a simulation carried out using software designed to simulate networks such as OMNET or OP-NET are better to infer the faulty character of malicious nodes. Also by extending our algorithm to cope with other attacks such as route diversion and security can be further investigated. In our current algorithm we identify nodes that forward false reputation we assume that nodes cannot pretend to be some other node in other to increase their reputation. We would also like to extend our algorithm to minimize overhead of reputation tables therefor also coping with energy constraints. Ultimately we would like to make our algorithm as effective as possible for scalable Adhoc networks.

Acknowledgment

We wish to acknowledge Ertan Onur, Kishor Chandra, and other contributors from the Embedded Software Group at the University of Technology in Delft for their help provided which have been used in the preparation of this work.

References

[1] Wu S.L., Tseng Y.C Wireless Ad Hoc Networking, Auerbach Publications. In *Proceedings of IJCAI, http://en.wikipedia.org/wiki/Wireless*$_ad-hoc_network,$ 2007.

[2] S. Marti, T.J. Guili, K. Lai and M. Baker. Mitigating Routing in Mobile Ad Hoc Networks In *MOBICOM 2000 Boston, USA.*

[3] S. Buchegger , J.Y. Le Boudec. Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks) In *MOBIHOC02, June 9-11 2002, EPFL Laussane, Switzerland.*

[4] Jiangyi Hu. Cooperation in Mobile Ad Hoc Networks In *Computer Science Department, January 2005, Florida State University, USA.*

[5] S. Buchegger , J.Y. Le Boudec. Coping with False Accusations in Misbehavior Reputation Systems for Mobile Ad-hoc) In *EPFL Technical Report, 2003, EPFL Laussane, Switzerland.*

[6] S. Buchegger , J.Y. Le Boudec. A Robust Reputation System for P2P and Mobile Ad-hoc Networks, In Proceedings of the Second Workshop on the Economics of Peer-to-Peer Systems) In *2004, EPFL Laussane, Switzerland.*

[7] Pietro Michiardi, Refik Molva. Core: A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks) In *IFIP-Communicating and Multimedia Securtiy Conference, 2002.*

[8] S. Bansal and M. Baker. Observation-based Cooperation Enforcement in Ad Hoc Networks) In *http://arxiv.org/pdf/cs.NI/0307012, July 2003.*

[9] Pietro Michiardi, Rek Molva. Reputation methods for routing security for mobile ad hoc networks, in: Proceedings of SympoTIC '03 In *Joint IST Workshop on Mobile Future and Symposium on Trends in Communications, Bratislava, Slovakia, October 2003, IEEE Press, 2003, pages 130-137.*

[10] Dave B. Johnson and David A. Maltz. The dynamic source routing protocol for mobile ad hoc networks. In *Inter- net Draft, Mobile Ad Hoc Network (MANET) Working Group, IETF, October 1999.*

[11] Charles E. Perkins and Elizabeth M. Royer. Ad hoc On-Demand Distance Vector Routing In *Proceedings of the 2nd IEEEWorkshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999, pp.90-100.*.

[12] L.Buttyan and J.-P. Hubaux. Enforce Service Availability in Mobile Ad-Hoc WANs In *In proceedings of MobiHoc, 2000*.

[13] L.Buttyan and J.-P. Hubaux. Toward Mobile Ad-Hoc Wans: Terminodes In *Technical Report No. DSC/2000/006, Swiss Federal Institute of Technology, Lausanne, July 2000*.

[14] L.Buttyan and J.-P. Hubaux. Stimulating Cooperation in Self Organized Mobile Ad Hoc Networks In *Technical report No. DSC/2001/046, Swiss Federal Institute of Technology, Lausanne, August 2001*.

[15] L.Buttyan and J.-P. Hubaux. a Virtual Currency Stimulate Cooperation in Self Organized Mobile Ad Hoc Networks In *Technical report No. DSC/2001, Swiss Federal Institute of Technology, Lausanne,2001*.

[16] Sheng Zhong, Jiang Chen, and Yang Richard Yang. A simple, Cheatproof, Credit-based System for Mobile Ad hoc Networks In *Proceedings of IEEE Infocom '03, San Francisco, CA, April 2003*.

[17] Pietro Michiardi, Refik Molva. Game theoretic analysis of security in mobile adhoc networks In *Research Report RR-02-070, April 2002*.