A Reputation-based Incentive Framework for Mobile Ad Hoc Networks

Vivek Richhariya
Department of Computer Science & Engineering
LNCTS, Bhopal
Madhy Pradesh, India

Praveen Kaushik
Department of Computer Science & Engineering
MANIT, Bhopal
Madhy Pradesh, India

ABSTRACT
Abstract—In mobile ad hoc networks (MANIT), nodes forward the packet to other with the help of intermediate nodes within the transmission range and as they are expected to cooperate to make the networks reliably. In ad hoc network, node may have limited resources. Due to this, some nodes (selfish node) may not to forward packets to save resources for their own use. To discourage such misbehaviour, we propose reputation-based incentive mechanism to motivate the selfish nodes to cooperate in order to packet forwarding. Incentive will be earned by the intermediate nodes which are responsible for forwarding the packet. In this paper, a cluster head will be used as reputation management of each node in the network. This paper highlights various views of cooperation enforcement mechanism and reliability. We perform an overall analysis of our paper by simulation using the network simulator (NS- 2) with the help of AODV protocol.

Keywords
Incentive scheme; reputation; ad hoc network

1. INTRODUCTION
A mobile ad hoc network is a collection of wireless nodes that can dynamically be set up anywhere and anytime without using any pre-existing network infrastructure. In other words, it is an autonomous system of mobile hosts in which they are connected by wireless links are free to move randomly and often act as routers at the same time. Every node in an ad hoc network must be willing to forward packets for other nodes. Therefore, every node in MANETs acts both as a host and as a router [1].

When data transfer is required between any pair of non-adjacent nodes, the network relies on the nodes between them to forward data packets. However, because mobile nodes are typically constrained by power and computing resources, so a selfish node may not be interesting to use its resources to always forward packets that are not of its concern, even though it would expect others to forward its packets [2]. In this circumstance, encouraging the nodes' cooperation in the packet relaying process is of primary importance. Therefore we want to motive the node become cooperative by assigning different incentive and instead of punishing the selfish node.

In this paper, the detection of selfish node are performed by using promiscuous overhearing of neighboring node when node drop packet in order to save their energy. Apart from this, the reputation value and incentive value of each node are placed at the cluster head. With these values cluster head isolate the selfish node from the network.

The remainder of this paper is organized as follows. Section 2 provides types of non cooperative nodes. Section 3 presents related works in node cooperation in MANETs using reputation approach. The overview of the proposed system is presented in section 4 and section 5 respectively, followed by the simulation result which was implemented on the NS2 simulator in section 6. Then, the conclusion is drawn in section 7.

2. TYPES OF NON COOPERATIVE NODES
In an ad hoc network, the communication range of mobile nodes is limited on account of power constraint. Therefore, when communication is done between two nodes beyond the transmission range, node depends on intermediate nodes to forward the packets. Due to this reason, sometimes these intermediate nodes do not work as expected. In order to preserve their limited resources such as bandwidth, energy etc, such nodes are called non cooperative nodes or misbehaving nodes[3]. They are of following types:

Selfishness
The limited battery-power, one of MANET characteristics, encourages nodes to use the network for their own communication only, and not for the gain of other nodes. Refer to routing protocols [4], the following selfish behaviors are considered.

- Do not participate in route discovery process (Category 1): In this category, a selfish node drops routing messages or it may modify the Route Request and Reply packets by changing TTL value to smallest possible value. For example, in DSR protocol, selfish nodes may drop all the route request packets they receive or not forward a route reply packet to some destinations.
- Participate in route discovery process but not forward data packet (Category 2): In this category, a selfish nodes participates in the routing protocol, but may drop part or all the data packets that do not belong to it. This node is interested in saving its battery power, apart from having the capability to receive and forward its own packets.
- Do not reply or send hello messages (Category 3): In this category, node enters to idle status most of the time and does not even send HELLO messages to its neighbors, so that they are not aware to its existence. Only when it wishes to communicate with other nodes, it starts the routing protocol. This behavior, called "sleep period operation", .

- Intentionally delay the RREQ packet (Category 4): A selfish node may delay the RREQ packet up to the maximum upper limit time. For this ,it will certainly keep off itself from routing paths.
- Other condition (Category 5): Node usually performs the routing and the forwarding properly, but when its energy falls under some threshold or in case of temporary overload, it may act as nodes of category 1, 2, 3 or 4.

Malicious Node
Malicious nodes aim to damage other nodes without considering their own gain or their battery life as a main concern. If malicious nodes are exist in a MANET, they may attempt to reduce network connectivity by pretending to be cooperative.

Since providing services or forwarding messages will incur a cost to a node, a selfish node probably does not provide services or forward messages, therefore we need providing some incentives to the selfish nodes to encourage them to provide services or forward others' messages. The purpose of applying the incentive and reputation mechanism to the ad hoc networks is to encourage all nodes in the network cooperating with each other honestly, and to make the network more reliable.

3. RELATED WORK

The problem of nodes cooperation in MANETs has received a lot of research interest now a day. More recently, cooperation enforcement methods have been proposed for trust establishment in MANET. These proposed schemes, categorised credit-based and as reputation-based, are considered suitable for ad hoc networks, where key or certificate distribution centers are absent or ephemerally present. For MANITs that consist of devices with limited memory resources, battery and processing. Cooperation enforcement methods do not provide strong authentication of entities. Rather, they contribute to the identification of the trustworthiness of peers and to the enforcement cooperation using mutual incentives. Standard Recently, a lot of research has focused on the cooperation issue in MANET. Several related issues are briefly presented here.

Buchegger and Le Boudec [5] present the CONFIDANT protocol. CONFIDANT deals with not only the selfish but also several types of misbehavior such as silent route change or frequent route updates. Each node monitor the behavior of its next hop neighbors in a similar manner to watchdog. But deciding the criteria for maintaining the friends list by Trust Manager is difficult. Bansal et al [6] have proposed a protocol called OCEAN (Observation-based Cooperation Enforcement in Ad hoc Networks), which is the enhanced version of DSR protocol. Each node maintains the ratings for neighbor who directly interact with it. These ratings are not propagated to any other node. Due to this,OCEAN fails to deal with misbehaving nodes properly. CORE (Collaborative Reputation) [7] is a reputation based system proposed by Michiardi et al similar to CONFIDANT and aims to detect and isolate selfish nodes. The node reputation is heavily weighted towards past reputation; therefore, cooperative node with low battery condition would not be detected as misbehaving nodes right away. The limitation with CORE is that the most reputed nodes may become congested as most of the routes are likely to pass through them. Khairul Azmi et al [8] present a new mechanism to detect selfish node. Each node is expected to contribute to the network on the continual basis within a time frame. Those which fail will undergo a test for their suspicious behavior. This scheme is also a based on monitor node. A monitoring node hears a request from its neighbouring node to forward a data packet; it will first check the time difference between *last request* and *last action* and status of the requestor. Misbehavior detection and reaction are described in [9], by Marti, Giuli, Lai and Baker. The paper presents two extensions to the DSR algorithm: the watchdog and the path rater. The watchdog identifies misbehaving nodes by listening promiscuously to the next node transmission but not detect misbehavior in presence of ambiguous collisions, receiver collisions, limited transmission power, false misbehavior and partial dropping.

4. PROPOSED WORK

Our paper determines the selfish node and instead of punishing the selfish node. We want motivate the node to become cooperative by assigning different incentive. Incentive assignment is directly mapped with their contribution. In this section, we will present the basic scheme of our reputation based incentive mechanism; Our basic scheme consists of three components namely

- Neighbour Monitoring.
  - Direst Trust
  - Trust Identification by Feedback (Indirect Trust)
- Calculates the Reputation Value
- Assignment of Incentive based on Reputation Value

4.1 Neighbour Monitoring:

For this purpose, we collect the information about the packet forwarding behaviour of the neighbour. With the help of promiscuous mode, each node has capability to overhear neighbour's transmission. It helps a mobile node A to maintain the list of neighbour node $\mathbf{L_A}$. This neighbour node list maintains all of its neighbour nodes that node A learns of by overhearing. On the basis of information collected, each node calculates the trust parameters such as direct trust and reputation factor. These are vital parameters to calculate trust of each node.

4.1.1 Direct Trust (DT)

It is a measure of involvement of a node in the routing process. This parameter is equally important to reputation parameter for trust estimation. If a node actively participates in the packet forwarding process then its direct trust will be high and its value ranges from 0.1 to 1. The value of can be determined as follows.

For this, node A also maintain two values, for each of its neighbour (denoted by K) and here A be the monitoring node and K be the monitored node as below.

- $\mathbf{T_A^K}$: The total number of packets (i.e. data packets, route request packets & route reply packets) should be forwarded by K that node A has transmitted to K.
- $\mathbf{A_A^K}$: The total number of packets(route message plus data packet) that have actually been forwarded by K and observed by A.

Whenever node K receives a packet which is supposed to be forwarded either from node A or from another neighbours and node A overhears the transmission. In order to maintain the neighbour's record, above two values are updated by the following rules.

- When a packet is sent by A to K for forwarding, the value of $\mathbf{T_A^K}$ is incremented by one. Here the packet may be either route messages (route request or route reply) or data packet so that the increment would be done by one in account of forwarding route messages or data packet only.
- Since node A overhears the transmission and check whether node K forwards the packet (either route request, or route reply or data packet) as expected. If node A find out that K has forward the packet before a preset time-out expires, the value of $\mathbf{A_A^K}$ is incremented by one in account of forwarding route request or route reply or data packet only.

For $\mathbf{T_A^K}$ and $\mathbf{A_A^K}$, node A calculates direct trust values for each neighbour node K.

Direct trust is also called Neighbour sensing. When we want to know if we can trust some node Y, we can route some packet via Y and see (by promiscuous mode) weather Y forwards them correctly or not. For every packet X sends to Y, X puts a copy of it in a cache. If X sees Y forwarding the packet correctly X promotes Y for that. If X sees that Y changed the packet or if X does not see the packet for some time, X punishes Y. Then the packet is deleted from the cache. Direct trust of a particular node K is calculated by a node A as follows:

$$DT(A,K) = W(R_{req}) \times R_{req} + W(R_{rpl}) \times R_{rpl} + W(D) \times D \quad (1)$$

where W(.) is a weight assigned to a particular event, Rp, Rq, D are normalized route reply packet, route request packet and data packet respectively. The values of R_{req} , R_{rpl}, D are determined as follows:

$$R_{req} = \left[\frac{A_A^K}{T_A^K}\right] R_{req} \,, \; R_{rpl} = \left[\frac{A_A^K}{T_A^K}\right] R_{rpl} \,,$$

$$D = \left[\frac{A_A^K}{T_A^K}\right] D \quad (2)$$

and $W(R_{req}) + W(R_{rpl}) + W(D) = 1$ (3)

A trust computation method based on direct observations to establish trust among monitor nodes. Every node measures the trust of the other nodes by analyzing their behaviour over time. For instance, x observes the behaviour of y and judges whether the behaviour is correct or not. Each opportunity x has of observing the behaviour of y is recorded in an experience record cache. Over the time, these experiences will become stale. Therefore, x will assign some weight values (decreasing function with time) to the past history.

4.1.2 Trust Identification by Feedback (Indirect Trust)

During the interaction between two nodes X and Y, they provide feedback about each other based on their performance at routing and know about trust. For instance, if X now wants to get references for Y, he creates a requests, set himself as sources, sets Y as target and broadcasts it to his neighbours (ttl=1). Every node N receiving this request then looks if he has a direct trust value for Y and if yes creates a reply (from him to X) which is carrying this value. After some time X can then combine the received values (as feedback) to trust identification for Y:

$$Indirect\ Trust\ (X,Y) = \frac{\sum_{i=1}^{n} DT\left(X, N_i\right) * DT\left(N_i, Y\right)}{n} \quad (4)$$

Trust identification by feedback or indirect trust is represent by IDT. Now if intermediate node forward packet correctly to its neighbouring node, its trust value is increased by one else trust value is decrease by one.

Based on direct trust & trust identification by feedback (indirect trust), node A keep the record for each neighbours, called record of neighbour's trust (denoted by $\mathbf{R_A^K}$), for the neighbour node K. The record of neighbour's trust $\mathbf{R_A^K}$ consists of following entries.

Table 1 : Record of Neighbour's Trust $\mathbf{R_A^K}$

Node ID	Packet Sent	Packet Forwarded	Packet Dropped	Direct Trust	Indirect Trust

Where Node ID is unique id of each node. Each node maintains the record that how many packet has been sent to its neighbour and out of them that have actually been forwarded by the neighbour. Based on these information direct and indirect trust will be calculated.

4.2. Calculates the Reputation Value

The reputation value involves in the allotment of direct trust and indirect trust to the happenings monitored by neighbouring monitor. For this, node A calculate the reputation (R) for each of its neighbour (denoted by K) and here A be the monitoring node and K be the monitored node as below

$$R(A,K) = DT(A,K) + IDT(A,K)$$

Where

$$IDT\ (A,K) = \frac{\sum_{i=1}^{n} DT\left(A, N_i\right) * DT\left(N_i, K\right)}{n} \quad (5)$$

And DT(A,K) and IDT(A,K)) indicates the direct trust A on node K and indirect trust for K respectively.

4.3. Assignment of Incentive to node

Here, we provide incentives based on node behaviour adopting. A policy that makes cooperation the best option for every node to exercise. Such policies are normally a function of the observed behavior of nodes in the network. In previous section, reputation is considered as one of the metrics in the assessment of a node's behavior. According to this mechanism, a node assigns reputation values to its neighbors based on its direct interactions with them and on indirect reputation information obtained from other nodes. The analysis helps to assess the robustness of the reputation scheme against different node strategies and derive conditions for cooperation. Even though reputation schemes are effective in providing incentives, the design of a good reputation system to judge a node is often complex and involves trust management between the individual nodes.

Initially each node has fixed amount of incentive which is the essential requirement for the sender to forward its packets. When a source node wants to send packet to another node (destination), it will lose some incentive depends upon the size of packet. If intermediate node forwards a packet, its incentive increases and if it receives a packet and do not forward the packet, incentive will be decreased. The node will not be ignored just by an unsuccessful transmission, but its behaviour

will be observed for some time by its neighbouring nodes, till its incentive goes under threshold value. Once the incentive will go under threshold value, that node will be ignored, considering node's selfish behaviour and next packets will not be given to it for forwarding. In other words, the incentive will be earned by the intermediate nodes which are responsible for forwarding the packet. To earn more incentive, a node must forward others' packets.

The overall incentive assigenment can be understand by the following flow chart.

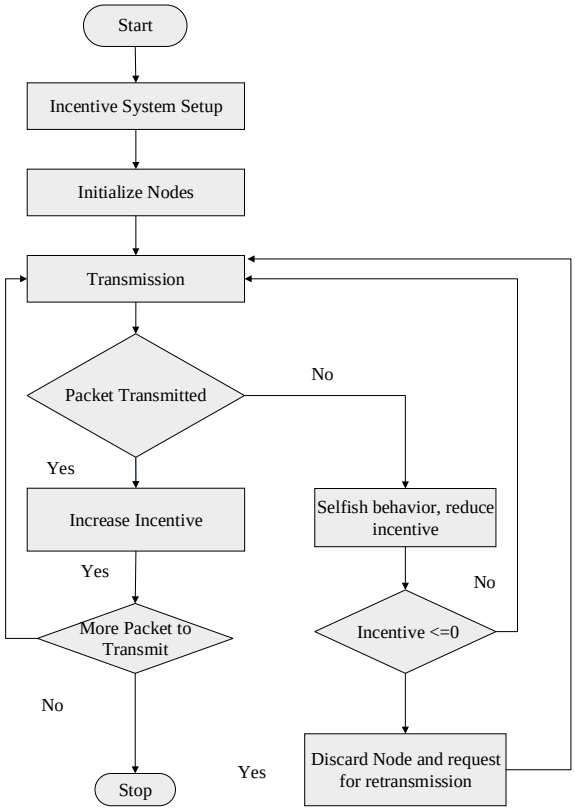

Figure 1: Flow Chart of Incentive Assignment

5. PROPOSED ALGORITHM

This section represents the basic scheme of reputation based incentive of selfish node. The network architecture in figure 4 of proposed scheme consists of n number of mobile nodes and a cluster head. In comparison to the previous, this scheme uses cluster head as a reputation manager. The advantage of using cluster head is that if it fails, a new cluster head take the responsibility.

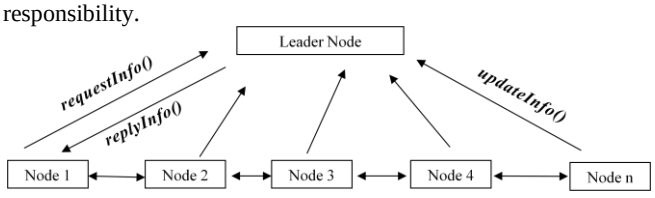

Figure 2: Flow Chart of Incentive Assignment

Proposed scheme works in three steps. Getting incentive and reputation value of each node, Detection of selfish node and Convert the selfish node to cooperative node. Let each node have fixed amount of initial incentive IN and reputation value R. During the communication of packet, if node forwards the packets, the incentive will be given to node. Once the incentive will go under pre defined threshold (IN_THRESH) value or if incentive of selfish node becomes 0 (in the case of static selfish node), that node will be ignored and next packets will not be given to it for forwarding. Now if intermediate node forward packet correctly to its neighboring node, its reputation is increased by one else reputation value is decrease by one. If reputation of any node is less than a pre defined threshold (R_THRESH), node becomes selfish. Then selfish node change its behaviour and convert to cooperative node.

The value of each node's incentive and reputation is kept at cluster head database table called INR list as shown in table 2.

Table 2: INR List

Node ID	Incentive Value	Reputation Value

Where, Node ID is a unique id of each node. The value of incentive and reputation is updated through a small message called *updateInfo* containing values *(nid, incentive, reputation).* Each time a node sends other's message to its neighbouring node, it forwards *updateInfo()* message to the clusterhead for updating incentive and reputation values in INR list.

At route discovery phase, each time a node wants to send its packet to other node, it first communicates using *requestInfo()* with the clusterhead , that knows about the node incentive and reputation value of each intermediate nodes present in the path.

requestInfo(sn_id, dn_id,int_nid (1,2,….))

Where *requestInfo()* is used to get value from clusterhead, *sn_id* is source node id, *dn_id* is destination node id, *int_nid* contain id of intermediate nodes

replyInfo(sn_id, dn_id,int_nid (1.1,2.2,….))

Where 1.1, 2.2 and so on contain value of incentive and reputation of respective intermediate nodes.

If any node is found having low incentive value and low reputation value, it is considered as selfish node. If selfish node is present in the path, isolation of such node is carried out by not appending the node in the path. Hence no packet is forwarded through that node and another path is chosen by the sender node. Then selfish node changes its behaviour and converts to cooperative node.

6. RESULT ANALYSIS

We evaluate the throughput and analyze the influence of the non-cooperation nodes. The parameters of the simulated networks are shown as follows.

Parameter	**Value**
Number of Nodes	100
Routing Protocol	AODV
Maximum mobility speed of nodes	CBR
Communication Type	10 m/sec.
Simulation Area	1000m x 1000 m
Simulation Time	250 sec
Speed	1-10 m/s
Packet Sizes	512bytes

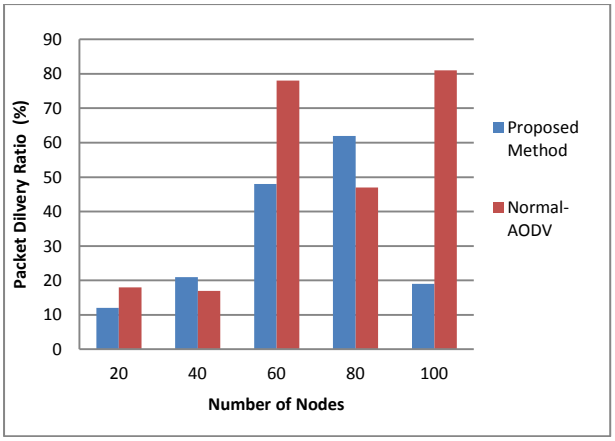

Figure 3: PDR V/s Number of Nodes

From the above figure 3, we can say that the value of PDR is not increasing constantly and lies between 82% to 89% ,when we use AODV protocal with selfish node. When we encourage the node by the incentive approach with the reputation value, the value of PDR increase from 92% to 96%.

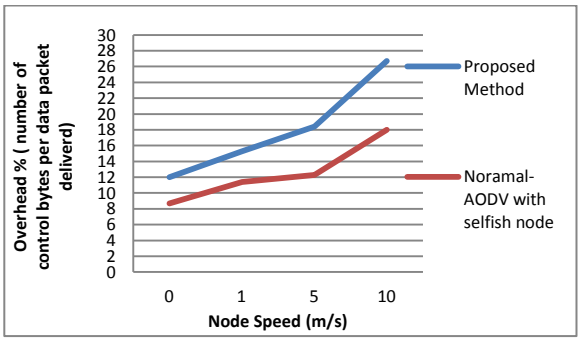

Figure 4: Overhead V/s Nodes speed

From the above figure 4, it is clear that the newly proposed Reputed-AODV protocol has a higher overhead than the normal AODV secure routing protocol. This is due to the fact that the Reputed-AODV uses extra data acknowledgement (DACK) packet for each data packet sent. This DACK packet is used to give positive recommendations after each successful data packet transfer. Thus, when nodes are moving at speed of 10 m/s, the overhead percentage rises from 18%, in case of normal AODV, to 26.7%, in case of Reputed-AODV. Though the overhead percentage added by the Reputed-AODV is significant, this reputation-based scheme still improves considerably the network throughput.

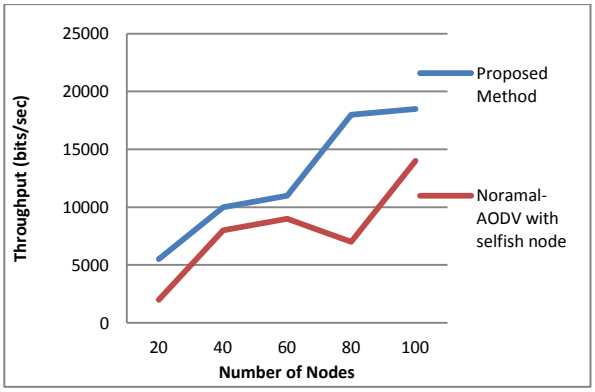

Figure 5: Throughput V/s Number of Nodes

From the above figure 5, two curves represent the throughput of normal AODV and Reputed AODV. The graph demonstrates that AODV with the reputation technique extension always performs better than the normal AODV. As we our expectation, the selfish node which failed in forwarding packets are removed from the routing cache. The result is that good path are used for transmitting packets.

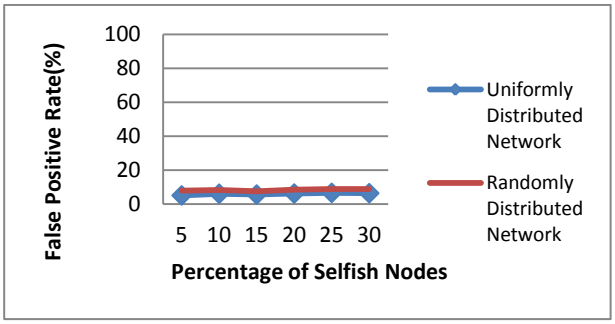

Figure 6: False Position Rate V/s Percentage of Selfish Nodes

From the above figure 6, we can see highest false positive rate is below 7% in the uniformly distributed network. In the case of randomly distributed network, the highest false positive rate is below 9% which is relatively low. The false positive is caused mainly by node movement since some link layer breaks are detected as forwarding level misbehavior. Therefore, it will decrease when the node motion become slower.

7. CONCLUSION & FUTURE WORK

In this paper, we proposed reputation based incentive system to motivate the selfish nodes to cooperate in packet forwarding or providing other services. This system is fully self-organized and the incentive of each node is determined from reputation values

The limitation of our system is that we also use a tamper resistant hardware as the protection mechanism, and it is not a practical assumption for the general scenario. The ideal approach for the incentive scheme is pure software designed; we regard this objective as future works. The results of performance evaluation show that the selfish nodes can cooperate with each other in the general situations.

8. REFERENCES

[1] C. E. Perkins, "*Ad hoc Networking*", Addison Wesley, 2001.

[2] Bing Wu, Jianmin Chen, Jie Wu, Mihaela Cardei, "A Survey on Attack and countermeasures in Mobile Ad Hoc Network", Springer 2006

[3] Qi Zhang and Dharma P. Agrawal, "Impact of Selfish Nodes on Route Discovery in Mobile Ad Hoc Networks", IEEE Communications Society, pp. 2914-2918, 2004.

[4] Matthias Hollick, Jens Schmitt, Christian seipl, (2004) ''On the Effect of Node Misbehaviour in Ad hoc Network" Proc. IEEE Conference on Communication, Vol 6, pp 3759– 3763.

[5] S. Buchegger and J-Y. Le Boudec, (2002) "Performance Analysis of the CONFIDANT Protocol:Cooperation Of Nodes, Fairness In Dynamic Ad-hoc Networks", Proc. of

the IEEE/ACMSymposium on Mobile Ad Hoc Networking and Computing (MobiHOC).

[6] S. Bansal and M. Baker. "Observation-Based Cooperation Enforcement in Ad hoc Networks", July 2003. Available on: http://arxiv.org/pdf/cs.NI/0307012.

[7] P. Michiardi and R. Molva. Core: "A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks", In Proceedings of the 6th IFIP . Communications and Multimedia Security Conference, pages 107 121, Portorosz, Slovenia, September 2002.

[8] Khairul Azmi, Abu Bakar and James Irvine. " A Scheme for Detecting Selfish Nodes in MANET using OMNET++", 2010 Sixth International Conference on Wireless and Mobile Communications, pp 411-414, 2010.

[9] Marti, S., Giuli, T. J., Lai, K., & Baker, M. (2000), "Mitigating routing misbehavior in mobile ad-hoc networks", Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom), ISBN 1-58113- 197-6, pp. 255-265.

[10] P Das S. Perkins C.E., Belding-Royer E.M. Ad-hoc on-demand distance vector (aodv) routing. RFC 3561, IETF Network Working Group, 2003.

[11] Perkins C and E. Royer, "*Ad-hoc on-demand distance vector routing*," in Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications, February1999, pp.90–100.

[12] AODV homepage. http://moment.cs.ucsb.edu/aodv/aodv.html

[13] Schiller J. Mobile Communications. Addison Wesley, 2nd edition, 2003.