ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

A COLLABORATIVE WATCHDOG AND CLASSIFIER BASED SCHEME TO DETECT AND AVOID SELFISH NODES IN MANET

Mr. Swapnil S. Shinde*, Dr. B. D. Phulpagar
*M.E. Student, Computer Engineering, P.E.S. Modern COE (SPPU) Pune - Maharashtra, India.
Asst. Professor, Computer Engineering, P.E.S. Modern COE (SPPU) Pune - Maharashtra, India.

ABSTRACT

Wireless mobile ad hoc networks are self-configuring, dynamic networks in which nodes are free to move. A mobile ad hoc network (MANET) is a continuously self-configuring, infrastructure-less network of mobile devices connected without wires, sometimes untrustworthy. From last decade, mobile ad hoc networks have become a very popular research topic. Communication range among mobile nodes in ad-hoc network is limited; hence several hopes are needed in a network to transmit a packet from one node to another node. In mobile ad hoc network, some nodes may selfishly decide only to cooperate partially, or not at all, with other nodes as its a cost intensive activity. This behavior of selfish nodes could then degrade the overall data accessibility which results into performance degradation of overall network. We surveyed some key technique for detecting selfish nodes in MANET. This paper provides a survey on different technique used to detect selfish nodes in such network as well as compare them (to study)in order to reduce the effect of selfish nodes in mobile ad hoc networks. Moreover one important aspects of this paper is to propose specific technique that would evaluate the selfishness behavior of nodes in the network in less time and effectively.

KEYWORDS: Mobile Ad hoc Networks (MANET), AODV, Self-configuring, Cost-intensive, Selfish nodes.

INTRODUCTION

Wireless mobile ad hoc networks are self-configuring, dynamic networks in which nodes are free to move. A mobile ad hoc network (MANET) is a continuously self-configuring, infrastructure-less network of mobile devices connected without wires. Ad hoc is Latin and means "for this purpose". Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Setting up of fixed access points and backbone infrastructure is not always viable Infrastructure may not be present in a disaster area or war zone. Infrastructure may not be practical for short-range radios; Bluetooth (range ~ 10m).

The term ad hoc networking typically refers to a system of network elements that combine to form a network requiring little or no planning. The network is ad hoc because it does not rely on a pre-existing infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks. Instead, each node participates in routing by forwarding data for other nodes, so the determination of which nodes forward data is made dynamically on the basis of network connectivity. Dynamic Source Routing [DSR] and AODV are some algorithms that have been designed to handle such transmission of data [3].

Applications of mobile ad hoc networks have been developed mainly for crisis situations (e.g. natural disasters, military conflicts and emergency medical situations). In these applications, all the nodes of the network belong to a single authority and have a common goal. With the progress of technology, it has now become possible to deploy mobile ad hoc networks for civilian applications as well. Examples include networks of cars parking and provision of communication facilities in remote areas. In such networks nodes do not belong to a single authority and they do not pursue a common goal. In addition, these networks could be larger, have a longer lifetime, and they could be completely *self-organizing*, meaning that the network would be run solely by the operation of the end-users. In such networks, there is no good reason to assume that the nodes cooperate. Indeed, the contrary is true: In order to save resources (e.g., battery power, memory, and CPU cycles) the nodes tend to be “selfish”.

This paper provides a survey on different technique used to detect selfish nodes in such network as well as compare them in order to reduce the effect of selfish nodes in mobile ad hoc networks. The paper comprises of following

section: Related work: shows the literature study, proposed method to detect selfish nodes, Result, Conclusion and Future Work.

LITERATURE SURVEY

Literature studied shows that there are various technique to detect selfish node, some of them are studied and explained below. Techniques used to detect selfish nodes can be classified into three categories:

Review of Detection Methods:

Reputation Based Scheme:

This scheme works in a collaborative manner. Reputation simply means to opinion about a thing. Here nodes communicate with each other in order to give feedback about particular nodes cooperative behaviour. Every node gives feedback in terms of a reputation value. In this way every node collects high reputation value to build trust and confidence about good behaviour and cooperation in network. Low reputation value is considered to be indication of selfish behaviour while high reputation value indicated cooperative behaviour of nodes. The reputation value of a selfish node is clear indication to the other nodes about its cooperation in the network. The network will detect the selfish nodes then the message about this will get propagated to the entire network and the selfish node will be eliminated from the network [9].

Credit Based Scheme:

In this scheme [8], incentive is given to cooperating nodes for the transmission function in network. Main idea here is "serve & earn". This incentive based scheme uses the concept of virtual credit or electronic currency or similar payment schemes. The incentives are given for packet forwarding in order to motivate the non-cooperative node to participate. This scheme needs a setup virtual payment system. It uses two models as-

(i) The Packet Purse Model:

(ii) The Packet Trade Model:

Acknowledgement Based Scheme:

The acknowledgement based schemes ensures the forwarding of a packet by a node using an acknowledgement. In this scheme a node sends an ack packet to source once it is being forwarded. If a source node does not get this ack packet this means misbehaviour of node is observed [14].

As the field of MANETs is increasing day by day, efforts are made to focus on the subject of securing such networks. Most challenging and vital issue is that MANETs must have secure way for transmission and communication. Many researchers and groups have proposed ways to secure the MANET from selfish node attack out of which some are presented below.

2ACK Scheme:

K. Balakrishnan et. al. [14] have proposed a scheme called 2ACK scheme which is a network layer scheme to detect the selfish nodes. This scheme uses an acknowledgement packet called 2ACK packet for detection. In this scheme the next hop node in the route will send back the 2 hop acknowledgment packet ie 2ACK. This acknowledgment packet is used to indicate that the data packet has been received successfully. The first router from the sender will not serve as the sender of 2ACK.

Watchdog:

Marti et. al. [9] have proposed the watchdog mechanism which is implemented on every node. It monitors nearby nodes in order to identify the misbehaving nodes. When a node forwards a packet to the watchdog, it checks whether the next node in the path will forward this packet or not. If watchdog observes that if the node does not forwarding the packet then it is considered as selfish node. The watchdog will avoid such selfish nodes from the routing path and selects alternative path.

Pathrater:

Marti et. al. [9] have proposed this mechanism wherea path metric is calculated for each routing path. This is achieved by setting up a mechanism called as pathrater with every node. Each node runs this mechanism and gives rating after every successful transmission of packet.After calculating the path metric for every path to the particular destination, the path with highest metric will be chosen as the most reliable path.

CONFIDANT:

Buchegger et. al. [2] have proposed a technique which is somewhat similar to watchdog and pathrater and it is known as CONFIDANT (Cooperation of Nodes, Fairness in Dynamic Ad Hoc Networks). The CONFIDANT protocol contains four important components i.e. Monitoring System, Reputation System, Trust Manager and Path Manager.

CORE:
Michiardi et. al [13] have proposed CORE (Collaborative Reputation Mechanism) system to improve the coordination among nodes. It uses two basic components which are 1) Reputation table and 2) Watchdog mechanism. It enforces the cooperation among the nodes by using reputation report mechanism. Each node performs some computation to calculate the reputation value for all neighbour nodes. The reputation report contains values ranges from positive to negative. This mechanism allows to pass only positive reputation reports.

OCEAN:
Bansal et. al. [3] have proposed a protocol called OCEAN (Observation-based Cooperation Enforcement in Ad hoc Networks). It also uses the monitoring and reputation mechanism. The OCEAN mechanism is basically having following five components 1) Neighbour Watch 2) Route Ranker 3) Rank-Based Routing 4) Malicious Traffic Rejection and 5) Second Chance Mechanism.

SORI:
Q. He et.al. [11] have proposed secure and Objective Reputation-based Incentive (SORI) scheme. This method encourages the packet forwarding. It consists of three components and they are (1) Neighbour monitor, (2) Reputation propagation and (3) Punishment.

Sprite:
Zhong et. al. [15] have proposed a scheme called Sprite. It uses a Credit Clearance Service (CCS). It is used to define the credit and charge of each node. To calculate the charges and credits it uses Game theory methods. Each node will get a receipt of message that it has received or forwarded. Each node keeps the receipt of the message and it will forward the receipt to the CCS. The credit of a node is totally depending on the forwarding behaviour of a node. The forwarding is considered as successful only if the next node on the path reports a valid receipt to the CCS. If the node forwards the message then credit will be raised otherwise credit decreases.

Selfish Node Behaviors:
Selfish nodes [16] are inclined to get the greatest profits from the networks and at the same time these nodes trying to conserve their own resources like bandwidth, battery life or hardware. A Selfish node only communicates to other nodes if its data packet is required to send to some other node and refuses to cooperate other nodes whenever it some data packets or routing packets are received by it that it has no interest in. Hence data packets are either refused to retransmit or are dropped for being received by a Selfish node. The Selfish nodes behaviors in routing can be as follows:

Nodes which do not send Hello packet:
The principle target of this sort of Selfish node is hiding itself and to abstain from being included in the others transmission way.

Nodes which do not forward RREP messages:
Because of this kind of Selfish behavior whole network will be paralyzed. In AODV, the source node will get a RREP message from the destination node through some intermediate nodes to establish a complete transmission path, but here the communication path will not be established because this kind of Selfish nodes will not forward the RREP message. Hence the source node will broadcast Route Request (RREQ) message continuously.

Nodes which do not forward Data messages:
The misbehavior of this type of Selfish node impacts the performance of MANET y dropping all the data messages that are received by these nodes. Instead of relaying these data messages these will be dropped.

Nodes forwarding RREQ messages with delay:
When this kind of Selfish node gets a Route Request(RREQ) message it forwards this RREQ message after some lag near the upper bound of time out for not to participate in a route.

Nodes which do not forward RREQ messages:
In MANET, if this type of Selfish nodes receives some RREQ messages, then instead of forwarding these RREQ messages, these messages are dropped and thus these kind of Selfish nodes skips being the route member for other nodes. Thus avoiding forwarding these messages for others as a result more nodes are required for building a transmission path.

Selfish Behavior Depending on the Nodes Energy:
This type of Selfish nodes acts normally if its energy level lies between full energy level and some threshold k1. They act like do not forward data messages Selfish node if its energy level lies between k1 and some k2 and if its energy level is below k2 then they behave like do not forward RREQ message Selfish node.

PROPOSED SYSTEM

A mobile ad hoc network (MANET) is an infrastructure-less network. In mobile ad hoc network, some nodes may selfishly decide only to cooperate partially, or not at all, with other nodes in order to save its own resources. This behavior of selfish nodes could then degrade the overall data accessibility which results into performance degradation of overall network.

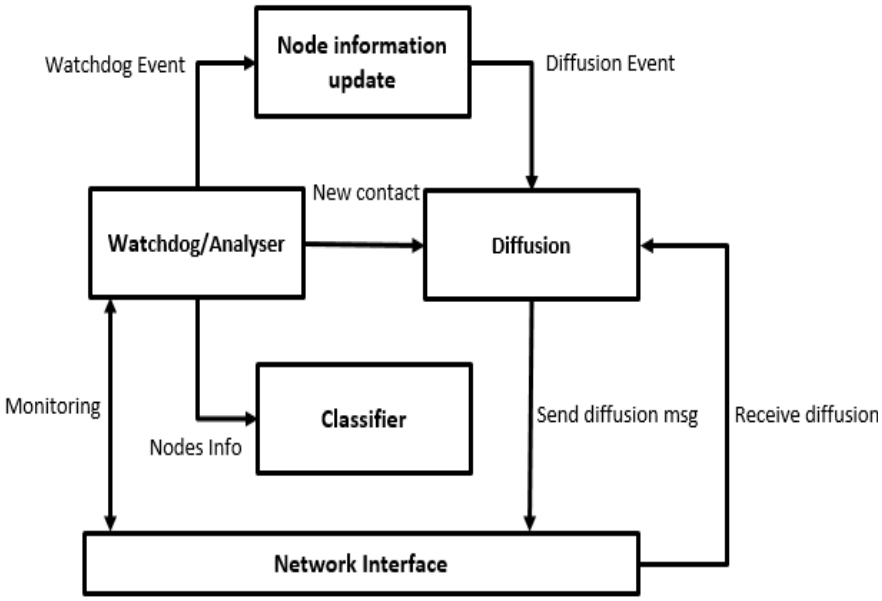

Fig -1: System Architecture of proposed model.

Process:

Following is the actual process of proposed system.
Input: Path (Source to Destination)
Processing:

1. Path formation and selfish node is generation.
2. Communication overhearing by watchdog.
3. Behavior message diffusion to neighboring nodes
4. Packet data analysis submission to Classifier.
5. Classification Process

Output: Selfish Node Detection

States in System:

In proposed system nodes have four states [1].

a) Initial state: Initially node does not have any information about any selfish node
b) Selfish contact (Positive) : It is a state when a node detects a selfish node using its watchdog and historical record
c) Collaborative contact: It is a state when contacts between pairs of nodes occurs to transmit there detection information.
d) Partial Selfish contact (Positive): It is a state when a node detects a partial selfish node using its watchdog and historical record

Mathematical Modeling:

Let S be a system defined as,

S = {sn, dn, n,p,wd,cl,path_calc}

Where,
n = nodes
p = packet
sn = source node
dn= destination node

path_calc(sn,dn) : calculate path from source node to destination node

wd = watchdog (It gets nodeinfo and checks if node is dropping packets beyond set threshold.)
wd(nodeinfo) = {spkt,rpkt,dpkt}
Where,
spkt = send packet
rpkt = received packet
dpkt = dropped packet

cl = classifier
cl(nodeinfo) = {spkt,rpkd,dpkt,isselfish}
It classifies nodes as selfish or partial selfish with the help of analyzed data by watchdog. Here classifier uses linear SVM to classify nodes.

$$\boldsymbol{D = \{(x_i, y_i) | x_i \epsilon p, y_i \epsilon \{-t, t\}\}_{i=1}^{n}}$$

Where the y_i is either t or −t, indicating the class to which the point x_i belongs. Each x_i is a *P* dimensional real vector (here x_i is nothing but the set of packet dropped by each node). We want to find the maximum-margin hyper plane that divides the points having $y_i = t$ from those having $y_i = -t$. (Where t = positive threshold and –t =negative threshold). Any hyper plane can be written as the set of points *x* satisfying maximum-margin hyper plane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors. In simpler way we are classifying the nodes have positive threshold and negative threshold. Positive threshold means which satisfies the packet dropped limit, and negative threshold means which is under the packet dropped limit.

Architecture of Proposed System:
Different components of system are explained as below.
Watchdog:
The Local Watchdog has two functions: [1] the detection of selfish nodes and the detection of new contacts. The local watchdog can generate the following events about neighbor nodes: Positive event when the watchdog detects a selfish node, Negative event when the watchdog detects that a node is not selfish, and No detection event when the watchdog does not have enough information about a node (for example if the contact time is very low or it does not overhear enough messages). The detection of new contacts is based on neighborhood packet overhearing; thus, when the watchdog overhears packets from a new node it is assumed to be a new contact, and so it generates an event to the network information module. Furthermore, the watchdog can generate false positives and false negatives. A false positive is when the watchdog generates a positive detection for a node that is not a selfish node. A false negative is generated when a selfish node is marked as a negative detection.
Diffusion:
The Diffusion module has two functions: the transmission as well as the reception of positive detections. A key issue is the diffusion of information. [1] When the diffusion module receives a new contact event from the watchdog, it transmits a message including this information to the new neighbor node. When the neighbor node receives a message, it generates an event to the network information module with the list of these positive (and negative) detections. As the number of selfish nodes is low compared to the total number of nodes, positive detections can always be transmitted with a low overhead. However, transmitting only positive detections has a serious drawback: false positives can be spread over the network very fast. So the information to be diffused should be overviewed through classifier. As per the decision made by classifier same information is propagated through the network.

Information Update:

Updating or consolidating the information is an important activity. This is the function of the Information Update module. A node can have the following internal information about other nodes: No Info state, Positive state and Negative state. A No Info state means that it has no information about a node; a Positive state means it believes that a node is selfish, and a Negative state means it believes that a node is not selfish. A node can have direct information (from the local watchdog) and indirect information (from neighbor nodes). Proposed model is event driven, so the state of a node is updated when the Positive or Negative events are received from the local watchdog and diffusion modules. [1]

Classifier:

The classifier module is playing important role in overall architecture. By using the packet counter values of current state and historical state it makes two classes of nodes as: Fully Selfish and Partial Selfish. Certain threshold value is set to make these to classes. An honest node is not passed as an input to the classifier. If a certain threshold is exceeded by a node then it is directly marked as fully selfish node. If a node is reaching a nearer value of threshold then instead of marking it to the fully selfish class, we are checking the historical count and current count of that node. If this node has behaved in a same way i.e. packet dropping nature then it's obvious that in current situation also it will behave in a same manner. But history shows the good packet forwarding behavior for that node then we are marking that node in partial selfish class. We believe on probability that after some time node will continue to route packets and will fall down well below to the threshold value. Support vector machine can be used for this class making task. In machine learning, support vector machines (SVMs, or support vector networks) are supervised learning models with related learning algorithms that examine data and distinguish patterns, intended for categorization and deterioration analysis. The essential SVM takes a set of input data and predicts, for every given input, which of two feasible classes forms the input. SVM used to classify the node into two groups normal node and malicious node. Nevertheless, this can produce a delay on the detection (more events are needed to get a better decision). Second, the decision about a selfish node is taken using the most recent information.

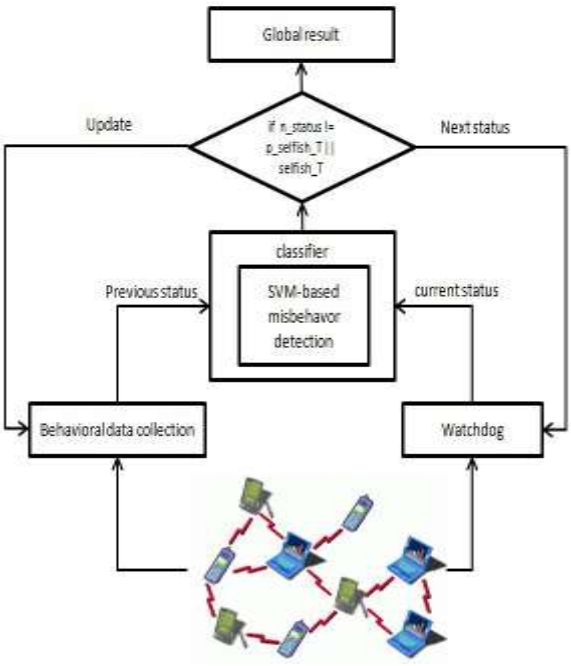

Fig -2: Working of Classifier.

Algorithm for Classifier:

```
//To check current status:
cur_stat = dropP[count]/recP[count];
cur_stat = cur_stat*100;
```

```
//To check previous status:
 hist_stat= historydrop[count]/historyrec[count];
 hist_stat = hist_stat*100;
//Classification Rules:
if(hist_stat<Selfish_Thr &&hist_stat>=PSelfish_Thr)
{    saveIsPartialSelfish(Node);
 }
if(hist_stat<Selfish_Thr && hist_stat>=PSelfish_Thr && isDiffusionMsg==true)
{    saveIsSelfish(Node);
 }
 if(hist_stat>=Selfish_Thr)
{
   saveIsSelfish(Node);
 }
```

SIMULATION AND RESULTS

Following parameters are considered for the simulation of system.

Parameters	Value
Number of nodes	20
Routing Protocol	AODV
Packet Size	256,512,768,1024 bytes
Traffic model of sources	Constant bit rate
Simulation time	100,200,300,400 sec

Table 1: Simulation Parameters

System is developed using Java and MySQL. At first we have to select the source and destination along with simulation time, packet size. Selfish nodes will be randomly selected between the paths.

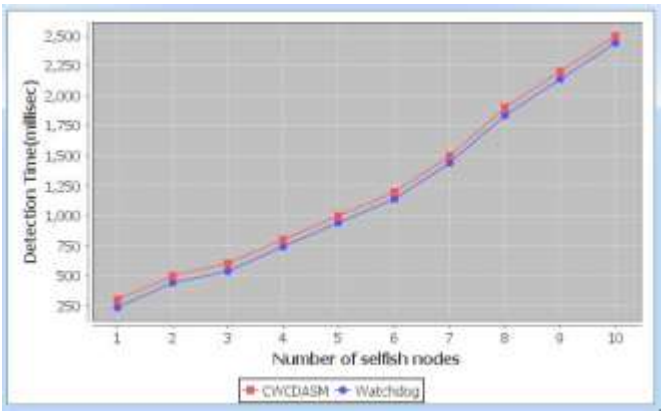

Fig -3: Detection Time

When evaluated, it shows detection time slightly higher than the existing system. Figure 3 shows this detection time comparison. This is justified as we are using some time for classification and diffusion of that result. As system is event based so detection time is slight higher.

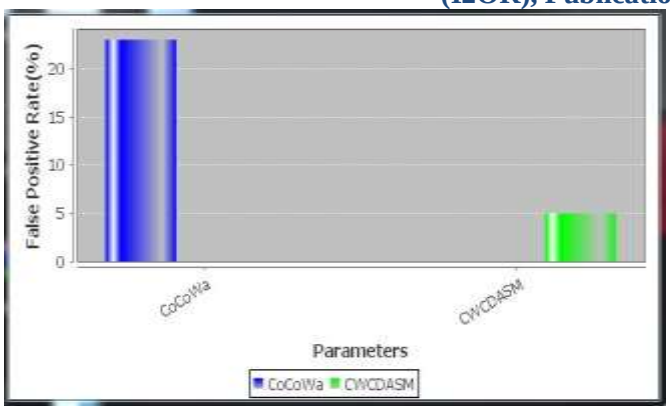

Fig -4: False Positive Rate

But at the same time rate of false positive is reasonably decreased than the existing system. Figure 4 shows this fact. False positives are when the watchdog generates a positive detection for a node that is not a selfish node.

CONCLUSION AND FUTURE WORK

In this paper, we addressed the problem of identifying and isolating selfish nodes that refuse to forward packets in wireless ad hoc network. The impact of such nodes has been shown to be detrimental to network performance, lowering the network throughput and dramatically increasing the end-to-end delay. To mitigate the problem of malicious packet dropping, we developed a comprehensive selfish node detection and suppression system using three major modules; watchdog, classifier and diffusion module. All three functions are coordinated in a distributed manner without the need for centralized control. We analyzed the false positive rate of the implemented system and previous system and found difference between them. The decision about a selfish node is taken using the most recent information. Nevertheless, this can produce a delay on the detection (more events are needed to get a better decision). Finally the implemented system detects and isolates the selfish node and provides available alternative path to continue the packet forwarding thus keeping network stable and increasing the network performance. One problem that remains with Collaborative Watchdog is that all the thresholds need to be set manually in order to get good detection results. So in the future we will try to find ways how these values can be set and adjusted automatically during operation. This work can be extended to detect other types of selfish nodes which can help in improving performance of MANET.

ACKNOWLEDGEMENT

I would like to thank my guide Dr. B. D. Phulpagar for his timely support. Without his guidance it would have not been possible to complete this work. I would also like to extend my sincere thanks to Prof. Deipali Gore and Dr. S. A. Itkar for continuous motivation to present this work.

REFERENCES

1. Enrique Hernandez-Orallo et al. “CoCoWa: A Collaborative Contact-Based Watchdog for Detecting Selfish Nodes”, IEEE transactions on Mobile Computing, vol. 14, no. 6, June 2015.
2. S. Buchegger et al., “Self-policing mobile ad hoc networks by reputation systems”. Communications Magazine, IEEE, 43(7):101 – 107, July 2005.
3. S. Bansal and M. Baker, “Observation-based cooperation enforcement in ad hoc networks” Stanford University, Tech. Rep., 2003.
4. Khairu lAzmi Abu Bakar and James Irvine “Contribution Time-based Selfish Nodes Detection Scheme” ISBN: 978-1-902560-24-3 © 2010 PGNet.
5. M. Hollick, J. Schmitt, C. Seipl, and R. Steinmetz”, On the effect of node misbehavior in ad hoc networks. In Proceedings of IEEE International Conference on Communications, ICC’04, pages 3759–3763. IEEE, 2004.
6. C. K. N. Shailender Gupta and C. Singla, “Impact of selfish node concentration in MANETs”, International Journal of Wireless and Mobile Networks (IJWMN), 3(2):29–37, Apr 2011.

7. C. Toh, D. Kim, S. Oh, and H. Yoo, "The controversy of selfish nodes in ad hoc networks", In Proceedings of Advanced Communication Technology (ICACT), volume 2, pages 1087 –1092, Feb. 2010.
8. Y. Yoo, S. Ahn, and D. Agrawal, "A credit-payment scheme for packet forwarding fairness in mobile ad hoc networks", In Proceedings of IEEEICC, volume 5, pages 3005 – 3009 Vol. 5, may 2005.
9. S. Marti, T. Giuli, K. Lai, and M. Bakar, "Mitigating routing misbehavior in mobile ad hoc networks," in Proc. 6th Annual Int. Conf. on Mobile Computing and Networking (MobiCom'00), August 2000, pp. 255–265.
10. K. Liu, J. Deng, P. K. Varshney, and K. Balakrishnan, "An acknowledgment-based approach for the detection of routing misbehavior in manets," in IEEE Transactions on Mobile Computing, 2006, pp. 536–550.
11. Q. He, D. Wu, and P. Khosla, "Sori: A secure and objective reputationbased incentive scheme for ad-hoc networks," in WCNC 2004, 2004.
12. S. Buchegger and J. L. Boudec, "Performance analysis of the confidant protocol: (cooperative of nodes – fairness in dynamic ad hoc networks)," in Proc. IEEE/ACM Workshop on (MobiHoc'02), June 2002, pp. 226–336.
13. P. Michiardi and R. Molva, "Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in (CMS'02), September 2002.
14. K Balakrishnan, J Deng, and P K Varshney, "TWOACK: Preventing Selfishness in Mobile Ad Hoc Networks", Proc. IEEE Wireless Comm. And Networking, pp. 2137- 2142, 2005
15. S. Zhong, J. Chen, and Y. Yang, "Sprite: A Simple, Cheat-Proof, Credit- Based System for Mobile Ad-Hoc Networks", Technical Report, Yale University, July 2002, pp. 1987-1997.
16. Wu, Lien-Wen, and Rui-Feng Yu, "A threshold-based method for sel_sh nodes detection in MANET." Computer Symposium (ICS), 2010 International IEEE, 2010.