Journal of Computer Science 7 (5): 723-730, 2011
ISSN 1549-3636

Misbehaving Node Detection in Mobile Ad Hoc Networks using Multi Hop Acknowledgement Scheme

[1]Usha Sakthivel and [2]S. Radha
[1]Sathyabama University, Chennai, Tamil Nadu, India
[2]S.S.N. College of Engineering Chennai, Tamil Nadu, India

Abstract: Problem statement: For any node in a Mobile Ad hoc Network conservation of battery power and bandwidth are the priority. Hence, they try to reduce the overload they would otherwise incur when they forward packets. This selfish behavior of a node affects the throughput of the network. The nodes may also choose a back off value of shorter duration. These problems are handled effectively by the methodology proposed in this study. A conscious effort has been made keeping the constraints of the MANETs in mind. **Approach:** Misbehavior is best identified at the lower levels as the upper levels of the OSI standard primarily deals with the data the packets carry and less about how, so the network and the MAC layers is where the primary focus lies. Thus, keeping these points in mind, we propose algorithms that work along with the 802.11 MAC protocol to monitor the behavior of neighboring nodes by listening to the channel, specifically monitoring parameters like back off values sent by the nodes. A counter is maintained which is incremented every time node misconduct is detected, subsequently after a particular value is cross the node is labeled as misbehaving and the information is broadcast over the network. **Results:** Performance parameters like throughput, packet delivery ratio were monitored with traffic of the magnitude 10 to 60 nodes. Also the performance of the network based on the percentage of selfish nodes present in the network was monitored and a graph was generated based on the statistics. **Conclusion:** An algorithmic approach for misbehaving node detection and isolation in ad hoc networks by modifying the protocol being used in the lower layers which consequently improves performance of the network had been proposed. Simulation results show considerable performance increase upon implementing the proposed algorithm. Further research can confirm the practicality of the proposed idea.

Key Words: Misbehaving node, MANETs, MAC, collective network arbitration protocol, packet delivery ratio, throughput

INTRODUCTION

The problem of misbehavior could occur in the network layer (Marti *et al*., 2000) and the MAC layer. The tendency of a node to deviate from the accepted norm is classified into two categories, selfish and malignant. The former being a node which deems it not necessary to forward those packets which are not destined to itself, chiefly because of the greed on the part of the node to conserve battery power (Natsheh and Buragga, 2010). The second kind of misbehaving node is the one with the explicit aim to bluff the neighbors into thinking that it is behaving properly by even wasting some resources while actually misleading them.

The mobile nodes in a MANET in courtesy to the other nodes in the network and to keep up the fairness of distribution in the network 'channel' are expected to wait for a pre specified period of time between successive (Kaabneh *et al*., 2009) transmissions. The MAC layer of the network is where the medium contention resolution mechanisms are implemented. The 802.11 IEEE specification mandates that each node should choose random back off values within a certain range and should be idle for the amount of time equal to the back off value before starting a new transmission.

As one might expect the MANET is a self made network without any arbitrator to chastise nodes which fails to follow the protocols. A node might choose nonrandom and the back off value in order to transmit more frequently. This will on one hand enable that node to more effectively, utilize the channel and improve its throughput. On the down side it divests the other nodes of their rightful access to the channel. This plays a huge role in defeating the goals of fairness in a

Corresponding Author: Ms. Usha Sakthivel, Sathyabama University, Chennai, Tamil Nadu, India

network. All the nodes in the network share the network bandwidth and thus a single node selfishly increasing its bandwidth allocation (Hu *et al*., 2003) should be stopped.

This study begins by discussing briefly some of the existing solutions to routing and MAC (Buttyan and Hubaux, 2003) misbehavior detection followed by a novel proposal to comprehensively reduce routing and MAC misbehavior.

Related work: PPM (Marti *et al*., 2000) is one of the credit based methods in which the forwarding of packets by intermediate nodes is encouraged by providing some resources other than physical, the presence of which is made indispensable for sending packets in the future. In the packet purse model the originator is charged for the message it wishes to send. The charges are handled in currencies called nuggets.

In the PTM model each node has to buy the packets for a certain no of nuggets and can sell it to the next node for some amount of nuggets. This ensures that the packet purse which contains the nuggets need not be carried all along the path. Also, because of this scheme the source does not need to know the total amount of nuggets required in advance. This also means that it is not necessary for the source to bear the entire cost of forwarding but the destination has to. Since the destination pays for the packet forwarding service, there is a scope for multicast packet transfer mode with this model.

In the SPRITE model (Zhong *et al*., 2003) here is a centrally located credit clearance system. A group of nodes which has access to the network interface via a wireless overlay are considered. Each node should possess a certificate provided by an authorized central authority. The SPRITE works above the DSR protocol. In general a node will gain more credit if it forwards a packet for some other node. The same node would lose a part of credit if its own message is to be forwarded.

The WATCHDOG (Kaabneh *et al*., 2009) is used to detect misbehaving nodes. The Watchdog method uses the passive method of over hearing the links of the next node to see whether they have forwarded the packet. This is because each node can listen to all the links of there is no link encryption, the nodes can even check for the integrity of the messages.

PATHRATER model proposes to use link data as well as misbehaving node data to select a path. Each node maintains a metric for every node that it knows. And each node also maintains a metric for each path it knows. The path metric is calculated as an average of the metrics of the individual nodes in the path. So if a node finds that there are various paths that could be used to reach the destination, it chooses the one with the highest metric.

The 2-ACK schemes a network layer (Balakrishnan *et al*., 2005) technique to detect misbehaving links. It is implemented over DSR. It is used as an add-on over the DSR. It defines a packet (2-ACK packet), which has a fixed route of two hops in the direction opposite to the original packet flow.

The idea to detect the MAC misbehavior has largely been, paper (Natsheh and Buragga, 2010) and very little has been achieved in actual practice. The DOMINO (Raya *et al*., 2004) was perhaps first comprehensive idea whichdealt with detection in infrastructure based networks by installing software in the Access points. But this idea comes short in the method in which misbehavior detection is performed. The solution achieved is only suboptimal. Further its applicability in ad hoc networks requires to be tried.

Selfish MAC layer misbehavior (Kyasanur and Vaidya, 2003), where hosts deviate from the specified bakeoff strategy.Konorski proposes a modified back off algorithm using black bursts and with a game theoretic analysis, shows that the protocol is resilient to selfish misbehavior.Konorski's work assumes that all hosts can accurately measure the duration and originator of each black-burst, which is hard to guarantee in a wireless network.

One such innovative methodology (Gunasekaran *et al*., 2008) attempted to mitigate this issue by having the receiver assign the back off values for the sender. But again this assumes that the receiver could be trusted. The receiver might choose to send back off values on the shorter side if it were to benefit by receiving data more frequently. Thus using this as a generic solution is far from reality.

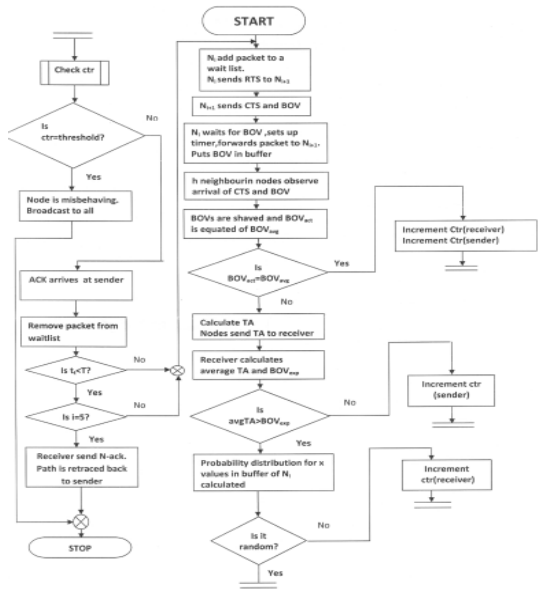
MATERIALS AND METHODS

N-ACK scheme: The Nack scheme extends the 2 Ack scheme in trying to isolate misbehaving nodes in a MANETs. The Nack scheme requires an end to end Ack packet to be sent between the source and the destination. The destination on receipt of the data packets sent by the source, responds with a Nack packet.

Each node maintains a list of data packets sent and another list of data packets forwarded. As soon as a node initiates a data packet as a source, it adds the id of the packet to the list of data packet sent. As the node receives the Nack packet for the data packet it removes the corresponding data packet id from the data packet sent list.

The data packet and the Nack packet keep track of the route they travel. The Nack would try to reach the source from the destination with the help of the path, which is found node is found to be misbehaving in the pre calculated path the intermediate nodes are free to divert the Nack packet through alternative paths. But the new path will be stored in the Nack packet along with the older path, which is extracted from the original message.

On receipt of the Nack packet, the source node compares the two paths that are in the Nack packet. If there is no variation in the paths, then the source node concludes that there are no potential misbehaving nodes in the path. In case the two paths vary, the node in the source to destination path, from where the path varies in the destination to source path is isolated. This node is marked as a potential misbehaving node by the source node. For each potential misbehaving node, a threshold is maintained. If the number of times a node is adjudged as a potential misbehaving node exceeds the threshold, then the node is flagged as misbehaving and information is sent to all the neighboring nodes advising them about the misbehaving node in the actual message packet, delivered to the destination. If a node is found to be misbehaving in the pre calculated path, the intermediate nodes are free to divert the Nack packet through alternative paths. But the new path will be stored in the Nack packet along with the older path, which is extracted from the original message.



N-ACK scheme: Flowchart to monitor the behavior of neighboring nodes and broadcasting the information of misbehaving nodes over the network

On receipt of the Nack packet, the source node compares the two paths that are in the Nack packet. If there is no variation in the paths, then the source node concludes that there are no potential misbehaving nodes in the path. In case the two paths vary, the node in the source to destination path, from where the path varies in the destination to source path is isolated. This node is marked as a potential misbehaving node by the source node. For each potential misbehaving node, a threshold is maintained. If the number of times a node is adjudged as a potential misbehaving node exceeds the threshold, then the node is flagged as misbehaving and information is sent to all the neighboring nodes advising them about the misbehaving node. The process is similar to the protocol followed by a source node to keep track of data packets initiated. Here the intermediate nodes keep track of the forwarded data packets and Nack packets in the forwarded message packets list. The judgment of a neighboring node as potentially misbehaving node is done when an Ack is not received within a pre set time out. As before, the number of times a neighboring is termed as potentially misbehaving node determines whether or not it should be termed as misbehaving nodes in the path.

In case the two paths vary, the node in the source to destination path, from where the path varies in the destination to source path is isolated. This node is marked as a potential misbehaving node by the source node.

For each potential misbehaving node, a threshold is maintained. If the number of times a node is adjudged as a potential misbehaving node (Huang *et al*., 2009; Babakhouya *et al*., 2008) exceeds the threshold, then the node is flagged as misbehaving and information is sent to all the neighboring nodes advising them about the misbehaving node.

Further each node must send back a normal Ack to its immediate source node after receipt of any kind of packet. This would help the intermediate node to judge about its immediate neighboring node and advice the other nodes about the credibility of the neighboring nodes. The process is similar to the protocol followed by a source node to keep track of data packets initiated. Here the intermediate nodes keep track of the forwarded data packets and Nack packets in the forwarded message packets list. The judgment of a neighboring node as potentially misbehaving node is done when an Ack is not received within a pre set time out. As before, the number of times a neighboring is termed as potentially misbehaving node determines whether or not it should be termed as a misbehaving node.

To consider the case in which the Nack packets are lost, the source node will wait for a certain time out period and then re send the original data packets assuming the data packets were lost. If the N ack packet is lost either due to misbehaving nodes or some other reason, the destination would receive the same packet again. This should prompt them about the fact that the Nack it sent has not reached the source. Considering it as the work of misbehaving nodes the destination now should go for an alternating path. If the problem persists in multiple paths the common node in the path could be isolated as the misbehaving node. On the other hand if the data packets are lost in the first case, the destination would receive the data packets for the first time

Collective Network Arbitration Protocol (CNAP): To counter the problem of MAC misbehavior on a holistic scale, one has to consider the sender and receiver behavior as merely roles assumed by each node in course of a data transfer. Thus the discrimination of a node as a sender and a receiver is merely in the context of a data transfer and the same does not extend to the misbehavior detection scheme introduced here. But for the sake of illustrating the effectiveness and the capability of the protocol to handle both sender and receiver misbehavior, we term a node as sender or receiver. Thus the rationale behind the classification of a node as a sender or a receiver is different in this regard.

The CNap proposed in this study is an extension of the specifications made in the IEEE 802.11 protocol. As a pre requisite each node is expected to maintain a set of information about each of its neighboring nodes. The definition of a neighboring is rather obscure in the sense that the nodes are highly mobile and maintaining a static list of adjacent nodes which are in the radio range is extremely difficult. This discussion though is out of the scope of the topic under consideration. Let us just assume that we are able to maintain the information required. Each node maintains a counter (Ctr) for each of the node in its neighborhood list. This counter is initiated to zero and can have a maximum value equal to a threshold (UL) which is predefined.

The transmission process has a precursor step in which two broadcast messages are exchanged between the potential sender and the receiver. The sender begins with the RTS (Request to Send) message and the receiver replies with CTS (Clear to Send). Since this process is transparent and the broadcast messages are visible to the neighboring nodes of the sender and the receiver the neighboring nodes are able to identify the roles assumed by the nodes.

Backoff calculation

SendBOV=Random number*slot time*CW
(Calculated by sender)

RecvBOV=Random Number*slot time*CW
(Calculated by receiver)

Scenario 1: Sender misbehavior: In the first scenario let us assume that the node now acting as the sender is misbehaving while the receiver node is behaving normally. After the initiation of the RTS by the sender the receiver replies with CTS and a BOV (back of value). This BOV is used to instruct the sender that it has to wait for DIFS time plus the specified BOV before it attempts to send next data. Since we assume that our receiver is well behaved we can safely term the BOV as suitably random.

Now the sender attempts to send the data before the pre defined BOV time slots are over. The 'H' neighboring nodes of the sender observe the arrival of the CTS and the first attempt to send data. They separately calculate the time slots that elapsed between the occurrences of the above said events as turnaround time. While the receiver itself might

Calculate the time elapsed when it receives the data from the sender there is a possibility that the time slots counted by the sender and receiver might vary. Let us assume the condition in which the network in the locality of the sender is idle while in the vicinity of receiver it is busy. Now the sender is bound to count additional time slots while the receiver mutedly waits for the channel to be free. Thus even if the sender is not misbehaving the receiver might assume it is misbehaving if it receives the data sooner than the expected BOV time slots are elapsed. Thus we take steps to remove any penalty for an unsuspecting node which is not misbehaving.

The 'H' turnaround values calculated by the 'H' nodes in the neighborhood are sent to the receiver. The receiver now takes up the job of determining whether the sender is misbehaving or not. The receiver now calculates the average of the 'H' values received. Now the receiver checks for the following condition.

If Avg['H' values] <BOF Ctr++.

If the average of the H values sent by the neighboring nodes of the sender is less than the expected BOF then the counter is incremented by one. After each subsequent increment the Ctr is compared with the UL. If the counter value is more than the threshold then the sender node is termed as misbehaving. This information is broadcast to all the nearby nodes thus effectively blacklisting the node. The threshold value is so set, to allow the sender node the benefit of the doubt that some of the nodes in the

neighboring region are unable to calculate or send correct turnaround values.

Now after each time a node is suspected of misbehavior its counter is incremented and then node calculates the new back off value. Because the sender is suspected of misbehavior steps are taken to null or void the throughput advantage that was gained by the sender. The new back off time is calculated with the old back off time and a penalty is added to it.

Penalty calculation: The penalty added should be proportional to the amount of throughput achieved by the misbehaving sender. The aim is to discourage the node to misbehave:

ew BOF= old BOF +P

The formulation of the penalty is done as a function of the counter variable corresponding to the sender node maintained in the receiver:

P=f (ctr) = ctr*slot time

Scenario 2: Receiver misbehavior: Now let us assume that the receiver is misbehaving while the sender exhibits normal behavior. This case comes into consideration if the receiver was to derive any benefit by receiving data more frequently from the sender. When the receiver is a client and sender is a server allowing a single client to utilize more bandwidth than other clients is not favored.

After the sender has initiated the transmission with a RTS, the receiver sends a CTS and a back off value. In order to receive more frequently from the sender, the selfish receiver might choose to send a non random back off value which is on the shorter side. To prevent this type of misbehavior a node which assumes the role of a sender is expected to add the following functionality to its existing architecture.

The sender node maintains the history of 'x' recent back off values sent by the receiver. It then calculates an autocorrelation function for every 'x' recent value. If it finds the two sets of values too dissimilar it could suspect the receiver node of sending non random back off values. It could then check the 'x' values and if a majority of them are less than a threshold value the sender node is suspected of trying to utilize more bandwidth. Each time this behavior is observed a counter is incremented. As before if the counter exceeds the UL value the suspected node is termed as misbehaving node.

Scenario 3: Colluding nodes: In the worst case scenario both the sender and the receiver might collude to exploit the bandwidth in the channel shared by many other nodes. To prevent this condition a new functionality is added to each node. The receiver node calculates the back off value in response to the RTS from a sender. The calculation of the BOF is defined as follows:

BOF=f (rand, slot time)*CW min

CW min is the minimum value of the congestion window. The rand is a random value chosen by the receiver. A receiver might be motivated to choose this value without randomness. To detect this, the rand value is expected to be broadcasted to the neighboring nodes of the receiver node.

Each of the 'H' neighboring nodes now calculates the BOF using the rand value and exchanges them with each other by a broadcast. Each of the neighboring nodes now calculates the average of the H values and compares them with the expected BOF:

AVG [BOF 1-5] = BOF_{exp}

This ensures that none of five nodes could possibly collude with the sender and the receiver and derive and benefits. Any of the received BOF values, if found too deviant from the average BOF value, is a clear indication of a collusion attempt.

On the other hand if the sender does not follow the back off time given by the receiver and the receiver does not report it the neighboring nodes could listen in on the transmission and calculate the actual BOF. This could be used to judge whether the sender is following the BOF provided by the receiver and whether receiver is concerned about the misbehavior of the sender. Thus this model intuitively handles the inherent problem that occurs with Mac misbehavior.

Mobility model: A mobility model should attempt to mimic the movements' of real MNs. Changes in speed and direction must occur and they must occur in reasonable timeslots.

The Random Waypoint Mobility Model includes pause times between changes in direction and/or speed An MN begins by staying in one location for a certain period of time (i.e., a pause time). Once this time expires, the MN chooses a random destination in the simulation area and a speed that is uniformly distributed between *[minspeed, maxspeed]*. The MN then travels toward the newly chosen destination at the selected speed. Upon arrival, the MN pauses for a

specified time period before starting the process again.In order to alleviate this type of behavior and promote a semi-constant number of neighbors throughout the sim- ulation, the Random Direction Mobility Model was developed In this model, MNs choose a random direction in which to travel similar to the Random Walk Mobility Model. An MN then travels to the border of the simulation area in that direction. Once the simulation boundary is reached, the MN pauses for a specified time, chooses another angular direction (between 0 and 180 degrees) and continues the process

RESULTS AND DISCUSSION

To analyze the performance of our CNap and Nack algorithm in mitigating the problems due to misbehaving nodes, we made some modifications to 802.11 IEEE and the AODV specifications.

We have evaluated our extensions using the following metrics:

Network throughput: This is the ratio successfully received data packet to the actually sent data packets in the network.

$$T = \frac{\sum node\_rd}{\sum node\_sd}$$

Node_rd = Total No of successfully received data packets by node i.
Node_sd = Total no of sent data packets by node i

Average end to end delay: This is the average of the time taken by the packets to reach the destination in the network.

$$Avg\_latecncy = \sum_{I=1} \frac{T\text{-}eed(I)}{\frac{T\text{-}pkt(i)}{T\text{-}rec\text{-}node}}$$

Where N = Total no of nodes
T-rec-node = no of receiver nodes
T-eed(I = Sum of end to end delay of all received packets at node i
T-pkt(i) = Total no of received packets at node if

Packet delivery ratio: The ratio between the numbers of packets originated by the application layer to those delivered to the final destination.

Routing overhead: The number of routing packets transmitted per data packet delivered at the destination.

The Fig. 1 plots the decrease in the throughput as the percentage of selfish nodes increase. The fall of the throughput is very gradual even when the total numbers of nodes are increased. This plays testimony to the fact that the penalty scheme used in the CNap is smoother on nodes which are not misbehaving and thus the overall throughput of the network is not reduced to the great extent.

The Fig. 2 plots the decrease in the throughput along with the increase in the number of nodes. While the fall of the throughput is inevitable, once again we see a gradual and easy decrease in the throughput. The CNap provides a high degree of misbehavior detection while maintaining the network throughput in an optimum level.

The Fig. 3 depicts the gradual decrease in the packet delivery ratio with increase in the incidence of misbehaving nodes in the network under AODV with our Nack implementation.

The Fig. 4 Portrays a slow increase in the delay factor because of our Nack scheme implemented over AODV.

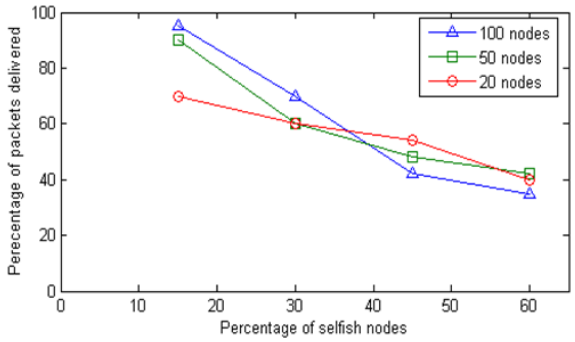


Fig 1: Percentage of selfish nodes Vs Packets delivered

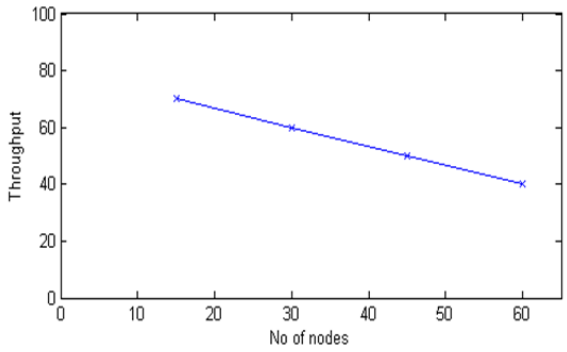


Fig 2: No of nodes Vs Throughput

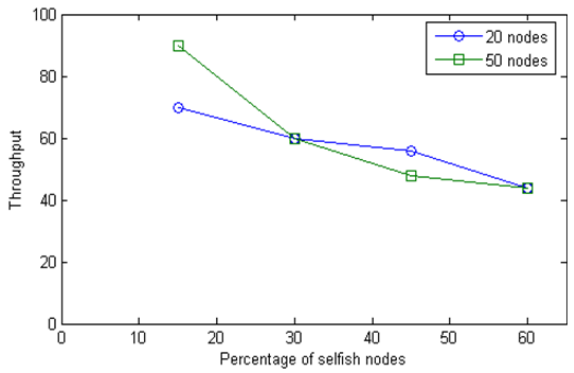


Fig 3: Percentage of selfish nodes Vs Throughput

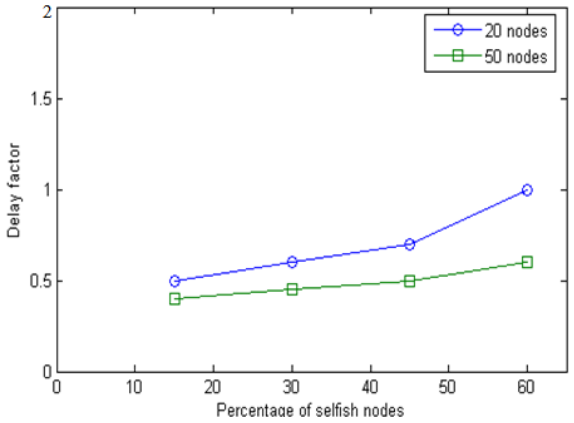


Fig 4: Percentage of selfish nodes Vs Delay factor

CONCLUSION

The Nack and Cnap schemes provide a comprehensive scheme to counter most of the misbehavior patterns in a MANET without compromising the throughput of the network. This solution also offers an alternative to the very many sub optimal solutions that are around right now. The future work is towards the aim to minimize false detection of sender or receiver nodes as misbehaving due to collisions and variations in local network environment.

REFERENCES

Babakhouya, A., Y. Challal and A. Bouabdallah, 2008. A simulation analysis of routing misbehaviour in mobile ad hoc networks. Proceedings of the 2nd International Conference on Next Generation Mobile Applications, Services, and Technologies, Sept. 16-18, IEEE Computer Society Washington, DC, USA., pp: 592-597. DOI: 10.1109/NGMAST.2008.56

Balakrishnan, K., J. Deng and V.K.Varshney, 2005. TWOACK: Preventing selfishness in mobile ad hoc networks. Proceedings of the IEEE Wireless Communications and Networking Conference, Mar. 13-17, IEEE Xplore, USA., pp: 2137-2142. DOI: 10.1109/WCNC.2005.1424848

Buttyan, L. and J.P. Hubaux, 2003. Stimulating cooperation in self-organizing mobile ad hoc networks. Mobile Networks Appl., 8: 579-592. DOI: 10.1023/A:1025146013151

Gunasekaran, R., U.V. Rhymend, R. Sudharsan, P.S. Sujitha and U. Yamini, 2008. Detection and prevention of selfish and misbehaving nodes at Mac layer in mobile and hoc networks. Proceedings of the Canadian Conference on Electrical and Computer Engineering, May 4-7, IEEE Xplore, Niagara Falls, ON., pp: 001945-001948. DOI: 10.1109/CCECE.2008.4564883

Hu, Y., D.B. Johnson and A. Perrig, 2003. SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks. Ad Hoc Networks, 1: 175-192. DOI: 10.1016/S1570-8705(03)00019-2

Huang, R., Y. Shuang and Q. Cao, 2009. Simulation and analysis of protocols in ad hoc network. Proceedings of the International conference on Electronic Computer Technology, Feb. 20-22, IEEE Xplore, Macau, pp: 169-173. DOI: 10.1109/ICECT.2009.66

Kaabneh, K., A. Halasa and H. Al-Bahadili, 2009. An effective location-based power conservation scheme for mobile ad hoc networks. Am. J. Applied Sci., 6: 1708-1713. DOI: 10.3844/ajassp.2009.1708.1713

Kyasanur, P. and N. Vaidya, 2003. Detection and handling of MAC layer misbehavior in wireless networks. Proceedings of the International Conference on Dependable Systems and Networks, June 22-25, University of Illinois , USA., pp: 173-182. DOI: 10.1109/DSN.2003.1209928

Marti, S., T. Giuli, K. Lai and M. Baker, 2000. Mitigating routing misbehavior in mobile ad hoc networks. Proceedings of the 6th annual international conference on Mobile computing and networking, (MobiCom'00), ACM New York, NY, USA., pp: 255-265. DOI: 10.1145/345910.345955

Natsheh, E. and K. Buragga, 2010. Density based routing algorithm for spare/dense topologies in wireless mobile ad-hoc networks. Am. J. Eng. Applied Sci., 3: 312-319. DOI: 10.3844/ajeassp.2010.312.319

Raya, M., J.P. Hubuax and I. Aad, 2004. DOMINO: A system to detect greedy behavior in IEEE 802.11 hotspots. Proceedings of the 2nd International Conference on Mobile Systems, Applications and Services, (MOBISYS'04), ACM New York, USA., pp: 84-97. DOI: 10.1145/990064.990077

Zhong, S., J. Chen and Y.R. Yang, 2003. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications, Mar. 30-3 Apr., Yale University, USA., pp: 1987-1997. DOI: 10.1109/INFCOM.2003.1209220