A 2ACK with PFC Scheme to Detect Misbehavior Nodes in MANET

K V Chaitra
M.Tech, CNE
AMC Engineering College
Bangalore,India

J Selvin Paul Peter
Associate Professor, ISE Dept
AMC Engineering College
Bangalore,India

***Abstract*— Mobile Ad hoc networks (MANETs) are defenseless to having their powerful operation compromised by a variety of security attacks in view of the highlights like unreliability of remote connections between nodes, continually evolving topology, limited battery power, absence of centralized control and other. Nodes may misbehave either on the grounds that they are malicious and deliberately wish to disturb the network, or since they are selfish and wish to preserve their own scarce resources, for example, power. In this paper, a mechanism is been introduced that empowers the detection of nodes that exhibit packet forwarding misbehavior. The methodology is in view of the use of two techniques which will be utilized in such a way that the results produced by one of them are further transformed by the other to finally generate the list of misbehaving nodes. The first part identifies the misbehaving links using 2ACK technique and this information is fed into second part which utilizes the principle of conservation of flow (PFC) technique to recognize the misbehaving nodes. The issue with the 2ACK algorithm is that it can recognize the misbehaving link but cannot decide upon which one of the nodes connected with that link are misbehaving. Subsequently we utilize the principle of conservation of flow, PFC for the second part which recognizes the misbehaving nodes associated with that of the misbehaving link.**

Key Words—Mobile- ad-hoc network (MANET), 2ACK, principle of conservation flow, misbehaving link

I. INTRODUCTION

Mobile Ad hoc network (MANET) is an accumulation of digital data terminals outfitted with wireless transreceivers which can communicate with one another without the need of any fixed infrastructure. As it were, not at all like the infrastructure based mobile networks like the cellular networks which oblige a base station for the transmission of data packets, the manets are infrastructure less networks wherein the nodes can straightforwardly communicate with one another without the need of any routers. Every node consolidates the functionality of the router and can transmit packets specifically to those nodes which are inside its transmission range, furthermore advances packets on the behalf of any other node which wishes to transmit the data packets to some other node which is not inside its transmission range. In other words, to achieve the data communication between those nodes which are not inside the transmission scope of each other, data packets are handed-off over a sequence of intermediate nodes using the "store and forward multi-hop" transmission rule. The wireless nature and inherent highlights of mobile ad hoc networks makes them defenseless against a wide assortment of attacks by misbehaving nodes. Such attacks range from passive eavesdropping where a node tries to acquire unapproved access to data bound for another node, to active interference where malicious nodes obstruct network performance by not obeying universally satisfactory guidelines

Two sorts of MANETs exist: Closed and Open. In a closed MANET, all the nodes work in collaboration to fulfill a typical objective like emergency search or rescue operation . In an open MANET, all the nodes impart their resources so as to accomplish their individual objectives and guarantee global connectivity. In other words , as the nodes partake in network activities, the battery power continues decreasing. Henceforth there may be a circumstance where certain nodes want to benefit by utilizing the resources of other nodes which forward their data packets however they decline to give the service to other nodes . Such nodes basically drop the packets as opposed to sending them to their next neighbor. These nodes are said to be selfish or misbehaving nodes .

With a specific end goal to relieve the impacts of routing misbehavior, several methodologies were proposed [3], [4], [5] and [6]. In [3], a methodology was proposed which included two components watchdog and pathrater. The watchdog overhears the remote medium to check whether the following node in the path advances the packets to its next neighbor on the route. In the event that it does not forward the packets, it is considered as misbehaving node. Thusly, watchdog recognizes misbehaving nodes by indiscriminately listening to the next node's transmission. The pathrater uses the information gained by the watchdog to dispense with the misbehaving nodes from the routing path. Anyhow the disadvantage of this methodology is that, watchdog may not recognize a misbehaving node in the vicinity of ambiguous collisions, receiver collisions or nodes equipped for controlling their transmission power. Such shortcomings are the consequence of utilizing unbridled listening to figure out if a node has forwarded packet or not.

In this paper a scheme is been proposed which productively identifies the misbehaving nodes so that they may be discarded from the network. It is in based on the utilization of two techniques: 2ACK scheme and PFC (Principle of Flow of Conservation) scheme. The 2ACK technique is utilized to identify the misbehaving links through two-hop acknowledgment packets called 2ACK packets which are alloted a fixed d two-hop route in the opposite direction of flow of data packets. At whatever point a node advances a data packet, it expects to get a 2ACK packet from the destination of the next hop link. Taking into account the number of packets which missed the 2ACK packets within e a certain time limit, a choice about the misbehaving link is taken. This information of the misbehaving link is used by PFC technique which decides the misbehaving node(s) connected with the misbehaving link. The choice is based on the analysis of inflow and the outflow of the data packets associated with the nodes. As pointed out in [8], the 2ACK procedure determines the misbehaving link and it must be figured out which one of the two nodes associated with this link are misbehaving. Our proposed scheme meets expectations at deciding the misbehaving node through the knowledge of misbehaving link by utilizing the 2ACK procedure took after by the PFC procedure.

II. RELATED WORK

A number of approaches have been proposed to detect and overcome the problem of node misbehavior. In [1], Miranda and Rodrigues adopted a methodology where every node i keeps up information about every other node j in the manifestation of a data structure represented as Statusi[j] which shows node i's impression about node j. It additionally looks after a credit counter where credits are earned for packet sending activity. Aside from this, every node i additionally keeps up two rundowns for every node j which contains those nodes to which node j will furthermore, won't give service. This data is intermittently broadcasted in the form of self-state message which is utilized by different nodes to upgrade their own rundowns.

Marti et al. [3] proposed a scheme for misbehavior detection which includes two components specifically Watchdog and Pathrater. The Watchdog module overhears the medium to ensure that the next-hop node steadfastly advances the packet to it's next neighbor. It keeps up a buffer of all recently sent packets and holds up to overhear the next-hop node forwarding the packet within a certain time limit. Under those circumstances, the packet is cleared from the buffer. Otherwise, the next-hop node is accused as misbehaving. The Pathrater uses the allegations of Watchdog to rate each of the paths maintained in it's cache. Based on this rating the path which best avoids misbehaving nodes is chosen. This scheme requires that nodes operate in promiscuous mode due to which detection o misbehavior may fail or false alarms may be raised in th presence of ambiguous collisions, receiver collisions and limited transmission power

Buchegger and Le Boudec [5] proposed a protocol called CONFIDANT. It is based on selective altruism and utilitarianism. It includes four vital components specifically: Monitor, Reputation System, Path Manager, and Trust Manager. The Monitor of every node keeps track of the forwarding behavior of next-hop node by overhearing the medium. Any suspicious behavior causes the warning to be passed on to the Reputation System which takes into account the significance and the frequency of the event based on which the rating of the suspicious node is decreased. When the rating reaches below a threshold, control is passed on to the Path Manager which accordingly modifies the route cache and the Trust Manager propagates the Alarm message to all the nodes. The criteria for choosing the threshold which decides the misbehavior is difficult. Since the scheme is based on indiscriminate overhearing by the nodes, it experiences the same downsides as the Watchdog and Pathrater system.

III. PROPOSED SYSTEM

The proposed scheme works in two phases which take place in parallel: In the first phase, 2ACK technique [8] is used to carried out on all misbehaving links found on the route formed by any basic routing algorithm like Dynamic Source Routing (DSR). All such links are blacklisted and in the second phase, each node scans the blacklist periodically to check if any of it's neighbors are associated with blacklisted links. For each such neighbor, the PFC technique is applied to determine whether it is a misbehaving node

A. 2ACK Technique

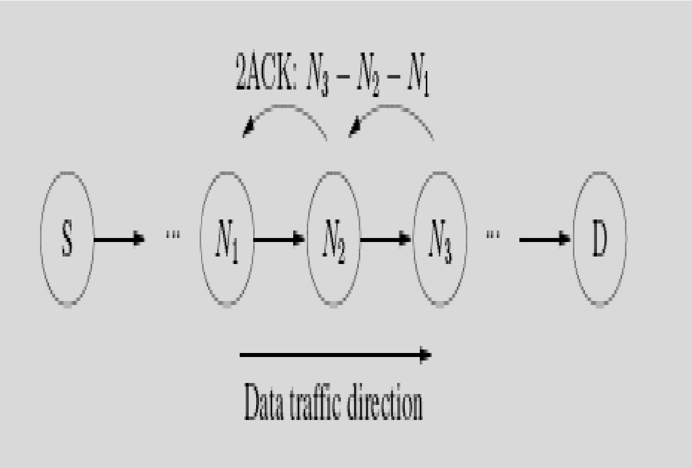


Fig.1 2ACK Scheme

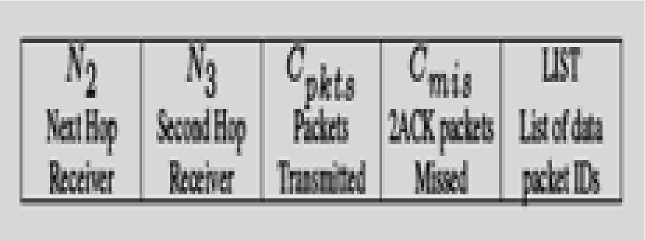
The 2ACK strategy is a network- layer technique which is used to detect misbehaving links so that they may be avoided from the routing paths in future. It can be utilized as an add-on to the current routing protocols like DSR. A special type of two hop acknowledgement packet called as 2ACK packet is used for misbehavior detection. A 2ACK packet is assigned a fixed route of two hops (three nodes) in the opposite direction of the data traffic route. Figure 1 illustrates the operation of the 2ACK scheme.

At whatever point a route must be shaped from source S to the destination D, we first utilize the essential routing protocol like DSR. To apply the 2ACK method, we envision the whole route as set of consecutive overlapping triplets: for instance if S-N1-N2-N3-N4-N5-D represents a route from Source to destination then the 2ACK method is applied to each triplet of the set:

{(S, N1, N2) (N1, N2, N3) (N2, N3, N4) (N3, N4, N5) (N4, N5, D)}

Let N1, N2, and N3 be the three consecutive nodes (triplet) along a route and the route begins from the source hub S, to the closes with end node D. As node N1 sends data packet to N2, and N2 advances the same to N3 etc, all the while, if the information transmission is not affirmed by the node N1, then such an act may be termed as ambiguity that exists without misbehaving nodes. These issues are serious in an open MANETs with the presence of potential misbehaving nodes.

In the proposed 2ACK scheme an unequivocal acknowledgment is created and will be sent by the destination node N3. As node N3 gets the data packet effectively, it conveys a 2ACK packet over two hops to N1 alongside the ID of the corresponding data packet . Accordingly the triplet [N1->N2->N3] is derived from the route of the original data traffic. Such a triplet is utilized by N1 to monitor the link N2->N3. The 2ACK transmission happens for each set of triplets along the route. Subsequently, not just the first router from the source serve as a 2ACK packet sender., but also other nodes in route, before the destination expect the destination node.



N_2 Next Hop Receiver	N_3 Second Hop Receiver	C_{pkts} Packets Transmitted	C_{mis} 2ACK packets Missed	LIST List of data packet IDs

Fig. 2. Data structure maintained by the observing node

The 2ACK receiver monitors the link periodically by keeping the data about the number of data packets sent and the number of data packets which have not been acknowledged within the pre-defined time period known as wait time. Each node N1 in the triplet sends a data packet to N2 and increments the counter Cpkts. It additionally includes the ID of node N2 to the list of nodes from whom the 2ACK packets are anticipated that would arrive. This list is kept up as a data structure known as LIST(see Fig. 2). It then begins waiting for the 2ACK packet from N2. In the event that it is received within the wait-time, then the ID of N2 is expelled from LIST. Otherwise another counter called Cmiss is incremented. Cmiss represents the number of 2ACK packets which were not received within wait time. Periodically, say for every T sec, N1 performs the monitoring of link behavior N2-N3. The proportion Cmiss / Cpkts is compared with a threshold Rmiss. If Cmiss / Cpkts > Rmiss then it means that the number of missed 2ACK packets crosses the threshold which indicates that the link N2-N3 is misbehaving and it is blacklisted

B. *PFC Technique (Principle of Conservation of flow)*

2ACK algorithm can recognize the misbehaving link but cannot decide upon which one of the nodes connected with that link are misbehaving. Subsequently we utilize the principle of conservation of flow, PFC Once a link is blacklisted by the 2ACK procedure , the PFC procedure takes both nodes associated with that link as input and discovers behavior of both nodes independently. The misbehaving nodes are all blacklisted so that such nodes can be penalized by not including it in any sort of network activity. According to the PFC technique in a static network model, a direct relation exists between the rate of inflow traffic and the rate of outflow traffic associated with a node. More specifically, the number of packets received by a node xj from other neighboring nodes for forwarding and the number of packets sent by xj to other neighboring nodes for forwarding should be equal.

The principle of flow of conservation over an ideal static network model can be stated as follows

$$\sum_{\forall i \mid x_i \in Y_j} S_{ij}\ (t_1) \;=\; \sum_{\forall i \mid x_i \in Y_j} R_{ij}\ (t_1)$$

where the following notations are used

- xj be a node such that xj ∈ *X*, where *X* = {x1, x2, x3 …xN} is the set of all nodes in the network, *N* is the total number of nodes in the network, and *j= 1, 2, 3 … N.*
- Let Yj be the subset of nodes in the network which are neighbors of , i.e. Yj is the neighborhood of xj. It follows that xj ∉ Yj and also Yj ⊂ X.
- Let Δ*t* be the period of time elapsed between two points in time *t0* and *t1* such that Δ*t = t1 – t0.*
- Let Rij be the number of packets that node xi has successfully sent to node xj for xj to forward to a further node; xi ∈ Yj, xj ∈ Yj, *i ≠ j* and Rij *(t0) = 0.*
- Let Sij be the number of packets that node xi has successfully received from node xj that did not originate at xj; xi ∈ Yj, xj ∈ Yj, *i ≠ j* and Sij *(t0) = 0.*

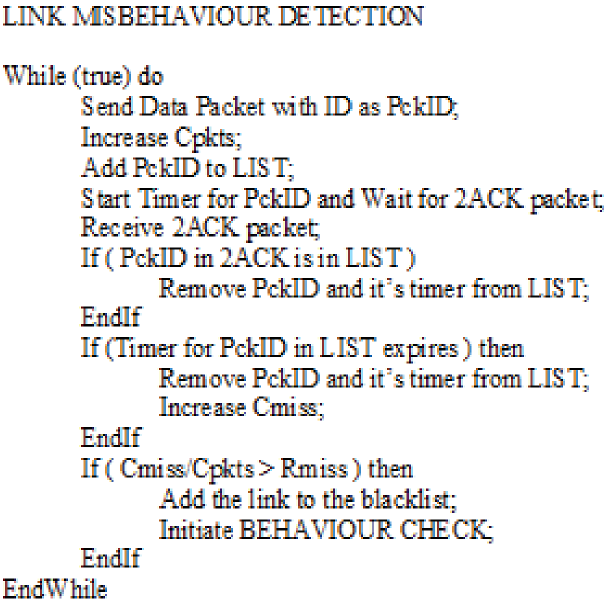
The above equation holds good over an ideal static environment in which no collisions occur during the time period Δt. But in a MANET's environment, the ideal condition does not exist. First the wireless medium is error prone and packets get lost during transmission. Secondly nodes compete to use the medium, hence there is a possibility of collisions. Nodes may exhibit malicious behavior unintentionally, especially in a MANET because of several reasons like the unavailability of resources like CPU cycles, buffer space and bandwidth when the packet has to be forwarded. Hence a threshold has to be setup in order to accommodate unintentional misbehavior by a node which may result in packet dropping. Mathematically, it can be represented as follows which is the modification to the equation given above.

$$(1-\gamma_{threshold}) \sum_{\forall i \mid x_i \in Y_i} S_{ij}\ (t_1) \;\leq\; \sum_{\forall i \mid x_i \in Y_i} R_{ij}\ (t_1)$$

where the threshold factor *γthreshold* can take values between 0 and 1.

IV. PROPOSED ALGORITHM

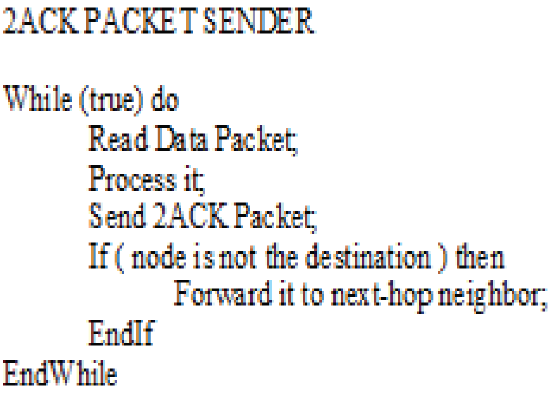
Every node keeps running the 2ACK algorithm whenever a route must be created from a source node S to a destination node D. The 2ACK strategy includes the logical formation of overlapping triplets upon the routing path from source S to destination D. The module LINK MISBEHAVIOUR DETECTION (see Fig.3) is executed by a node which is logically the first one in a triplet along the route. It forwards / sends the data packet and applies the idea of 2ACK technique to determine the misbehaving link. The module 2ACK PACKET SENDER (see Fig. 4) is executed by a node which is logically the last one in the triplet along a route. It receives the data packet and as per the 2ACK technique, if it is well behaving then, it is supposed to send 2ACK packet over two hops in the reverse direction to that node which is the first one in the triplet.



```
LINK MISBEHAVIOUR DETECTION

While (true) do
        Send Data Packet with ID as PckID;
        Increase Cpkts;
        Add PckID to LIST;
        Start Timer for PckID and Wait for 2ACK packet;
        Receive 2ACK packet;
        If ( PckID in 2ACK is in LIST )
                Remove PckID and it's timer from LIST;
        EndIf
        If (Timer for PckID in LIST expires ) then
                Remove PckID and it's timer from LIST;
                Increase Cmiss;
        EndIf
        If ( Cmiss/Cpkts > Rmiss ) then
                Add the link to the blacklist;
                Initiate BEHAVIOUR CHECK;
        EndIf
EndWhile
```

Fig. 3. Link Misbehavior Detection Algorithm



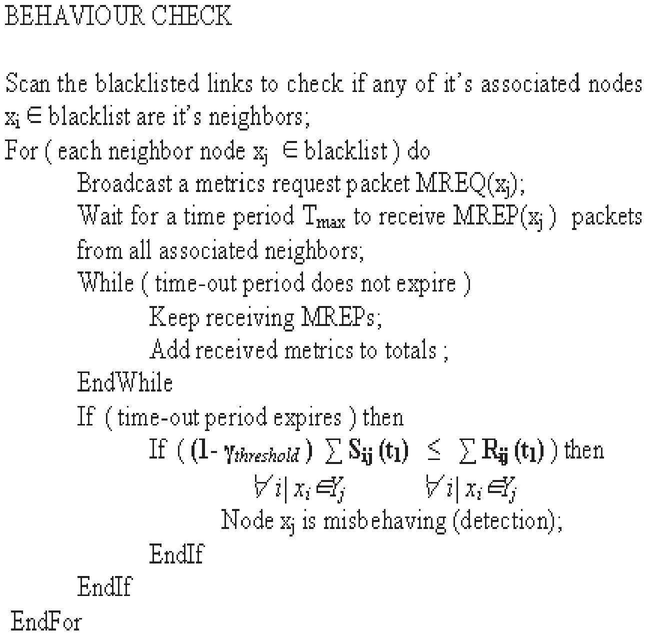
```
2ACK PACKET SENDER

While (true) do
        Read Data Packet;
        Process it;
        Send 2ACK Packet;
        If ( node is not the destination ) then
                Forward it to next-hop neighbor;
        EndIf
EndWhile
```

Fig.4. 2ACK Packet Sender Algorithm

Once a link is blacklisted, each of the nodes checks to see if any of their neighbors are associated with this link. The module BEHAVIOUR CHECK (see Fig. 5) performs this task. It gathers the metrics associated with the neighbor node by broadcasting MREQ packets which stand for metrics request packet. It then sets a timer and starts waiting for MREP (metrics reply) packets from each of the associated neighboring nodes which are all accumulated. Once all MREP packets are received, it checks to see if PFC condition is satisfied to arrive at a conclusion of whether the node being checked is well behaving or misbehaving. The sending of MREQ packets by BEHAVIOUR CHECK module invokes another module called as METRICS REQUEST HANDLING.

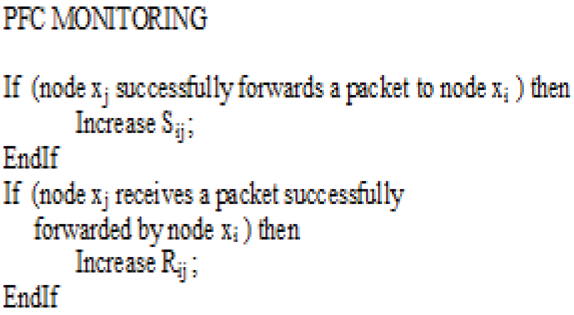
The module PFC monitoring keeps running in background on each node to gather inflow and outflow metrics associated with each neighboring node. These metrics are stored along with their time stamp and will be used by a module known as METRICS REQUEST HANDLING which provides the information to BEHAVIOUR CHECK module in the form of MREP packets.



```
BEHAVIOUR CHECK

Scan the blacklisted links to check if any of it's associated nodes
$x_i \in$ blacklist are it's neighbors;
For ( each neighbor node $x_j$ $\in$ blacklist ) do
        Broadcast a metrics request packet MREQ($x_j$);
        Wait for a time period $T_{max}$ to receive MREP($x_j$ ) packets
        from all associated neighbors;
        While ( time-out period does not expire )
                Keep receiving MREPs;
                Add received metrics to totals ;
        EndWhile
        If ( time-out period expires ) then
                If ( $(1- \gamma_{threshold}) \sum_{\forall i | x_i \in Y_j} S_{ij}(t_1) \leq \sum_{\forall i | x_i \in Y_j} R_{ij}(t_1)$ ) then
                        Node $x_j$ is misbehaving (detection);
                EndIf
        EndIf
EndFor
```

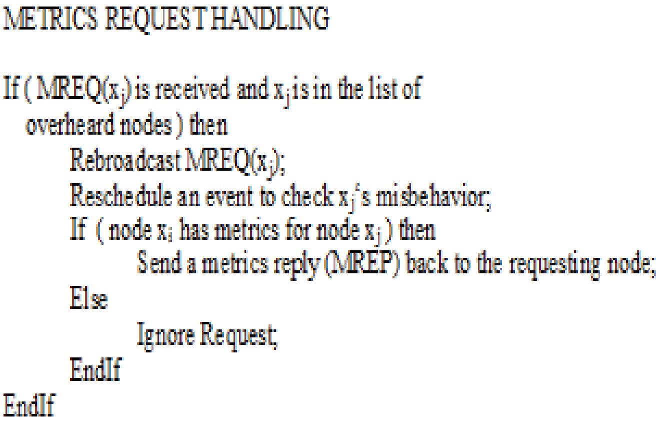
Fig.5. Behavior Check Algorithm



```
PFC MONITORING

If (node $x_j$ successfully forwards a packet to node $x_i$ ) then
        Increase $S_{ij}$;
EndIf
If (node $x_j$ receives a packet successfully
    forwarded by node $x_i$ ) then
        Increase $R_{ij}$ ;
EndIf
```

Fig. 6. PFC Monitoring Algorithm



```
METRICS REQUEST HANDLING

If ( MREQ($x_j$) is received and $x_j$ is in the list of
    overheard nodes ) then
        Rebroadcast MREQ($x_j$);
        Reschedule an event to check $x_j$'s misbehavior;
        If ( node $x_i$ has metrics for node $x_j$ ) then
                Send a metrics reply (MREP) back to the requesting node;
        Else
                Ignore Request;
        EndIf
EndIf
```

Fig.7. MREQ Handling Algorithm

V. CONCLUSION

The proposed scheme proficiently performs the identification of misbehaving node by the combination of 2ACK and the PFC techniques. Since the 2ACK strategy detects the misbehaving link but cannot decide which one of the two associated nodes are misbehaving, we augmented the technique by applying PFC monitoring as the next step to detect the misbehaving nodes once the misbehaving link is detected. The computational overhead as contrasted to the original PFC technique is reduced significantly since we are examining only those nodes behavior which are associated with misbehaving links.

REFERENCES

[1] H. Miranda and L. Rodrigues, "Preventing Selfishness in Open Mobile Ad Hoc Networks," Proc. Seventh CaberNet Radicals Workshop, 2002.

[2] L. Buttyan and J.-P. Hubaux, "Security and Cooperation in Wireless Networks," http://secowinet.epfl.ch/, 2006.

[3] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehaviour in Mobile Ad Hoc Networks," Proc. MobiCom, Aug. 2000.

[4] L. Buttyan and J.-P. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks," ACM/Kluwer Mobile Networks and Applications, vol. 8, no. 5, 2003.

[5] S. Buchegger and J.-Y. Le Boudec, "Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes, Fairness in Dynamic Ad-Hoc Networks," Proc. MobiHoc, June 2002.

[6] S. Zhong, J. Chen, and Y.R. Yang, "Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc Networks," Proc. INFOCOM, Mar.-Apr. 2003.

[7] D. Johnson, D. Maltz, Y.C. Hu, and J. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," Internet draft, Feb. 2002.

[8] Kejun Liu, Jing Deng, Pramod K. Varshney, Kashyap Balakrishnan "An acknowledgement-Based Approach for the Detection of Routing Misbehaviour in MANETs", IEEE Transactions on Mobile Computing, vol. 6, No. 5, 2007.

[9] Gonzalez et al.: Detection and Accusation of packet forwarding misbehavior in mobile ad-hoc networks, Journal of Internet Engineering, vol. 2, pp.1, 2008.