An Agent-Based Approach To Nodes' Misbehaviour Detection In Mobile Ad-Hoc Networks

Otor Samera U., Akinyemi Bodunde O., Adekunle Adeyelu., Akumba Beatrice O., Aderounmu Ganiyu A.

Abstract: Existing Misbehaviour Detection Systems in Mobile Ad-hoc Networks (MANETs) are challenged with routing overhead and high latency resulting from complexity and failure to isolate and block misbehaving nodes for the reason that it is difficult to detect them as they participate fully in route finding. In this work, a Mobile Agent-Based Acknowledgement scheme (MAACK) was formulated to address this problem using an object oriented algorithm deployed to report misbehaving nodes to the source and destination by registering the Internet Protocol (IP) address of misbehaving nodes in their header. The scheme was simulated using Network Simulator-3 (NS-3) and results benchmarked with an existing scheme; the Enhanced Adaptive Acknowledgment (EAACK) using packet delivery ratio, routing overhead and latency as performance metrics in the two scenarios. The results showed that the MAACK paradigm guaranteed a higher packet delivery ratio, lower latency and routing overhead than the EAACK scheme. The model can be adapted by Ad-Hoc network protocol developers.

Keywords: MANET, Misbehaviour, Mobile agents.

1 INTRODUCTION

A Mobile Ad-Hoc Network (MANET) is a set of mobile nodes (hosts) linked together temporarily, They communicate with each other via wireless links either directly or relying on other nodes as router for a particular purpose such as rescue missions, teleconferencing to mention just a few [1]. Existing mechanism in Mobile Ad-hoc Networks is faced with challenges of overhead and latency. It is difficult to detect misbehaving benign nodes because these nodes participate fully in route finding. More so, it is difficult to detect misbehaving nodes in the presence of partial dropping of packets since nodes selectively drop packets. Existing scheme in an attempt to address these problems employed complex solutions resulting to latency and network overhead. Due to limited resources of MANET overhead and latency may result to lack of cooperation between nodes. By reducing these metrics it is possible to reduce packet droppers in the network as most packet droppers do so as a result of limited resources of MANET. Hence, there is a need for a model that will detect, isolate and block these nodes with less complexity, less overhead and latency. Therefore this study aimed at employing a Mobile Agent-based Acknowledgement (MAACK) scheme to identify and block nodes that misbehave in MANETs by dropping of packets using the various advantages of mobile agents to take care of overheads experienced in acknowledgement schemes.

2 RELATED WORKS

Reliable network connectivity in wireless networks is achieved if some counter measures are taken to avoid data packet forwarding against malicious attacks. A lot of routing misbehavior models have been designed by researchers to avoid malicious attackers as firewalls and encryption techniques are no longer sufficient and effective for protecting ad-hoc wireless network [1]. Credit-based schemes were first proposed to address the problem [2]. According to Zhong et al.[3], Younghwan et al.[4], Dimitris et al.[5], Hamed et al.[6] Guo et al.[7] and Hungmin et al., [8], this system provides incentives for nodes which perform faithful networking functions. That is, nodes get paid incentives in the form of virtual currency or similar type of payment setup, for providing services to other nodes. Credit-based systems succeeded in stimulating cooperation in networks with selfish nodes using credit payments, even though credits are useful only when an action and its reward are not simultaneous. The significant problem with these systems is that it is very difficult to charge users fairly without introducing additional complexity. The mechanisms used to implement these incentives take up resources themselves, however if the number of cheating nodes is not too high, then the benefit derived from the application of the incentive mechanisms may be outweighed by the resources they consume. Finally, when using tokens, there is the question of how the balance of tokens can be maintained for users; the average token level within the system needs to be kept at a reasonable level in order for incentives to work properly. Based on these problems, reputation-based schemes, in which reputation is built based on trust value of individual nodes was proposed. The trust value is determined by continuous interaction between the nodes as they earn each other's trust through past experience. Network nodes collectively detect and declare the misbehaviour of the suspicious node. This declaration is then propagated throughout the network so that the misbehaving node is removed or isolated from the rest of the network. Reputation based schemes are classified as acknowledgement based schemes and Agent based schemes. The earliest work upon which most reputation-based schemes were based is the Marti et al. [9] where

- *Otor Samera U. is currently pursuing p.hD degree program in Computer science in Obafemi AwolowoUniversity Ile-Ife, Nigeria, +2348146079250. E-mail: jopadev@yahoo.com*
- *Adeyelu Adekunle (phd) and Akumba Beatrice O: currently staff of Benue State University Makurdi Nigeria, Computer Science Department +2348067434301 and +2348964994010*

Watchdog and Pathrater models were proposed, and it runs on the Dynamic Source Routing (DSR) protocol. These models were capable of detecting malicious nodes rather than links which makes it popular choice in the field; however it fails to detect malicious misbehaviour in the presence of ambiguous collisions, receiver collisions, limited transmission power, false misbehaviour report, collusion and partial dropping. In attempt to solve the problems identified in [9], CONFIDANT (Cooperation of Nodes, Fairness in Dynamic Ad-hoc NeTworks) was proposed in [10]. It is an extension to DSR protocol which is similar to Watchdog and Pathrater. But it could give more opportunities for attackers to send false alarm messages that a node is misbehaving while this isn’t actually the case since this protocol allows nodes in the network to send ALARM messages to each other. This ALARM messages also introduced overhead to the network. Another protocol in this scheme is the 2ACK scheme in [11], where packets are acknowledged after they have been delivered by two consecutive intermediate nodes from the source. The advantage of this scheme is that it successfully solves the receiver collision and limited transmission power problems posed by Watchdog; however the acknowledgement process required in every packet transmission process also adds a significant amount of unwanted network overhead. Due to the limited battery power nature of MANETs, such redundant transmission process can easily degrade the life span of the entire network. Furthermore, Kang et al., [12] in attempt to reduce this overhead proposed Enhanced Adaptive ACKnowledgement (EAACK) to solve four significant problems of Watchdog mechanism, such as ambiguous collisions, receiver collisions, limited transmission power and false misbehaviour report. His work was further enhanced in [13] to include digital signature as a security means, however it incurred more overhead. Muhammad et al.,[14] also proposed a novel Adaptive Trust Threshold (ATT) computation strategy, which adapts the trust threshold in the routing protocol according to network conditions such as rate of link changes, node degree and connectivity, and average neighborhood trustworthiness. The topology factors that affect the trust threshold at each node was identified, and leveraging them to build a mathematical model for ATT computation. Simulation results indicate that the ATT strategy achieved significant improvements in packet delivery ratio, reduction in false positives, and increase in detection rate as compared to traditional static threshold strategies. More so, in Ming et al,[15] , detection and defense schemes to identify and defend against MAC-layer selfish misbehavior, respectively, in IEEE 802.11 multi-hop ad hoc networks was proposed. It is a realtime selfish misbehavior detection scheme for multi-hop ad hoc networks. It requires only several samples, and hence is more efficient and can adapt to channel dynamics more quickly. Based on the proposed detection scheme, three selfish misbehavior defense schemes against three typical kinds of smart selfish nodes was designed. Results showed that the smart selfish nodes could not degrade normal nodes' performance much without getting detected. In conclusion, existing schemes in an attempt to address misbehaviour challenges of MANET especially those identified by Marti et al.,[9], routing Overhead and complexity became a major challenge. Marti et al.,[9] model upon which these schemes base their solution is simple and achieved zero overhead however it fails to detect malicious misbehaviour in the presence of ambiguous collisions, receiver collisions, limited transmission power, false misbehaviour report, collusion and partial dropping. Therefore, there is a need to address these challenges. It was observed that some of these schemes perform better than others while addressing these challenges, however due to complexity, routing overhead, partial dropping and sometimes latency still remains a challenge. In this paper, attempt was made to develop a model for misbehaving benign nodes which combined mobile agent scheme with acknowledgement based scheme so as to address the problem of overhead experienced in acknowledgement schemes using the various advantages of mobile agents. The uniqueness of proposed scheme as compared to existing schemes is that, it is simple yet able to detect packet droppers while maintaining less network overhead and latency.

3 MATERIALS AND METHOD

The proposed model is referred to as the mobile agent acknowledgement (MAACK) scheme. It combines the advantages of mobile agent schemes with acknowledgement schemes. It runs on an existing routing protocol and it functions at the data forwarding stage of routing. The model presents two special reactive agent packets called Mobile Agent Acknowledgment (MAACKP) and Mobile Agent Not Acknowledged (MNAACKP) packets in addition to the default UDP agent packet which serves as the data packet. Only the source node can send UDP packet as it is assumed that the source and destination nodes are not malicious, while the MAACKP and MNAACKP agent packets can be instantiated only at the intermediate (forwarder) nodes that are malicious. The scheme starts by setting the threshold of malicious nodes called ErrorCount to zero in the routing table. It is assumed that a route to the destination has been discovered by the routing protocol and a source node S wishing to transmit a packet to destination node D forwards a UDP agent packet destined for a predefined destination as contained in the route, the UDP agent packet on reaching the forwarder node checks the routing table to determine the threshold of misbehaviour of the source of the packet, if the misbehaviour threshold referred to as the ErrorCount exceeds 2 the packet is dropped which serves as a punishment to the misbehaving node. However, if the threshold is zero or below 2 it forwards the packet as usual to the next forwarder node until it gets to the destination. This scenario is when there are no packet droppers in the network. All intermediate nodes apart from the source and destination nodes are referred to as the forwarder nodes and the threshold is set to 2 for early detection. In the second scenario in which some intermediate nodes are set to misbehave by dropping packets as they are received, A MAACKP agent is generated as soon as there is malicious drop. This MAACKP registers the address of the dropping node called ErrorHop(the IP address of the misbehaving node) in its header, looks for an alternative route from the routing table and moves to the destination node. The destination node upon receipt of a MAACKP knows that the packet was to be dropped by the ErrorHop registered by the MAACKP, then update the routing table and increase the misbehaviour threshold of the node for future reference.

However if an alternative route is not found by the MAACKP agent an MNAACKP agent is created which follows the reverse route back to the source registering in its header the ErrorHop also. The source node upon receipt of an MNAACKP updates its routing table for future reference and increase the misbehaviour threshold of the misbehaving node. The pseudocode for the proposed model is as presented below and the flowchart as illustrated in Fig. 1.

3.1 Pseudocode Of Proposed Model

```
Initialization
             ErrorHop = “0.0.0.0”
             ErrorCnt = 0
             For each packet received check routing table for
             misbehaving node
              ErrorHop
             And corresponding error count ErrorCnt
             if ErrorCnt >= 2 then Isolate misbehaving nodes
            ErrorHop
           else forward packet
          end if
         if drop packet by exhibiting misbehaviour then
         Create a MAACKP agent
         Set misbnode = ErrorHop
         Set errorcount = ErrorCnt
                    Lookup alternative route
                if route found then
               Forward MAACKP to destination
                Update routing table ErrorHop, ErrorCnt ++.
                 else forward MNAACKP to source
               Update routing table ErrorHop, ErrorCnt++
               end if
             end if
           end.
```

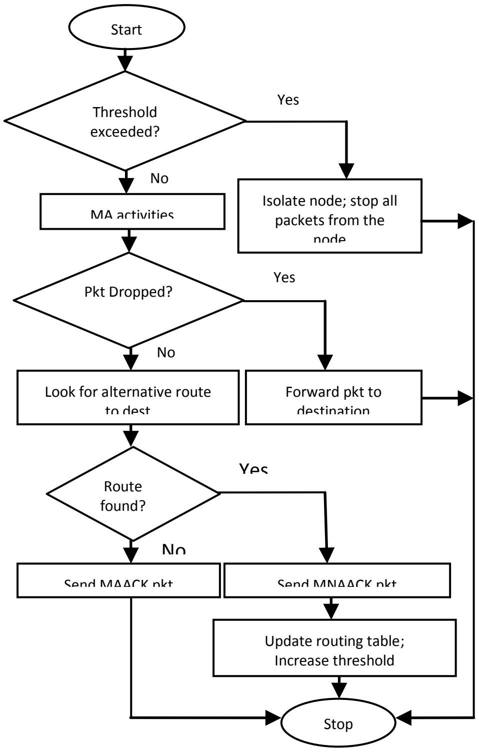
Start
Threshold exceeded?
Yes
No
MA activities
Isolate node; stop all packets from the node
Pkt Dropped?
Yes
No
Look for alternative route to dest
Forward pkt to destination
Route found?
Yes
No
Send MAACK pkt
Send MNAACK pkt
Update routing table; Increase threshold
Stop

Fig. 2. *Flowchart of the proposed MAACK Model*

TABLE 1
PARAMETER SETTINGS FOR MAACK MODEL

Parameters	Values
Number of Nodes	50
Dimension of Topology (x)	300m
Dimension of Topology (y)	1500m
Channel type	WI-FI
Mac Protocol	802.11b
MobilityModel	RandomWayPoint
Packet Size(Byte)	64
Number of Agents	3
Bit Rate	CBR
Number of Sinks	10
Number of Misbehaving Nodes	0% - 40%
Simulation Time(Secs)	100

Network Simulator 3(NS3) tools were employed to simulate the model formulated. The core of NS3 was written in C++ programming language and with Python scripting interface. The implementation of MAACK model is treated offline in this paper. The simulation was set up using the parameters in Table 1. Simulation was run on a Dell laptop on Linux operating system Ubuntu 10.04 LTS version.

4 RESULTS AND DISCUSSION

Performance comparison of Packet Delivery Ratio (PDR) was performed between the EAACK model and MAACK model. Two scenarios were considered for measuring Packet Delivery ratio. Scenario 1 isolates the misbehaving nodes while scenario 2 only punishes the misbehaving nodes by dropping all packets received from them. The degrees of packet delivery between the two schemes in scenario 1 and 2 are as shown in Fig. 3 and 4 respectively. In the first scenario, it’s been observed that MAACK achieved a higher delivery ratio than EAACK by a total of 28.9% this shows that the detection efficiency of our model MAACK as compared to EAACK is higher, a higher PDR shows that MAACK detect more malicious nodes than EAACK according to Kang et al [13].

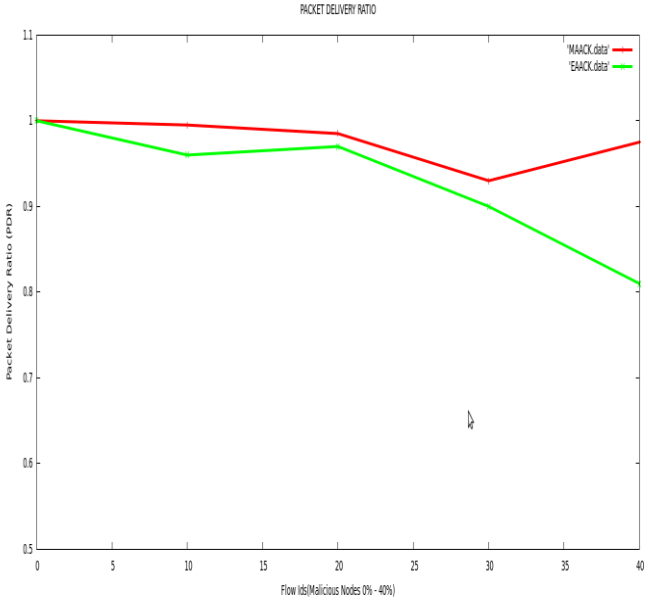

Figure 3: *Packet Delivery Ratio vs. FlowIDs(Malicious Nodes Range: 0%-40%) of Scenario1*

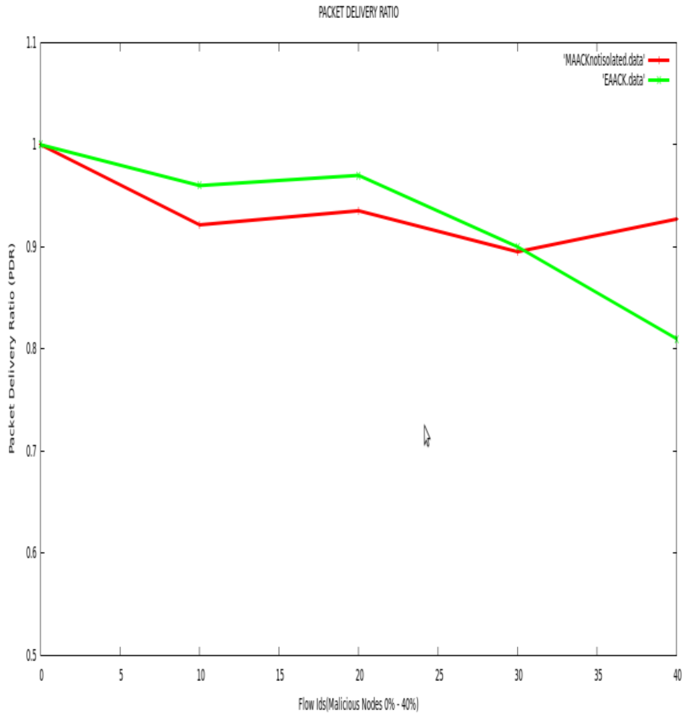

Figure 4: *Packet Delivery Ratio vs. FlowIDs(Malicious Nodes Range: 0%-40%) of Scenario2*

In the second scenario, it's been observed that EAACK outperforms MAACK by 8.13% at 0% to 30% malicious nodes, this is because the punitive measure applied to the misbehaving nodes was a dropping scheme, and therefore the flow monitor identifies them (packets dropped to punish misbehaving nodes) as a lost packets, hence the result. In future we will see how this can be captured separately. At 30% and above however the MAACK outperforms EAACK by 14.44% this is as a result of less flow of packets through some nodes in the network as packets from the malicious nodes are dropped. One other performance metric used in this simulation is the routing overhead. The routing overhead of the two schemes were compared. The routing overhead increases as the number of nodes increases on the network, the increase in EAACK scheme is significantly higher than the MAACK scheme as shown in Fig. 8. It is obvious in this simulation that there is a reduction in the routing overhead in MAACK scheme than EAACK scheme by 66.67%, 38.31%, 72.35%, 57.27% and 66.14% for 0% to 100% malicious nodes. Thus MAACK scheme has a better performance in detecting malicious nodes in terms of reduced overheads.

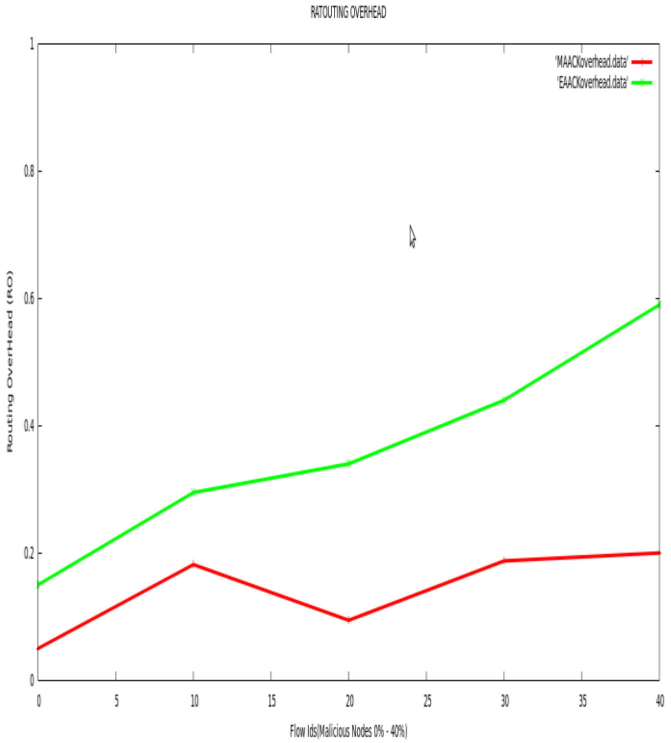

Figure 5: *Routing Overhead vs. FlowIDs (Malicious Nodes Range: 0% - 40%)*

The third performance metric used in this simulation is the end-to-end delay. The rates of latency of the two schemes were compared as shown in Fig. 6. It's been observed from the simulation that Latency was reduced by MAACK as compared to AODV protocol to normal with the use of Mobile Agents between flowIDs 90 to 100 at 47.37% to 100% which is one of the advantages of using mobile agent in this study. Also at FlowIDs 1, 10, 30, 40, 50 and 80 MAACK achieved a latency of. 55%, 92%, 75.57%, 49.85%, 83.70%, and 1.25% lower than EAACK and at flowIDs 20, 60 and 70 it achieves 47.86%, 0.5%, and 0.42% higher than EAACK. In summary at 90 and above it achieves 100% less. This shows that the mobile agent responded real time at this point with delay sum equal to zero.

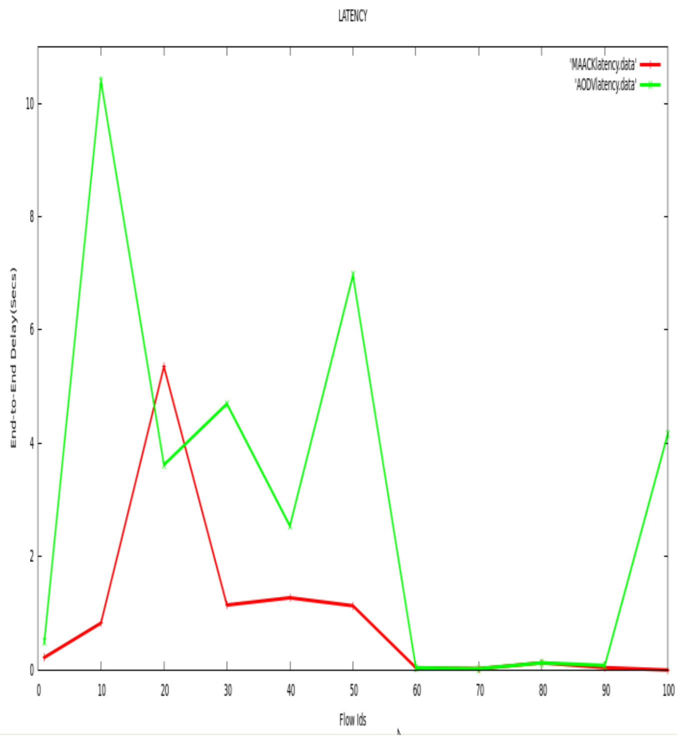

Figure 6: *End-to-End delay vs. FlowIDs*

5 CONCLUSION AND RECOMMENDATION

In this paper, an agent-based approach to identify and block misbehaving benign nodes in mobile ad-hoc networks was presented. This was achieved by using the various advantages of mobile agents to reduce routing overhead and latency resulting from complexity and failure to isolate and block misbehaving nodes. A comparative analysis was performed between the existing EAACK and the proposed MAACK schemes. The proposed model achieved a higher detection rate, reduced routing overhead and latency than the existing EAACK model. Therefore the proposed model can be incorporated into existing routing protocols to reduce network overhead and latency for better performance in MANETs.

Reference

[1] P. S. Neelavathy and D. Sridharan. "A Performance Comparison and Evaluation of Analysing Node Misbehaviour in MANET using Intrusion Detection System." International Journal of Computer Science and Engineering Technology (IJCSET). February 2011, pp.35-40.

[2] L. Buttyán and J. Hubaux J. "Stimulating cooperation in self-organizing mobile ad hoc networks. "Journal of Mobile Networks and Applications, vol 8 no. 5, October 2003. pp 579 – 592, 2003

[3] S. Zhong, J. Chen. and Y. R.Yang. " Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks." IEEE INFOCOM., pp.1987–1997, 2003

[4] Y. Younghwan, S. Ahn and D. Agrawal. "Impact of a simple load balancing approach and an incentive-based scheme on MANET performance." Elsevier Journal of Parallel and Distributed computing. Vol. 70 no. 2 pp 71-83, 2010

[5] E.C. Dimitris, D.G. Konstantinos and P. Athanasios. "ICARUS: hybrid inCentive mechAnism for coopeRation stimUlation in ad hoc networkS". Elsevier journal of Ad Hoc Networks. Vol. 10 no. 6 pp 976–989, 2012

[6] J. Hamed, K. Fayazbakhsh, B. Bakhshi and M. Dehghan. "A Novel Incentive-Based and Hardware-Independent Cooperation Mechanism for MANETs." In: IEEE Communication society in the WCNC 2008 proceedings., pp.1525-3511, 2008

[7] Y. Guo, J. Ma, C. Wang and K. Yang. "Incentive-Based Optimal Nodes Selection Mechanism for Threshold Key Management in MANETs with Selfish Nodes." International Journal of Distributed Sensor Networks (IJDCN). pp.1-13, 2013

[8] S. Hungmin, C. Chen Chiunghsun, and Y. Ku. "A novel acknowledgment-based approach against collude attacks in MANET". Elsevier journal of Expert Systems with Applications. Vol 39, pp.7968–7975.

[9] S. Marti, T.J. Giuli, K. Lai and M. Baker, (2000). "Mitigating routing misbehaviour in mobile ad hoc networks." In: Proceedings of the 6th annual international conference on Mobile computing and networking. Boston, Massachusetts: ACM. pp 255-265, 2000

[10] S, Buchegger, . and J.L. Boudec, "Performance Analysis of the CONFIDANT Protocol Cooperation of Nodes-fairness in dynamic ad-hoc networks". In Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking &computing. New York, USA. pp 226 - 236 , 2002

[11] L. Kejun, D. Jing , K.V. Pramod and B. Kashyap. "An Acknowledgment-Based Approach for the Detection of Routing Misbehavior in MANETs". IEEE Transactions on Mobile Computing. Vol. 6 no. 5 pp.488-502, 2007

[12] N. Kang, E. Shakshuki and T. Sheltami. "Detecting Misbehaving Nodes in MANETs." In the proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services .Paris, France, ACM, pp.216-222, 2010

[13] N. Kang, E. Shakshuki and T. Sheltami. "Detecting Forged Acknowledgements in MANETs". In proceedings of the IEEE International Conference on Advanced Information Networking and Applications. pp.488-494, 2011

[14] S. K. Muhammad, M. Daniele, I. K. Majid and B. Elisa. " Adaptive Trust Threshold Strategy for Misbehaving Node Detection and Isolation". Trustcom/BigDataSE/ISPA, 2015 IEEE DOI: 10.1109/Trustcom.2015.439 . USA: IEEE Xplore., 2015

[15] L. Ming, S. Sergio, L. Pan and S. Jinyuan. "MAC-Layer Selfish Misbehavior in IEEE 802.11 Ad Hoc Networks: Detection and Defense". IEEE Transactions on Mobile Computing Vol.14, no. 6 , pp 1203 – 1217, 2015