2ACK-Scheme: Routing Misbehavior Detection in MANETs Using OLSR

Prof. Shalini V. Wankhade

Abstract— **A Mobile Ad Hoc Network (MANET) is a collection of mobile nodes (hosts) which communicate with each other via wireless links either directly or relying on other nodes such as routers. Due to the dynamic change in topology finding route is very difficult. Some nodes misbehaves as they participate in route establishment phase but refuse to forward the data packets to conserve their own energy. For detecting routing misbehavior in MANETs lot of techniques are there such as watchdog, pathrater, TWOACK, SACK, End to End ACK scheme. But due to the disadvantages of the above scheme we propose a new scheme called 2ACK and the routing protocol used is Optimized Link State Routing (OLSR). The fundamental concept of the 2ACK scheme is to send two hop acknowledgment in opposite direction to routing path, we simulate the results in NS-2.**

Index Terms—**. OLSR, MANETs, DSR, 2ACK scheme.**

I. INTRODUCTION

There has also been a lot of development in the field of wireless networking in recent years and also a lot of techniques have been put forward for reducing this packet loss that occurs due to selfish nodes. Finding the network topology and correctly delivering data is the whole and sole responsibility of the nodes that are involved in the communication process. Wireless networks can be classified as Infrastructure based networks:- Infrastructure based networks have a centralized base station. Hosts in the wireless network communicate with each other, and with other hosts on the wired network, through the base station.

Ad hoc networks:- Ad hoc networks are characterized by the absence of any infrastructure support.

Hosts in the network are self organized and forward packets on behalf of each other, enabling communication over multihop routes, that attempt to ensure a reasonably fair throughput share for all the participating hosts. In the network some nodes are untrusted, some hosts may misbehave by failing to adhere to the network protocols, with the intent of obtaining an unfair share of the channel. The presence of selfish hosts that deviate from the contention resolution protocol can reduce the MANET's are self organizable and configurable hence also known as multi hop wireless ad hoc networks, where the topology of the network keeps on changing continuously. The procedure of MANET is not depending on existing base stations or infrastructure. Network clients in MANET may move freely and randomly. Therefore, the network topology of a MANET can be change unpredictably and speedily. As these nodes have the flexibility of moving from one place to other, there may be cases wherein a particular node which is a receiver for a particular packet, moves away from the range of sender. However the sender is not aware of this scenario and it might still keep on sending packets thus leading to packet and data loss. The other case is a bit more interesting, whenever communication takes place between any two nodes there are a lot of nodes involved in this communication process and acting as mediators. All these nodes agree to forward packets during the actual communication process but one of them actually turns selfish during the data transfer, this selfish 2 node keeps on dropping packets as when received instead of forwarding it to the next hop in the communication process. These selfish behavior results in packet loss and also the source is unaware of such misbehaving node in the path towards the destination There are two types of MANETs:

Open MANET:- An open MANET comprises of different users, having different goals, sharing their resources to achieve global connectivity, as in civilian applications.

Closed MANET:- In closed MANET where the nodes are all controlled by a common authority, have the same goals, and work toward the benefit of the group as a whole. Open environment of a MANET may lead to misbehaving nodes. Performing network functions consumes significant energy of participating nodes, as communication is relatively costly. Selfish nodes are unwilling to spend their precious resources for operations that do not directly benefit them [5]. MANET lack a centralized monitoring and management point, making it a challenging task to detect such misbehaving nodes effectively.

Non-cooperative actions of misbehavior are usually termed as selfishness, which is notably different from malicious behavior. Selfish nodes use the network for their own communication, but simply refuse to cooperate in forwarding packets for other nodes in order to save battery power. A selfish node would thus utilize the benefits provided

by the resources of other nodes, but will not make available its own resources to help others. They have no intention of damaging the network [4].

II MISBEHAVIOR IN MANETS

Adhoc wireless network maximizes network throughput by using all available nodes for routing and forwarding. Therefore, the more nodes that participate in packet routing, the greater the aggregate bandwidth, the shorter the possible routing paths, and the smaller the possibility of a network partition. However, a node may misbehave by agreeing to forward packets and then failing to do so, because it is overloaded, selfish, malicious, or broken. An overloaded node lack the CPU cycles, buffer space or available network bandwidth to forward packets. A selfish node is unwilling to spend battery life, CPU cycles, or available network bandwidth to forward packets not of direct interest to it, even

though it expects others to forward packets on its behalf. A malicious node launches a denial of service attack by dropping packets. A broken node might have a software fault that prevents it from forwarding packets [2]. Misbehaving nodes can be a significant problem. There are three types of selfish nodes as follows:

Selfish Nodes Type 1 (SN1) - These nodes participate in route establishment but refuse to forward data packets (which are usually much larger than the routing control packets).

Selfish Nodes Type 2 (SN2- These nodes participate in neither the route establishment phase nor forward data packets. They only use their energy for transmissions of their own packets.

Selfish Nodes Type 3 (SN3) - These nodes behave (or misbehave) differently based on their energy levels. When the energy lies between full energy E and a threshold T1, the node behaves properly. For an energy level between T1 and another lower threshold T2, it behaves like a node of type SN1. Finally, for an energy level lower than T2, it behaves like a node of type SN2. The relationship between T1, T2, and E is $T2 < T1 < E$. The existence of the SN2 type nodes is simply ignored by the routing protocol. Thus, these nodes do not pose a significant threat to the normal operation of the routing protocol, even though they may degrade network connectivity. The SN1 and SN3 categories of nodes, on the other hand, are more dangerous to routing protocols [10]. We study these in Optimized State Link State Routing (OLSR) protocol. The SN1 and SN3 nodes participate in the route establishment process through the exchange of control messages (HELLO and TC) but interrupt the data flow. The sending node may use available alternative path from the routing table, if one such path is available. The newly selected route may still include some of these SN1 type nodes, and hence the new route will also fail. This process will continue until the source of traffic concludes that data cannot be transferred. In this work, we focus only on the detection and mitigation of SN1 type misbehavior. SN3 type nodes will be detected when they behave similar to the SN1 type nodes [3].In some cases it might happen that the node which is dropping packets might have fault in the software running at its end. We need to therefore focus on how we can detect this misbehaving node in order to decrease the packet loss. There are a lot of schemes available in order to detect the routing misbehavior in MANETs.

III RELATED WORK

Various techniques have been proposed to prevent selfishness in MANETs

Credit-Based Schemes: The basic idea of credit-based scheme is to provide incentives for nodes to faithfully perform networking functions. In order to achieve this goal, virtual (electronic) currency or similar payment system may be set up. Nodes get paid for providing services to other nodes. When they request other nodes to help them for packet forwarding, they use the same payment system to pay for such services. The concept of nuggets (also called beans) is used for payments for packet forwarding. There are two models which use nuggets are: Packet Purse Model and Packet Trade Model.

Packet purse model: In the Packet Purse Model, nuggets are loaded into the packet before it is sent. The sender puts a certain number of nuggets on the data packet to be sent. Each intermediate node earns nuggets in return for forwarding the packet. If the packet exhausts its nuggets before reaching its destination, then it is dropped. In another implementation, each node maintains a counter termed the neglect counter. The counter is decreased when the node send packet of its own, but increased when it forwards packet for the other nodes. The counter should be positive before a node is allowed to send its packet. Therefore, the nodes are encouraged to continue to help other nodes. Another credit-based scheme, termed Sprite, has nodes that keep receipts of the received/forwarded messages. When the users have a fast connection to a Credit Clearance Service (CCS), they report all of these receipts. The CCS then decides the charge and credit for the reporting nodes. In the network architecture of Sprite, the CCS is assumed to be reachable through the use of the Internet, limiting the utility of Sprite. The main problem with credit-based scheme is that they usually require some kind of tamper-resistant hardware and/or extra protection for the virtual currency or the payment system.

Packet trade model: In the Packet Trade Model, each intermediate node ”buys” the packet from the previous node for some nuggets and ”sells” it to the next node for more nuggets. Thus, each intermediate node earns some nuggets for providing the forwarding service and the overall cost of sending the packet is borne by the destination.

Reputation-based schemes : In such schemes, network nodes collectively detect and declare the misbehavior of a suspicious node. Such declaration is then propagated throughout the network so that the misbehaving node will be cut off from the rest of the network. Two modules under this category are watchdog and path rater. Nodes operate in a promiscuous mode wherein the watchdog module overhears the medium to check whether the next-hop node faithfully forwards the packet. At the same time, it maintains a buffer of recently sent packets. A data packet is cleared from the buffer when the watchdog overhears the same packet being forwarded by the next-hop node over the medium. If a data [4] packet remains in the buffer for too long, the watchdog module accuses the next hop neighbor of misbehaving. Thus, the watchdog enables misbehavior detection at the forwarding level as well as the link level. Based on the watchdog’s accusations, the path rater module rates every path in its cache and subsequently chooses the path that best avoids misbehaving nodes. The CONFIDANT protocol is another example of reputation based scheme. The protocol is based on selective altruism and utilitarianism, thus making misbehavior unattractive. CONFIDANT consists of four important components-the Monitor, the Reputation System, the Path Manager, and the Trust Manager. They perform the vital functions of neighborhood watching, node rating, path rating, and sending and receiving alarm messages, respectively. Each node continuously monitors the behavior of its first-hop neighbors. If a suspicious event is detected, details of the event are passed to the Reputation System. Depending on how significant and how frequent the event is, the Reputation System modifies the rating of the suspected node. Once the rating of a node becomes intolerable, control is passed to the Path Manager, which accordingly controls

the route cache. Warning messages are propagated to other nodes in the form of an Alarm message sent out by the Trust Manager [7]. The Monitor component in the CONFIDANT scheme observes the next hop neighbor's behavior using the overhearing technique. This causes the scheme to suffer from the same problems as the watchdog scheme.

End-to-End acknowledgment schemes: There are several schemes that use end-to-end acknowledgments (ACKs) to detect routing misbehavior or malicious nodes in wireless networks. In the TCP protocol, end-to-end acknowledgment is employed. Such acknowledgments are sent by the end receiver to notify the sender about the reception of data packets up to some locations of the continuous data stream. The Selective Acknowledgment (SACK) technique is used to acknowledge out-of-order data blocks. The 2ACK technique differs from the ACK and the SACK schemes in the TCP protocol in the following manner: The 2ACK scheme tries to detect those misbehaving nodes which have agreed to forward data packets for the source node but refuse to do so when data
packets arrive. TCP, on the other hand, uses ACK and SACK
to measure the usefulness of the current route and to take appropriate action. The Best-effort Fault-Tolerant Routing (BFTR) scheme also employs end-to-end ACKs. The BFTR scheme continuously monitors the quality (i.e., packet delivery ratio) of the path in use. This is compared with the predefined expected behavior of good routes. If the behavior of the route in use deviates from the behavior of good routes, it is marked as "infeasible" and a new route is used. Since BFTR throws out the entire route before detecting the misbehaving nodes, the newly chosen route may still include the same misbehaving nodes. Even though the new route will be detected as infeasible by the source after a period of observation time, data packet loss will occur in traffic flows when using protocols such as UDP. Such a repeated detection process is inefficient.

.

IV 2ACK-SCHEME

Notations and Assumptions:-

- Data pkt Unique ID (DUID) : This is used to record the unique packet id of the sent Data packet.
- Data pkt Sent Time (DST) : This records the time at which Data packet is sent.
- Cpkts : This gives the total number of Data packets that are sent.
- D2ACK : This records the Data packet ID for which the observing node has received the 2ACK packet i.e., 2ACK Packet is received for this Data packet.
- 2ART (2ACK Receive Time) : This records the time at which the 2ACK packet is received.
- R-CNT : It counts the total number of 2ACK packets that are received by the observing node.
- Cmiss : It counts the total number of Data packets for which the 2ACK packet is not received.
- Rmiss : It is the ratio of the Cmiss to the total number of Data packets sent, i.e., Cmiss/ Cpkts.
- Diff-Time : It is the difference of the time intervals 2ART and DST.
- DUID(Data pkt Unique ID) : This records received Data packet unique ID.
- DTotal : It counts the total number of Data Packets that are received.
- Rstatus : This records the status of Data packets for which 2ACK packet is sent. R-status =1 means for this DUID the receiving node has sent 2ACK packet and vice versa.

- R2ACK-Total : This field is used to count the total number of 2ACK packets that are sent.
- R-ack : This field gives the ratio of the total number of data packets acknowledge to the total number of Data packets received

A. Overview of 2ACK-scheme: 2ACK scheme importantly simplifies the detection mechanism Details of the 2ACK Scheme. The 2ACK scheme is a network-layer technique to find links and to extenuate their effects. It can be implemented as an add-on to existing path protocols for MANETs, such as OLSR and any other routing protocols. The 2ACK scheme finds a good behavior through the use of a new type of acknowledgment bundle, termed 2ACK. A 2ACK bundle is assigned a fixed path of two hops (three nodes) in the contrary direction o the data traffic path. The 2ACK scheme is a network-layer technique to detect misbehaving links and to mitigate their effects. It can be implemented as an add-on to existing routing protocols for MANETs, such as OLSR. The 2ACK scheme detects misbehavior through the use of a new type of acknowledgment packet, termed 2ACK. A 2ACK packet is assigned a fixed route of two hops (three nodes) in the opposite direction of the data traffic route. It can be implemented as an add-on to existing path protocols for MANETs, such as OLSR and any other routing protocols. The 2ACK scheme finds a good behavior through the use of a new type of acknowledgment bundle, termed 2ACK. A 2ACK
bundle is assigned a fixed path of two hops (three nodes) in the contrary direction of the data traffic path. At N1, each ID will remain on the record for 't' seconds, the respite for 2ACK reception. If 2ACK bundles matching to this ID arrive in front the timer exits, the ID will be took out from the records. Other than, the ID will be taken out at the last of its look out time separation and a counter called Cmis will be incremented. If N3 receives a data bundle, then calculated whether it wants to send a 2ACK bundle to N1. In order to cut down the extra path overhead reason by the 2ACK outline, only a divide the data bundle will be acknowledged verses multi hop bundle. Such a divide termed the acknowledgment proportion, Rack. By changing Rack, we can dynamically tune up the overhead of Many-Hop bundle transmissions. Client N1 remarks the behavior of link N2→N3 for a session of time Tobs. At the last of the session, N1 determines the proportion of losing 2ACK bundles as Cmis / Cpkts and compare it with a threshold Rmis. If the proportion is greater than Rmis, link N2→N3 is announced misbehaving and that particular link is being removed from the routing table. Since only a divide of the get data bundle are acknowledged, Rmis could simplify $Rmis > 1- Rack$ neglect false alerts reason by such a partially acknowledgment technique. Every client getting such a 2ACK packet remarks the link N2→N3 as

misbehaving and sums it to the black records list of such misbehaving links that it controls. When a client begins its own data traffic after, it will avoid using such misbehaving connects as a part of its path.

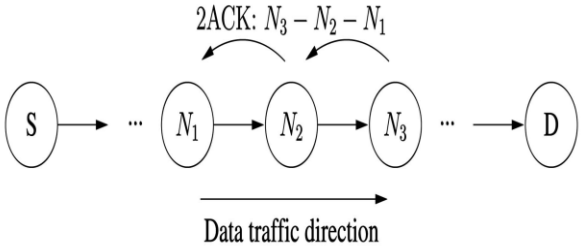

Fig: 4 Working of 2 ACK-Scheme

B. Comparison with Overhearing Techniques

Compared with the above schemes, the 2ACK scheme doesn't depend on end-to end acknowledgment. Instead, the 2ACK scheme tries to detect misbehaving links as the links are being used. Such a proactive detection approach results in quicker detection and identification of misbehaving links. In such a combined scheme, the Multi-Hop transmission and the monitoring processes are turned on only when routing carry outface degrades. It will further reduce the routing overhead of the 2ACK scheme. A scheme to choose routes based on the reliability index of each outgoing neighbor has each node maintaining a table of reliability indices of its neighbors Following are the disadvantages of overhearing techniques:-

Ambiguous Collisions: Ambiguous or doubtful collisions may occur at node N1. When a well behaved node N2 forwards the data packet toward N3, it is possible that N1 cannot overhear the transmission due to another concurrent transmission in N1's neighborhood. The 2ACK technique solves this problem by requiring N3 to send a Multi-Hop packet explicitly.

Receiver Collisions: Receiver or acceptor collisions take place in case of overhearing techniques when N1 overhears the data packet being forwarded by N2, but N3 fails to get the packet due to collisions in its neighborhood. The data packets will not be retransmitted by a misbehaving N2 because retransmission requires extra energy. Again, due to the explicit Multi-Hop packets our 2ACK scheme overcomes this problem.

Limited Transmission Power: A misbehaving N2 may engineer its transmission power in such a way that N1 can overhear its transmission but not N3 such that, this problem matches with the Receiver Collisions problem. It goes to a level of threat only when the distance between N1 and N2 is less than the distance between N2 and N3. The 2ACK scheme
does not suffer from limited transmission power problem.

Limited Overhearing Range: In order to transmit data to N3 a well-behaved N2 could apply low level of power transmission. Due to N1's limited overhearing range, it will not overhear the transmission successfully and will thus infer that N2 is misbehaving, causing a false alarm. Both this problem occur due to the potential asymmetry between the communication links. The 2ACK scheme is not affected by limited overhearing range problem.

C. Implementation of The 2ACK-Scheme Algorithm

The triplet N1→N2→N3 is used as an example to illustrate 2ACK code. Please Note one thing that such codes execute on each of the sender/receiver of the 2ack packets. Data Packet Sender Side Algorithm (Node N1) Find all the active(non idle) nodes. Find all routes between the active nodes. Choose the destination node(or Source's terminal node. Get the path to destination from the routing table. If Yes then send the data packet to destination. If No - Destination
is unreachable. After sending the Data packet Start timer and wait for the ACK (acknowledgment) of the third node in the opposite path as sent. After the timer expired check for the ACK (acknowledgment). If the ACK is not come then corresponding node attached to that link declared as misbehaving node.

2ACK Packet Sender Side (Node N3):-

1: Cpkts—>0 , Cak—>0, i, n //Initialization at node N3
2: while true do
3: if (data packet received) then
4: Cpkts ++ //Increase the counter of received packets
5: if (Cack/*Cpkts* <Rack) //The data packet needs to be acknowledged
6: prepare 2ACK
7: send 2ACK
8: Cack ++, i - - //Increase the counter of acknowledged packets
9: end
10: end
11:end

Receiver (Observer) Side(Node N1

Parallel process 1 (receiving hn)
1: while true do
2: if receive hn , from the 2ACK packet sender then
3: record hn, i , n //authentication for the packet it already done in the Ns2
4: end
5: end

Parallel process 2(receiving 2ACK packets)

6: while true do
7: randomly select (*Tstart* > *Currenttime*) //Start the observation
8: while (*Currenttime* < *Tstart*) do,
9: null
10: end
11: LIST —>Ø ,*Cpkts* —> 0, Cmis —> 0 //Initialization at node N1
12: while (*currenttime* < *Tstart* + *Tobs*) // Observation period is not expired
13: if (data packet forwarded) then
14: LIST <— LISTS U *dataID //Add a dataID to LIST*
15: Cpkts ++ // Increase the counter of forwarded packets
16: setup timer (t) for data ID //Record the time
17: end
18: if (2ACK packet received) then
19: search data ID carried by 2ACK from LIST
20: if (found) then // A 2ACK packet for a data ID received
21: check validity of hi
22: LIST <—LIST - data ID //Remove data ID from LIST

23: clear timer for ID
24: end
25: end
26: if (timeout event happens) then // 2ACK packet for a data ID is not received
27: LIST <— LIST-data ID // Remove data ID from LIST
28: Cmis ++; //Increase misbehavior counter
29 end
30 (*Cmis/Cpkts* > *Rmis*) then observation period expires
31 send link misbehavior report
32 end
33 end .

The process starts with route establishment procedure through the exchange of HELLO and TC control messages. Each node participating in the network functionality maintains a routing table. Using this routing table each node finds the path to reach all others destination nodes in the network. Consider a two hop path of N1*!*N2*!*N3. Let N1 be the source and N3 be the destination. Each node consists of a client and a server and the function of the server is to forward the data packets it receives from the same nodes client to the next hop on the path. When N1 wants to send a message to N3, the client of N1 forwards the message to the server of N1 which then forwards the message to the client of N2.Simultaneously N1 also starts a timer. The process continues till message reaches client of N3. After receiving the message the node N3 sends a ACK packet to N1.The function of N2 is to forward the packet to N1. On receiving the 2ACK packet N1 checks the status of the timer started for the packet sent to N3. If N1 receives the packet from N3 after expiry of the timer or it does not receive the packet at all then it will report Node N3 as a malicious node. The triplet N1→N2→N3 is used for reducing the network complexity and time. If we take more no of hopes then it will take more time. So we just consider the triplet for detecting misbehavior in MANETs. In case of misbehavior server doesn't forward the 2ACK packet to the next hop client. When N1's client does not receive a 2ACK packet, it increment parameter Cmis which keeps count of the number of packets miss. At the end of the process ratio Cmis /Cpkts is calculated. If the value is greater than Rmiss then the link is considered to be misbehaving and the information is displayed at the source node.

V. SIMULATION AND RESULTS

A. simulation

The process starts with route establishment procedure through the exchange of HELLO and TC control messages. Each node participating in the network functionality maintains a routing table. Using this routing table each node finds the path to reach all others destination nodes in the network. Consider a two hop path of N1→N2→N3. Let N1 be the source and N3 be the destination. Each node consists of a client and a server and the function of the server is to forward the data packets it receives from the same nodes client to the next hop on the path. When N1 wants to send a message to N3, the client of N1 forwards the message to the server of N1 which then forwards the message to the client of N2. Simultaneously N1 also starts a timer. The process continues till message reaches to client of N3. After receiving the message the node N3 sends a ACK packet to N1. The function of N2 is to forward the packet to N1. On receiving the 2ACK packet N1 checks the status of the timer started for the packet sent to N3. If N1 receives the packet from N3 after expiry of the timer or it does not receive the packet at all then it will report Node N3 as a malicious node. In case of misbehavior server doesn't forward the 2ACK packet to the next hop client. When N1's client does not receive a 2ACK packet, it increment parameter Cmis which keeps count of the number of packets missing. At the end of the process ratio Cmis /Cpkts is calculated. If the value is greater than Rmiss then the link is considered to be misbehaving and the information is displayed at the source node.

B. Assumptions and dependencies

The following suppositions are taken while building the system: The nodes of the network are randomly scattered over whole network and every node's location is independent of other nodes' locations. The source and the destination of each
transaction are chosen randomly among all nodes and nodes (other than the source and the destination) are selected as misbehaving nodes, separately. It is supposed that communication is bidirectional, such that symmetry of links is needed for the transmission of the designed 2ack packets. The routing misbehavior is thought in connection with OLSR protocol. It is assumed that no collusion among misbehaving nodes takes place. The misbehavior because of selfishness in most of the case is restricted to each node within MANETs.

C. Results

For this research work we have used 15-nodes topology, having bidirectional links. We are simulating the results in NS-2. We have compared the performance of plain OLSR and the OLSR+2ACK scheme. we kept misbehavior ratio parameter constant and compared it with other parameters. The graphs shows that we are improving the performance of the network by using 2ACK Scheme and the routing protocol as OLSR. We are detecting the misbehavior in the link for that just considering the triplet only. Following graphs shows delay vs misbehavior ratio, packet delivery ratio vs misbehavior ratio ,network overload vs misbehavior ratio and throughput vs misbehavior ratio:- The graphs shows the performance of OLSR which is improved by applying 2ACK-Scheme. We are also improving the packet delivery ratio by applying 2ACKScheme to OLSR. We are increasing the packet delivery ratio by using 2ACK with OLSR scheme. The network overload is reduced by applying the 2ACK-Scheme to OLSR protocol. We can compare the performance i.e. improved by this scheme. We are getting the higher throughput by applying 2ACK-Scheme to OLSR. We can compare the throughput by using the normal OLSR and throughput by using 2ACK+OLSR.

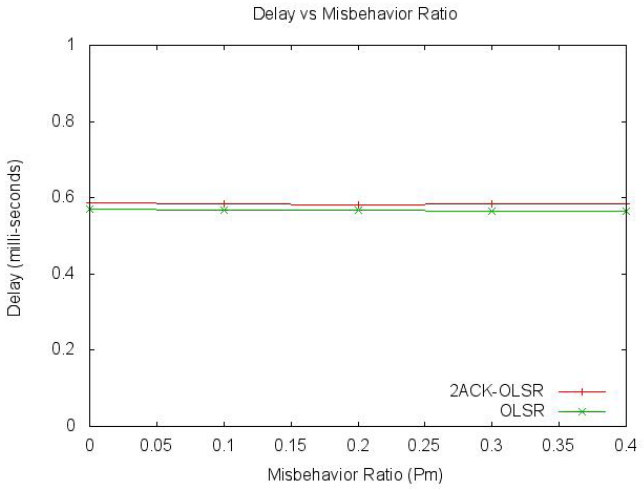

Fig: 5.1 Delay vs Misbehavior ratio

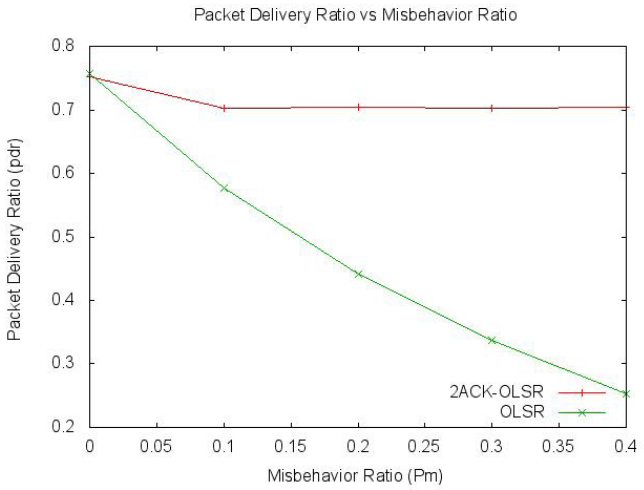

Fig: 5.2 Packet Delivery ratio vs Misbehavior ratio

VII. CONCLUSION

The proposed system is a simulation of the algorithm that detects misbehaving links in Mobile Ad Hoc Networks. The system implements the 2ACK scheme which helps to detect misbehavior by a 2 hop acknowledgment. The 2Ack scheme for detecting routing misbehavior is considered to be network-layer technique for mitigating the routing effects. The 2Ack scheme identified misbehavior in routing by using a new acknowledgment packet, called 2ACK packet. A 2ACK

packet is allotted to a fixed route of two hops (three nodes N1, N2, N3), in the direction opposite to the data traffic For routing purpose we have used OLSR protocol. The more troublesome task is to determine the characteristics of a single node. The main reason behind this is the communication is only between two nodes, and is not any node's sole act. So, any nodes associated with the misbehaving links should be punished carefully. In case of a link misbehaving, then any one of the two co-related nodes may be misbehaving in the association. In order to find the characteristics and punish a node, we should analyze the behavior of links around that node. A second case may arise wherein Node N1 floods Node N2 with packets thus causing N2 to drop packets. Hence it is imperative that before we declare a node to be selfish on the basis of the number of packets dropped, we should compute the ratio of number packets received against the number of packets dropped.

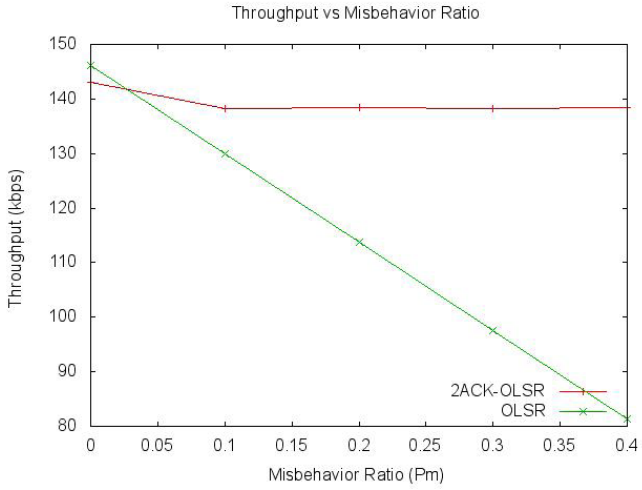

Fig: 5.3 Throughput vs Misbehavior ratio

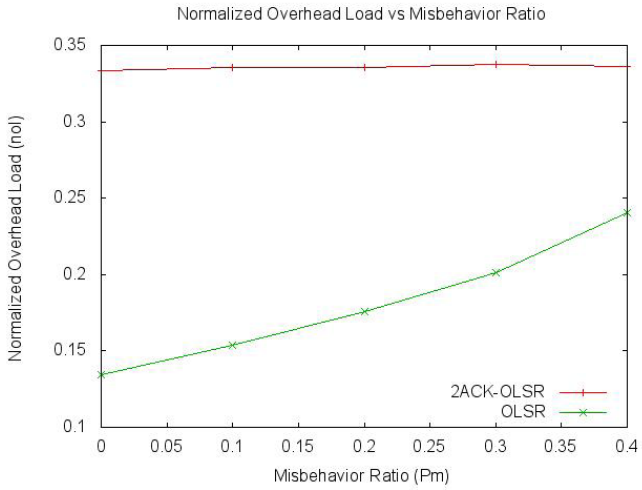

Fig: 5.4. Normalized Overload vs Misbehavior ratio

REFERENCES

[1] Aad I., Hubaux J.-P, "Denial of Service Resilience in Ad Hoc Networks",Proc. MobiCom, pp. 202-215, 2004.

[2] Baker M, Giuli T., Lai K. and Marti S, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks", *Proc. MobiCom, pp. 255-265, Aug. 2000*.

[3] Balakrishnan K., Deng J., and Varshney P. K, "TWOACK: Preventing Selfishness in Mobile Ad Hoc Networks", *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '05), pp.2137-2142 Mar. 2005*.

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

[5] Buttyan L.and Hubaux J.-P, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks", *Technical Report*, ACM/Kluwer Mobile Networks and Applications, vol. 8, no. 5, pp. 579-592, 2003.10

[6] Buttyan L. and Hubaux J.-P, "Enforcing Service Availability in Mobile Ad-Hoc WANs," Proc. MobiHoc", *pp, 255-265, Aug. 2000*.

[7] Buchegger S. and Le Boudec J.-Y, ”Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes, Fairness in Dynamic Ad-Hoc Networks”, Proc. MobiHoc, pp. 226-236, June 2002.

[8] Buttyan L., Hubaux J.-P.,and Jakobsson M, “A Micropayment Scheme Encouraging Collaboration in Multi-Hop Cellular Networks”, *Cryptography Conf., pp. 609-612, Jan. 2003*

[9] Chiasserini C.F., Nuggehalli P, *Srinivasan V., and Rao R.R.*, “Cooperation in Wireless Ad Hoc Networks”, Proc. INFOCOM, vol.2, pp. 808-817 Mar.-Apr. 2003.

[10] Conti M., Gregori E., and Maselli G, “Towards Reliable Forwarding for Ad Hoc Networks”, *Proc. Personal Wireless Comm. (PWC ’03), pp. 790-804, Sept. 2003*.

[11] Gross T., Hubaux J.-P., LeBoudec J.-Y., and Vetterli M, “Toward Self-Organized Mobile Ad Hoc Networks: The Terminodes Project”, *IEEE* Comm. Magazine, vol. 39, issue no.1, pp. 118-124, Jan. 2001.

[12] Feeney L.M. and Nilsson M, “Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment”, Proc. IEEE INFOCOM 2001.Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, Volume 3 (2001), pp. 1548-1557, 2007.