



Various Protocols to Manage Cooperation and Reputation in MANET (A Review)

Deeksha Narula
GIMT, KURUKSHETRA
India

Mohit Lalit
ASSTT. PROFESSOR, GIMT, KUK
India

Parveen Chaudhary
ASSTT. PROFESSOR, SDDIET, PKL
India

Abstract—MANETs rely on the cooperation of nodes for packet routing and forwarding. However, much of the existing work in MANETs assumes that mobile nodes 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 behaviour of mobile nodes in MANETs. These protocols are surveyed in detail in this paper.

Keywords— MANET, Reputation, Cooperation

I. INTRODUCTION

A MANET (Mobile Ad-hoc Network) is a self configuring system of mobile nodes connected by wireless links. In a MANET, the nodes are free to move randomly, changing the networks topology rapidly and unpredictably. MANETs are decentralized, and therefore all network activities are carried out by nodes themselves. Each node is both an end-system as well as a relay node to forward packets for other nodes. Although, the nodes cooperation compensates the lack of a fixed infrastructure, but the lack of a central power which makes a proper interaction between the nodes, lead to some undesirable behaviours from them in the network. Such behaviours can be done by a selfish node which doesn't want to cooperate in network operations. Such a node doesn't want to consume its resources for other nodes. In general, mechanisms that try to mitigate and stimulate the misbehaved or uncooperative node can be classified into two classes (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 behaviour to mitigate the selfish node behaviour. In the following, we illustrate these two classes in more details[1].

II. VIRTUAL CURRENCY BASED SCHEME

Virtual currency schemes use some form of incentive to enforce nodes cooperation. Nodes get the incentives upon serving the network and use these to gain service from the network. If a node does not have any incentives, it will not get any service from the network. 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 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 example of such systems are: Nuglets and Sprite.

A. Nuglets

Buttayan and Hubaux introduced a virtual currency scheme called nuglets, and present a mechanism of charging/rewarding service usage/provision to stimulate cooperation in self-organized mobile ad hoc network. Two models were presented for using the nuglets: packet purse model, in which the source of the packet is charged and packet trade model, in which the destination is charged. In the packet purse model, when sending the packet, the source loads it with a number of nuglets sufficient to reach the destination. Each intermediate node takes some nuglets for the forwarding service. In the packet trade model, packets are traded for nuglets by intermediate nodes. Each intermediary node buys the packet from the previous node for some nuglets and sells it to the next node for more nuglets. To implement either the packet purse model or the packet trade model, tamper-proof hardware is required at each node. Mechanisms that use nuglets have some other problems[4].

B. Sprite

S. Zhong et al. proposed Sprite, a simple, cheat-proof, credit-based system for mobile adhoc networks. Sprite uses credit to provide incentives for mobile nodes to cooperate and report actions honestly. The basic idea of their scheme is as follows: a Credit Clearance Service (CCS) is introduced to determine the charge and credit to each node involved in the transmission of a message. When a node receives a message, the node keeps a receipt of the message and later reports it to the CCS when the node has a fast connection with the CCS. In this scheme, the sender is charged, in order to prevent a denial-of-service attack to the destination by sending it a large amount of traffic. A node that has tried 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].

III. REPUTATION BASED SCHEME

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. 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.

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.

Global vs. Local Reputation: Global reputation refers to the case where every node knows the reputation of every other node in the network. In local reputation, however, information is based only on direct observations of one-hop neighbors.

In general reputation mechanisms can be classified into two classes: (i) the reputation value is updated based global reputation information such as the case in CONFIDANT and the CORE (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.

B. Global Reputation Protocols

1) Confident:

Buchegger and Boudec present a reputation based protocol, called CONFIDANT, for making misbehavior unattractive. CONFIDANT stands for Cooperation Of Nodes. Fairness. CONFIDANT aims at detecting and isolating uncooperative nodes, thus making it unattractive to deny cooperation. With CONFIDANT, each node has the following four components: a monitor, a trust manager, a reputation system and a path manager. These components interact with each other to provide and process protocol information.

MONITOR: The monitor is the equivalent of a neighbour watch where nodes locally monitor deviating behavior. The monitor reports any suspicious events and any incoming ALARM messages to the trust manager.

TRUST MANAGER: The trust manager makes decisions about providing or accepting route information, accepting a node as part of a route, or taking part in a route originated by another node.

REPUTATION SYSTEM: The reputation system in this protocol manages a table consisting of entries for nodes and their rating. The rating is changed only when there is sufficient evidence of malicious behaviour.

PATH MANAGER: The path manager performs the following functions: path re-ranking according to reputation of the nodes in the path; deletion of paths containing malicious nodes, action on receiving a request for a route from a malicious node (e.g. ignore, do not send any reply) and action on receiving request for a route containing a malicious node in the source route.[5]

2) Core:

P. Michiardi et al. proposed a mechanism called CORE (Collaborative Reputation mechanism), to enforce node cooperation in mobile ad hoc network. 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. Members that have a good reputation can use the resources while members with a bad reputation, because they refused to cooperate, are gradually excluded from the community.

CORE defines three types of reputation

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

CORE consists of two basic components: a watchdog mechanism and a reputation table. The watchdog mechanism is used to detect misbehaviour nodes. When a node forwards a packet, the node's watchdog verifies that the next node in the path also forwards the packet. If the next node does not forward the packet, then it is considered as misbehaving. The reputation table is 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 behaviour, a list of the recent indirect reputation values provided by other entities and the value of the reputation evaluated for a predefined function.[5]

C. Local Reputation Protocols

1) Ocean:

Observation-based Cooperation Enforcement in Ad hoc Networks, OCEAN 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) Neighbour-Watch monitors the behaviour of the neighbour node. Whenever misbehaviour is detected, Neighbour-Watch reports to the Route-Ranker, which maintains ratings of the neighbour nodes.

(b) Route Ranker: Maintains a rating for each of its neighbouring nodes. Each node is initialized to Neutral (0), every positive behaviour resulting in an increment (+1) of the rating, and every negative behaviour 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 avoid some route contains misbehaved nodes,

- (d) Malicious Traffic Rejection: rejects any traffic from misbehaved nodes.
 (e) Second Chance Mechanism: gives another chance to the node that misbehaved to return to become cooperative node [5].

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

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

where R_t and R_{μ} 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 neighbour node, called M, with respect to a certain node X drops below the untrustworthy threshold R_{μ} , then M is considered as a misbehaving node by node X. After that, node X will notify its neighbours about M's misbehaviour 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 neighbourhood, where k is the number of nodes in neighbourhood 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, its reputation value increases by μ , while if it discards the packet, its value is decreased by s where $s > \mu$ [3].

3) CEPF:

In the previous approaches, when a node recognized to be a selfish node, it will be punished, but there are some questions. How long a selfish node should be punished? There are times that a node should drop a packet as its queue may become full. Now, shouldn't such nodes be given a second chance?

This approach is also a reputation based approach which uses only the first hand or direct observations for upgrading the reputations. However, by using this mechanism, there is no choice to incriminate a node as a selfish node and the problems of previous approaches such as sending too much warning messages and also the lack of confidence about sending warning messages will be solved. In fact, only first hop neighbours who checks the behaviours of nodes can upgrade a node reputation and the other neighbours do not have this permission. In this approach, at the beginning, the reputations of all the nodes are the same. With any good behaviour, the reputation will be increased and with any bad behaviour the reputation will be reduced. In this approach, by using a priority processing system, the co-operator nodes can receive their services earlier than the nodes which were selfish and haven't cooperation in network. In other words, the request of co-operator node will be responded earlier and this is the encouragement of co-operator nodes. This approach consists of three main parts: Checking System, Reputation System, and Priority Processing System as below[1]

- *Monitoring system*

In each node, there is a watchdog module that its duty is monitoring the neighbour nodes and observing their behaviours. First, each node will check its first hop neighbours, and then it will save the number of packets which are sent and received by the nodes and next it will send them to the reputation system. This module upgrades the saved information in a specific time period

- *Reputation system*

The reputation system uses the proportion of the number of Packets which are sent by a node to the number of Packets which are received by a node as the cooperation coefficient of a node. This coefficient is the same as Reputation and considered as follow:

$$\alpha (\text{Cooperation Coefficient}) = \frac{\text{Number of Sent Packets}}{\text{Number of Received Packets}}$$

In each node there is a table which is used to maintain the reputation of the nodes which should check and monitor (1st hop neighbours). The information of this table will be upgraded depending on the values which are sent by the monitoring system.

α is a number between 0 & 1. The values which are near to 0 show that the cooperation of node is low and it is a selfish node but the values which are near to 1 show that the cooperation and as a result the reputation of node is high.

- *Priority Processing System*

The decisions of this part should be made base on the information which are produced by the reputation system. In this approach, any changes in the reputation of nodes can change the priority of them in receiving services. According to this matter that each node has a queue for forwarding the packets, in our proposed approach, this **queue considers as a priority queue that within, the received route request messages**, according to their priority, will be pushed. However, the priority module determines the priority of each packet depends on the cooperation coefficient field of it. In each node, when it receives one packet, the forwarding of packet is possible and it will forward the packet directly. But, when it receives multiple packets and the concurrent forwarding of packets is not possible, the received packets should wait it in the priority queue. In fact, in the priority processing system, before pushing a packet in the queue, the cooperation coefficient of packet will be considered. In other words, the packet which is in the front of the queue will be forwarded earlier as its sender node had the highest cooperation coefficient between other dispatcher nodes and the packet which is in the back of the queue will be forwarded later as its sender node had the lowest cooperation coefficient and it was the most selfish node between other sender nodes. For example, in fig.1, V5 receives multiple route request packet. Here, the priority is with the node which had the highest cooperation coefficient. Now, if we assume that node V0 has the highest cooperation coefficient, node V5 will pass its packet first.

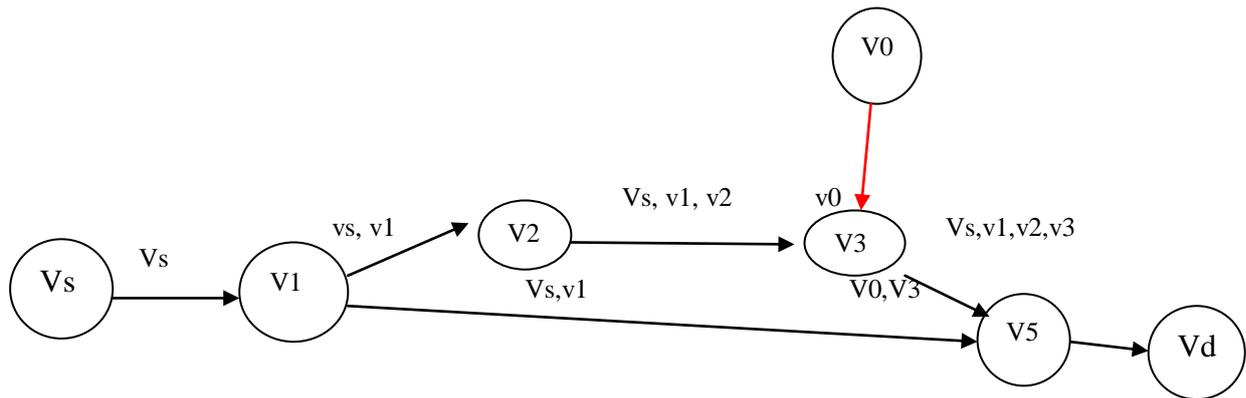


Fig. 1. In the example, node V5 will pass the packet with the highest cooperation coefficient first

IV. CONCLUSION AND FUTURE WORK

We conclude that various protocols to manage cooperation and reputation in manet may detects previously selfish nodes so as to regain their functions in the network. It may considers the priority of cooperator nodes to service them earlier and also it may punish the selfish nodes by service them later. These remarks can be proved by simulation results so the cooperation enforcement mechanism increase the probability of a successful forwarding, and the performance for networks.

REFERENCES

- [1] Zahra Safaei, MasoudSabaei, FatemehTorgheh, "An Efficient Reputation-Based Mechanism to Enforce Cooperation in MANETs", Published in IEEE 2009.
- [2] Kun Wang and Meng Wu, "A Trust Approach for Node Cooperation in MANET", 2007.
- [3] Claudio Lavecchia, PietroMichiardi, RefikMolva, "Real Life Experience of Cooperation Enforcement Based on Reputation (CORE) for MANETs"2003.
- [4] K. Mandalas , D. Flitzanis , G.F. Marias , P. Georgiias , "A Survey of Several Cooperation Enforcement Schemes for MANETs" , published In IEEE International Symposium of Signal Processing and Information Technology (2005).
- [5] Jiangyi Hu "Cooperation in Mobile Ad Hoc Network" January 11, 2005.
- [6] Jamal N. Al-Karaki, Ahmed E. Kamal "Stimulating Node Cooperation in Mobile Ad hoc Networks"