RESEARCHERID
THOMSON REUTERS

ISSN: 2277-9655
[Kaur * *et al.*, 6(12): December, 2017] Impact Factor: 4.116
IC™ Value: 3.00 CODEN: IJESS7

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

AN IMPROVED ACKNOWLEDGMENT SCHEME FOR DETECTING MISBEHAVING NODES IN MANETS

Rajveer Kaur[*1], Shaveta Rani[2], Paramjeet Singh[2]

[1*]Research Scholar, Department of Computer Science & Engineering, IKGPTU, Jalandhar, India.
[2]Professors, Department of Computer Science & Engineering, GZSCCET, Bathinda, India.

DOI: 10.5281/zenodo.1084326

ABSTRACT

In Mobile Ad-hoc NETwork (MANET), mobile devices communicate via wireless links without the aid of any fixed infrastructure. The need of corporation for routing and data forwarding, lack of central monitoring, wireless medium and energy constrained battery operated nodes make the MANETs susceptible to security threats. A Selfish node may refuse to forward the packets of other nodes to save its own resources or a malicious node may drop the packets to disrupt the network function. Such selfish or malicious nodes are named as misbehaving nodes. In this paper, we propose an acknowledgment based scheme I-ACK (Improved ACKnowledgment), to detect packet dropping attack by misbehaving nodes and prevent these nodes to be chosen for path establishment. It is improved version of existing scheme AACK. We simulate this scheme using NS2 and compare the results with TWOACK and AACK schemes. I-ACK gives better Packet Delivery Ratio and lesser Routing overhead as compared to above two schemes. It shows visibly improved performance for the longer routes.

KEYWORDS: Dynamic Source Routing Protocol, Packet dropping, Mobile Ad-hoc Networks, Misbehavior, Malicious, Selfish.

I. INTRODUCTION

A Mobile Ad hoc NETwork (MANET) is a collection of mobile devices connected through wireless links without any fixed infrastructure or central administration. Due to inherent features of MANETs like self-organizing and easy to deploy, they are being widely used in remote areas, campus networks, military, and other tactical applications such as emergency rescues. The increase of inexpensive, small and more powerful devices make MANET, a fastest emerging network [1].

In MANETs, mobile nodes which are within the radio range of each other can communicate directly, whereas the far away nodes depend on other nodes to communicate their messages. Thus, network operations like routing and packet forwarding depend on the cooperation between the nodes [2]. The nodes always trust whatever information they convey to each other. But this cooperation is very cost intensive, as it consumes network bandwidth, memory, energy and CPU time of a node. A mobile node may refuse to cooperate to save its own resources. Also, inbuilt features of MANETs like wireless medium, lack of central monitoring device and unpredictable dynamic topology make it vulnerable to different types of attacks. A non cooperative or a misbehaving node can easily cause a network failure without being detected.

In MANETs, a misbehaving node refers to a node that does not behave in a proper manner such as it may delay the packets, forward control packets while dropping data packets, or modify routing information. It deviates from its normal behavior which it is supposed to show according to a routing protocol [5].
Node misbehavior can be of three types [3][4]:

- Malfunctioned: A node malfunctions because of hardware and software problems, link breakage, or accidental physical damage.
- Selfish: A selfish node drops the data packets or may not participate in the routing functions in order to save its resources like battery life or CPU cycles. They use the services provided by other nodes but are reluctant to help others. It is passive misbehavior.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

- Malicious: This is active misbehavior. A node intentionally damages network and interrupts the network operations. A node participates in the routing process but may drop the data packets, fabricate the packet or impersonate other nodes with the intention to disrupt the network and affect its availability.

In this paper, we are proposing an Improved ACKnowledgment (I-ACK) scheme to detect packet dropping attack by misbehaving nodes in MANETs and avoid these nodes to be chosen for route establishment. The packet dropping attack can be black hole or Gray hole. A Black hole (attacker) drops all the packets instead of forwarding them. When a node drops the packets selectively instead of dropping all the packets e.g. packets of particular node, drops the packets after every fixed interval or drops the packets randomly then this is called Gray hole attack. The attacker or misbehaving node can be selfish or malicious.We chose Dynamic Source Routing (DSR) [7] protocol for our research work. It is a reactive and source routing protocol. Source routing means the source node provides the route in the packet header, to be followed by the data packet to the destination.

The rest of this paper is organized as follows. Section 2 provides the review of related work. Section 3 explains the proposed scheme, I-ACK, for detection and prevention packet dropping by misbehaving nodes. Section 4 presents simulation environment, performance metrics, results, and discussions. Finally, the paper is concluded in Section 5.

II. RELATED WORK

Researchers have proposed a number of schemes to detect and prevent the packet dropping due to node misbehavior in MANETs. As our scheme is acknowledgment based, this section briefly describes some of the previously proposed acknowledgment based schemes. In acknowledgment based schemes, nodes send special acknowledgment packets to ensure the successful acceptance of the data packets.

In TWOACK [8] scheme for every data packet received, nodes from a source to destination (except first two nodes) sends acknowledgment packet (TWOACK) back to a node which is two hops back on the source route. Suppose N1→N2→N3→N4→N5→N6 is source route. When N1 (source) sends a data packet to N2, N2 forwards it to N3 then N1 waits for the TWOACK acknowledgment packet, to get ensured that packet is successfully received by N3. The sets of every three consecutive nodes along the source route, follow the same procedure. If a node does not receive TWOACK packet for certain timeout period then it suspects next hop's forwarding link as misbehaving. After the threshold number of such attempts, the node informs the source that next hop's forwarding link is misbehaving.

The Selective TWOACK (S-TWOACK) [8] scheme is an enhanced version of TWOACK scheme. It reduces the routing load due to TWOACK packets. The nodes do not acknowledge every data packet received, they wait for a certain number of data packets (through the same set of three nodes) and then send an acknowledgment for multiple packets.

Like TWOACK scheme, 2ACK [9] scheme also uses two hops acknowledgment packets. It sends a 2ACK packet for the fraction of data packets instead of acknowledging every data packet received. To ensure the authenticity of 2ACK packets, they are digitally signed by the sender.

Adaptive ACKnowledgment (AACK) scheme [10] enhances the TWOACK scheme by working in two modes. By default, it works in AAck mode; it is end to end acknowledgment mode. In this mode, when a destination receives a data packet, it sends acknowledgment (AAck) packet back to the source to confirm the successful receipt of the data packet. If the source does not receive AAck packet within certain timeout threshold, nodes switch to TAck mode. AACK in TAck mode nodes work similar to TWOACK scheme but it detects the misbehaving node instead of detecting a misbehaving link. To detect the misbehaving node, the AACK [10] scheme classifies the nodes in the source route to three types: Source (S), Forwarder (F) and Destination (D) as shown in figure 1. Also, it is assumed that only intermediate nodes can be malicious.

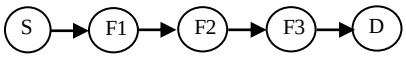


Figure 1. Source route containing Source(S), Forwarder (F) and Destination (D)

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Along the source route, there can be four possibilities of three adjacent nodes. They are:

- Case 1: Source-Forwarder-Destination (S-F1-D)
- Case 2: Forwarder-Forwarder-Destination (F1-F2-D)
- Case 3: Source-Forwarder-Forwarder (S-F1-F2)
- Case 4: Forwarder-Forwarder-Forwarder (F1-F2-F3)

According to TWOACK scheme if the first node in above cases (S or F), does not get TWOACK packet, it will report links F1→D, F2→D, F1→F2, and F1→F3 as malicious. According to AACK, which detects misbehaving node instead of a misbehaving link, in first two cases, S or F1 will report that node just before the destination is malicious (because destination cannot be malicious). In case 3, if F1 is malicious then S knows it, as it will not receive any acknowledgment from F2. If F2 is malicious then F1 will send an alarm to S. In case 4, F3 is reported as malicious by F1 because node F0 or S (the node just before node F1) is finding link F1→F2 works well, as it is getting acknowledgment from F2.

In Enhanced Adaptive ACKnowledgement (EAACK) [11] scheme, nodes work in three modes: ACK, S-ACK, and MRA (Misbehavior Report Authentication). The ACK mode is the default mode. In ACK mode, the destination sends end to end acknowledgments to the source for every data packet received. If the source node does not receive ACK packet within the desired time period, it sends S-ACK packet to the destination to switch the nodes along the route to S-ACK mode. In S-ACK mode, nodes send acknowledgment packet (S-ACK) two hops back along the source route similar to TWOACK scheme. If a node does not receive S-ACK packet within a predefined time, the node suspects next two nodes as malicious and send misbehavior report to the source. But unlike TWOACK, when a source node receives misbehavior report from a node, it does not mark a node as misbehaving instead, it switches to MRA mode. In this mode, the source sends MRA packet through a different route, to the destination to confirm whether misbehavior report is genuine or not. If it is genuine then reported nodes are misbehaving otherwise the reporting node is misbehaving.

EAACK2 [12] scheme is similar to EAACK scheme but in EAACK2, packets in S-ACK mode are signed with its digital signature. This ensures that the acknowledgment packets are genuine otherwise second node in the triplet can send acknowledgment packet pretending to be received from the third node.

In End-to-End ACKnowledgment (E2EACK) [14] scheme, when destination node receives the data packets, it sends ACK packet back to the source. Each intermediate node, for example, N2 in the route N1→N2→N3→N4, has to forward the ACK packet to the previous node (N1) in the route, to prove its honesty. If N2 wants to accuse N3 that it has not forwarded ACK to it, first N2 has to prove that it has correctly forwarded the data packet. For that, N2 asks its neighbors to send their digitally signed affidavit letters. If N1 does not receive the ACK of N4 or affidavit letters of N2 within some timeout period, N1 assumes that N2 is a malicious node.

Detecting misbehaving node instead of misbehaving link improves the packet delivery ratio of a network [20]. The authors explained this with the following example. Let's take a network of 20 nodes as shown in figure 2. They took two scenarios, one having misbehaving link detection scheme and the second with misbehaving node detection. Let's take node 10 as misbehaving. The node 2 has established route [2,6,10,14,18] to send data packets to node 18.

For the first scenario, as the node 10 drops the packets, the node 6 will report link 10→14 as misbehaving. Now source node finds a new route which does not contain link 10→14, (routes like [2,6,10,13,18], [2,6,10,15,18]) as shown in figure 2. But as node 10 is part of a number of links equal to its neighbors (10→13, 10→9, 10→5 etc.), it will still drop the data packets through these links until these links are detected. For the second scenario, when node 10 is declared as misbehaving, the nodes in the network will not use the routes having node 10. Thus, schemes with misbehaving node detection give better performance than schemes with misbehaving link detection.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

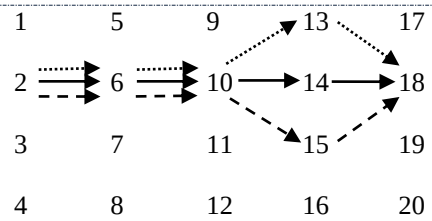


Figure 2. Different routes from source node 2 to node 18 via node 10.

III. PROPOSED SCHEME

In this section we propose our scheme Improved ACKnowledgment (I-ACK), it is improvement of previously proposed scheme AACK. In our scheme, we assume that links between the nodes in the network are bidirectional. Such type links are required for transmission of acknowledgement packets. We also assume that destination cannot be misbehaving.

Improved ACKnowledgment (I-ACK) scheme works in two modes: (i) Default mode (ii) I-TACK (Improved TWOACK) mode. In default mode, the source node asks intermediate nodes to send acknowledgments to confirm about the data packets received. I-TACK mode is similar to TWOACK scheme but it works improved way. Unlike TWOACK where all the nodes (except first two nodes in source route) from source to destination send TWOACK packets, I-TACK has lesser number of nodes sending two hop acknowledgment packets,

Data packets contain a unique sequence number which is incremented by one for every next data packet sent by the source. It tells the number of data packets sent by the source to the same destination. Every node keeps a counter for each source-destination (SD) pair and increments this counter on receiving a data packet for same SD pair. It tells a number of data packets received by a node for a particular SD pair.

Suppose we have source route N1→N2→N3…→N8. In default mode, the source N1 sends the data packet which contains the address of a random intermediate node (say N4). The node N4 has to send positive ACKnowledgement (ACK) or (NACK) back to the source node. The node N4 calculates R, it is the ratio number of data packets received by the node and a number of data packets sent by the source. If the node finds, R is less than some predefined threshold *Rthreshold*, it will send NACK (Negative ACKnowledgement) otherwise it will send ACK to the source.

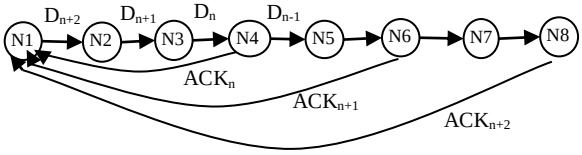


Figure 3. Working of nodes in default mode of I-ACK. D_{n-1}, D_n, D_{n+1} are data packets sent by source N1 to destination N8. D_n contains address of N4. D_{n+1} contains address of N6. D_{n+2} contains address of N8. ACK_n is an acknowledgment for D_n.

If the node sends ACK, then the source node asks for an acknowledgment from another intermediate node (say N6) between the current node (N4 in the example) and the destination node in the path. Again, the node sends ACK or NACK after finding the value of R. The source node keeps on asking for an acknowledgment from intermediate nodes till it reaches the destination. After receiving an acknowledgment from the destination node, the source starts again, by asking for an acknowledgment from a random intermediate node between source and destination. In this way, nodes keep on working in the default mode until the source receives NACK (as shown in figure 3). Figure 4 shows the flow of the working of nodes in the default mode of I-ACK.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

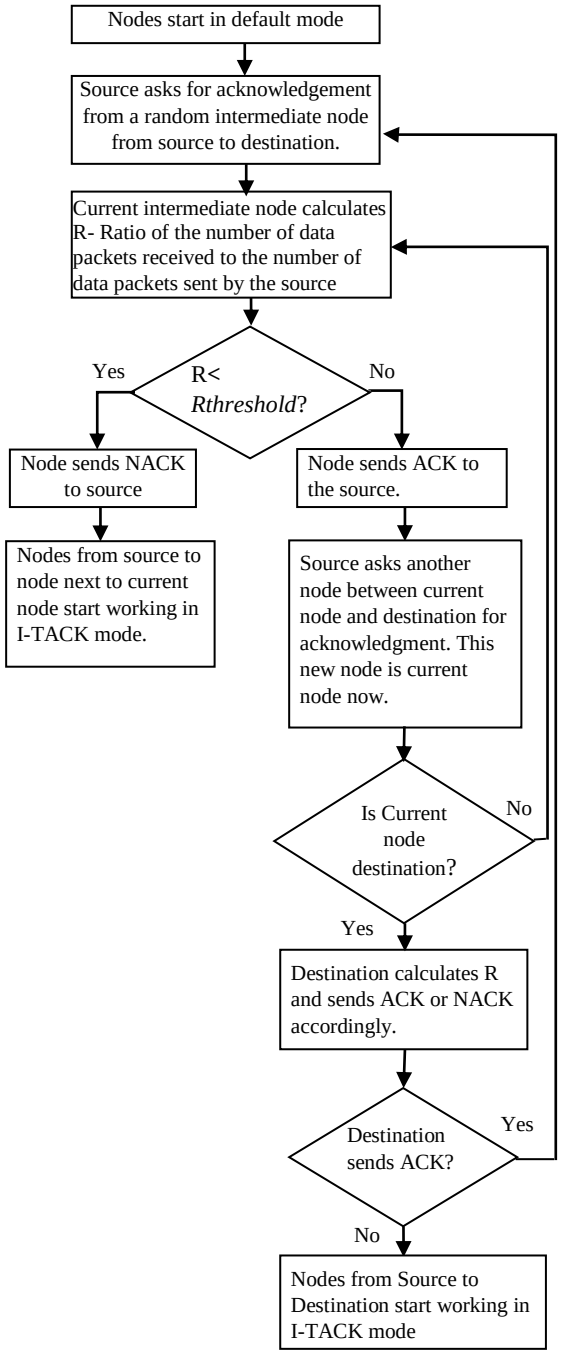


Figure 4. Flow of working of nodes in default mode

If a node (for example N4) sends a NACK packet, each node from source to node next to the node that sends NACK (N1 to N5), starts working in I-TACK mode as shown in figure 5. In this mode, similar to TWOACK, the nodes send two hops acknowledgment packets (I-TACK packet), back on the source route for each data packet received, but the difference is in the number of nodes sending two hop acknowledgments. In the TWOACK scheme, each node from source to destination (except first two nodes in source route) sends TWOACK packets but in I-TACK only the nodes from the source to the node next to the node (except first two nodes in the source route) that sent NACK, will send I-TACK packets. Thus, lesser number of nodes are involved in sending acknowledgment packets.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

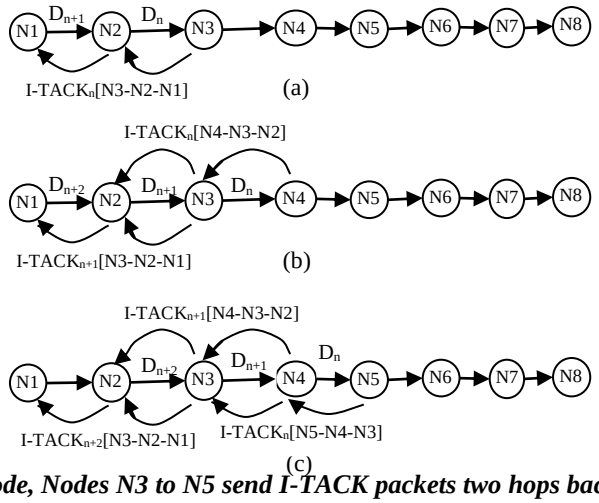


Figure 5. In I-TACK mode, Nodes N3 to N5 send I-TACK packets two hops back to acknowledge the receipt of every data packet (a) Step1 (b) Step2 (c) Step3.

Nodes have a counter *Mcounter* to count the number of missed I-TACK packets. If a node does not receive I-TACK packet within the certain predefined time period, it suspects next two nodes in the route and increments the *Mcounter* by one. If the value of *Mcounter* reaches the threshold value (*Mthreshold*), the node is sure that one of the nodes from next two nodes is misbehaving. To find which of these nodes is misbehaving; our scheme uses the same but improved technique as used in AACK. According to the AACK scheme, there can be four possibilities of three adjacent nodes, along with the source route. They are:

- Case 1: Source-Forwarder-Destination (S-F1-D)
- Case 2: Forwarder-Forwarder-Destination (F1-F2-D)
- Case 3: Source-Forwarder-Forwarder (S-F1-F2)
- Case 4: Forwarder-Forwarder-Forwarder (F1-F2-F3)

I-ACK works similar to AACK for first three cases. In first two cases, S or F1 will report that node just before the destination is misbehaving (because destination cannot be malicious). In case 3, if F1 is malicious then S will know it, as it will not receive any acknowledgment from F2. If F2 is misbehaving, then F1 will send an alarm to S. But for the fourth case, I-ACK works differently. For case four where three consecutive nodes are F1-F2-F3 (forwarder-forwarder-forwarder), the AACK scheme reports F3 as malicious (explained in section related work). But here F2 can also be malicious. For example, F2 is dropping the packets and also sending an acknowledgment to F0 (node just before F1), so F0 is thinking that nodes F1 and F2 are working well. But as F3 is not receiving data packets so it will not send acknowledgment packets to F1. Hence, F2 is malicious but AACK is reporting F3 as malicious.

Our scheme has overcome this problem of case four (F1-F2-F3). If a node (F1), does not receive I-TACK packets and its *Mcounter* reaches threshold value *Mthreshold* then F1 will find which node from next two nodes is misbehaving. If the first node out of three consecutive nodes (F1) is not a source or the third node (F3) is not a destination than the node starts working in promiscuous mode and observes the behavior of next node (F2). If F2 is forwarding the same traffic to the F3 as sent by F1 then F3 is misbehaving otherwise F2 is misbehaving. Then node F1 reports the source about the misbehaving node. The source will find a new route which does not include the misbehaving node.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the I-ACK scheme in comparison to existing TWOACK and AACK (with misbehaving node detection) schemes by varying the number of misbehaving nodes. We present simulation environment, scenarios, describe the performance metrics and simulation results.

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Simulation Environment

The simulations are carried on NS2.35 simulator running on Ubuntu 12.04 [16] with flat areas of 1000 x 1000m^2 and 800 x 800 m^2 and the number of nodes is varied as 50 and 100 to check the scalability of the schemes. User datagram protocol (UDP) Constant Bit Rate (CBR) traffic is used with packet size 512 bytes and rate 4 packets per second. Each simulation is run for 500 seconds. Each value of the results is obtained by taking the average of values taken by running the simulations three times with the different seed values. With increments of 10%, numbers of misbehaving nodes are varied from 0% to 40%. Various simulation parameters and their values are summarized in Table 1.

Table 1. Simulation Parameters

Parameter	Value
Simulation area	800 x800m2 & 1000 x 1000 m2
Simulation time	500 seconds
Number of Nodes	50 & 100
Mobility model	Random waypoint
Pause time	0 second
Speed	1m/s (Low speed) 20 m/s (High speed)
Traffic type	CBR
Packet size	512 Bytes
Routing Protocol	DSR
MAC protocol	CSMA/CA (IEEE 802.11)
Radio range	250 m

We are taking three simulation scenarios. In Scenario 1, the area is 800x 800 m^2, the speed of mobile nodes is 1m/sec, and nodes are 50. In scenario 2, the area is 800x 800 m^2, the speed of mobile nodes is 20m/s and nodes are 50 and in Scenario 3, the area is 1000x 1000 m^{2,} the speed of mobile nodes is 20m/s and nodes are 100

Performance Metrics

To evaluate the performance of the proposed scheme we have chosen the following parameters:

- Packet Delivery Ratio (PDR) is defined as the ratio of a number of packets received at the destination to the number of packets sent by the source.

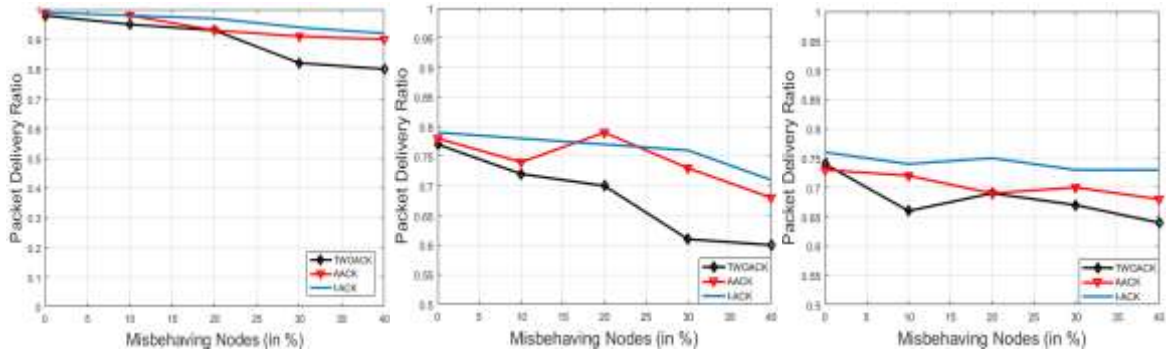
$$PDR = \frac{\sum packets\ received}{\sum packets\ sent} \quad (1)$$

- Normalized Routing Overhead (NRO) is defined as a number of routing packets transmitted per data packet delivered at a destination that is the ratio of the total routing-related control packet (RREQ, RREP, Route ERRor, ACK, NACK, and I-TACK) to the total data packets. Both forwarded and transmitted packets are counted.

$$RO = \frac{\sum Control\ packets}{\sum Data\ packets} \quad (2)$$

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

Results and Discussion

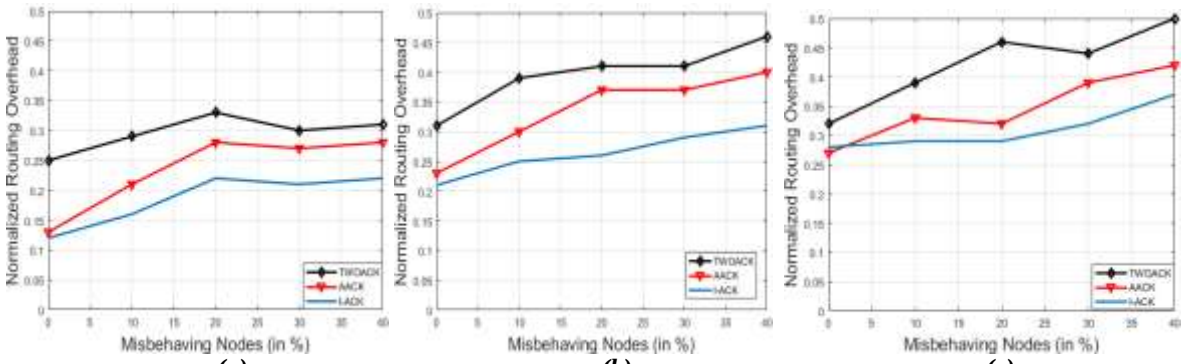


(a) *(b)* *(c)*

Figure 6. Packet Delivery Ratios for (a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Figure 6(a) shows Packet Delivery Ratio (PDR) of mobile nodes moving with the low speed of 1 m/sec. In figures 6(b) and 6(c), overall PDR for three schemes decreases due to high mobility (20m/sec) of mobile nodes. At high mobility link failure are more and routes in cache become stale.

Figure 6 shows I-ACK outperforms in packet delivery ratio for all the three scenarios. AACK and I-ACK are showing better PDR than TWOACK because they detect misbehaving nodes whereas TWOACK detects misbehaving links. In scenario 3, network area is larger, nodes are spread far apart, and source routes are longer, (having more number of hops) as compared to scenario 1 and scenario 2. Figure 6(c) shows, even for longer routes I-ACK has better PDR than AACK because in AACK there is an end to end acknowledgment in default mode, the source node comes to know about packet dropping very late, so it gives misbehaving nodes more time to keep on dropping before they are detected.



(a) *(b)* *(c)*

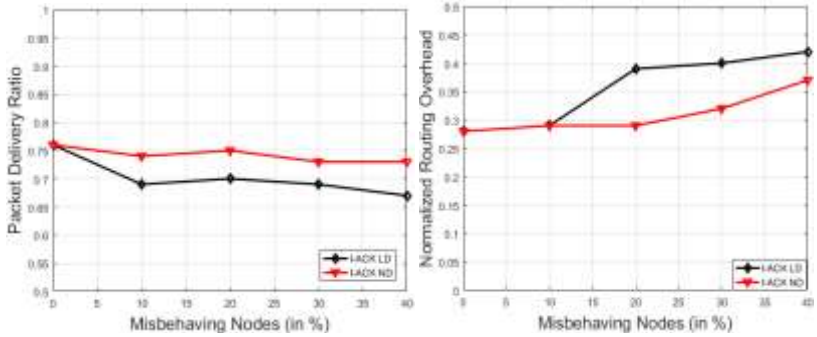
Figure 7. Normalized Routing Overhead for (a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Figures 7(a) and 7(b) show that, with an increase in speed, routing overhead of three schemes increases because with an increase in mobility of nodes, network topology changes frequently and link break occurs, thereby a number of control packets for establishing a new route increase, which leads to more routing overhead. Also with the increase in misbehaving nodes routing overhead increases as there is an increase in a number of control packets and acknowledgment packets. I-ACK and AACK have lesser routing overhead than TWOACK because in default mode they do not have two hop acknowledgment packets. In I-ACK and AACK schemes, for the routes longer than two hops, for each one hop increment in the route length, acknowledgment packet overhead is reduced by one for each data packet. For example, if the route length is three hops, I-ACK and AACK have one lesser acknowledgment packet than TWOACK, for each data packet sent. Similarly, for routes of length four, TWOACK has three two hop acknowledgment packets for each data packet sent. But I-ACK and AACK have one acknowledgment packet for each data packet. And also, even when there is no misbehaving node, TWOACK is having large routing overhead due to two hop acknowledgment packets.

Routing overhead of I-ACK is lower than AACK because in AACK scheme all the nodes in the source route (except first two nodes) send two hop acknowledgment packets in TAck mode but in our scheme lesser number of nodes (from the third node in the route to node, next to the node which sends NACK) are involved in sending

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

two hop acknowledgment packets. Hence a lesser number of acknowledgment packets. Figure 7(c) shows that I-ACK outperforms in routing overhead for the longer routes also. For longer routes as in scenario 3, reduction in the number of nodes sending two hop acknowledgment packets is more.



(a) *(b)*

Figure 8. (a) Comparison of Packet Delivery Ratio of I-ACK LD and I-ACK ND. (b)Comparison of Normalized Routing Overhead of I-ACK LD and I-ACK ND.

Figures 8(a) and 8(b) show the comparison of I-ACK with misbehaving Link Detection (LD) and I-ACK with misbehaving Node Detection (ND) for PDR and NRO respectively (for scenario 3). It is clear that I-ACK ND gives better performance than I-ACK LD. I-ACK ND detects the exact misbehaving node but I-ACK LD detects misbehaving link giving the misbehaving node more chances to drop through other links.

V. CONCLUSIONS AND FUTURE WORKS

Our proposed scheme I-ACK has detected the misbehaving nodes and has prevented the packet dropping by avoiding these nodes to be chosen while establishing the paths. We evaluated and compared the performance of our scheme with existing schemes TWOACK and AACK. The simulations are done with slow and high-speed moving nodes and for different areas by varying the number of misbehaving nodes. The Results show visible improvement in Packet Delivery Ratio and Normalized Routing Overhead. Our scheme works much better for larger areas where there are longer routes. It also shown by the results that detecting misbehaving nodes instead of misbehaving links gives better network performance. In this research, we will work to further reduce the routing overhead due to acknowledgment packets. Another issue of research is authentication of acknowledgment packets, as acknowledgment packets are vulnerable to be forged.

VI. REFERENCES

[1]. S. K. Sarkar, T.G. Basavaraju and C. Puttamadappa, "Ad Hoc Mobile Wireless Networks", Auerbach Publishers, New York, USA, pp.23-24, 2008.
[2]. R. Hekmat, "Ad-hoc Networks: Fundamental Properties and Network Topologies", Springer, pp.3-5, 2006.
[3]. Q. Li and G. Cao, "Mitigating Routing Misbehavior in Disruption Tolerant Networks", IEEE Transactions on Info. Forensics and Security, vol.7, no. 2, pp. 664-675, April 2012,
[4]. H. M Deng, W. Li and D. P. Aggrawal, "Routing Security in Wireless Ad hoc Networks", IEEE Communication Magazine, vol.40, no. 10, pp.70–75, 2002,
[5]. Rasika Mali and Sudhir Bagade, "Techniques for Detection of Misbehaving Nodes in MANET: A Study," Int Journal of Scientific & Engineering Research, vol. 6, no. 8, August 2015.
[6]. D. Soufiene, F. Naıt-abdesselam and Z. Zhang, "Mitigating Packet Dropping Problem in Mobile Ad Hoc Networks: Proposals and Challenges", IEEE Communications Surveys & Tutorials, vol.13, no.4, pp.658-672, 2011.
[7]. D.B. Johnson, D. A. Maltz and Y.C. Hu, "The Dynamic Source Routing Protocol for Mobile Ad hoc Networks", 2004, [online] Available: http://tools.ietf.org/pdf/draft-ietf-manet-dsr-10.pdf .
[8]. K. Balakrishnan, J. Deng and P. K. Varshney, "TWOACK: Preventing Selfishness in Mobile Ad Hoc Networks", in proc. of IEEE Wireless Communication and Networking Conf., Mar.2005, pp. 2137 – 2142.
[9]. K. Liu, J. Deng, P.K. Varshney and K. Balakrishnan, "An Acknowledgment-Based Approach for the Detection of Routing Misbehaviour in MANETs", IEEE Transactions on Mobile Computing, vol. 6, no.5, pp.536-550, 2007.

[10]. Al-Roubaiey, T. Sheltami, A. Mahmoud, E. Shakshuki and H. Mouftah, "AACK: Adaptive Acknowledgment Intrusion Detection for MANET with Node Detection Enhancement," in proc. 24th IEEE International Conference on Advanced Information Networking and Applications, pp 634-640, April 2010, doi: 10.1109/AINA.2010.136.
[11]. N. Kang, E. M. Shakshuki, and T. R. Sheltami, "Detecting Misbehaving Nodes in MANETs", in the proc. of 12th Int Conf on Information Integration and Web-based Applications & Services (iiWAS2010), Nov.2010, pp. 216-222.
[12]. N. Kang, E. M. Shakshuki and T.R. Sheltami, "Detecting Forged Acknowledgements in MANETs", in the Proc of Int Conf on Advanced Information Networking and Applications, Mar. 2011, pp.488-492.
[13]. E. M. Shakshuki and T. R. Sheltami, "EAACK-A Secure Intrusion-Detection System for MANETs," IEEE Transactions on Industrial Electronics, vol.60, no.3, pp.1089 –1098, Mar.2013,
[14]. Heydari and S. Yoo, "E2EACK: an End-to-End Acknowledgment-Based Scheme against Collusion Black hole and Slander Attacks in MANETs", Wireless Networks, vol.22, no.7, pp.2259-2273, 2016.
[15]. H. M. Sun, C. H. Chen and Y. F. Ku, "A Novel Acknowledgment-Based Approach Against Collude Attacks in MANET", Expert Systems with Applications, vol. 39, no.9, 2012, pp. 7968–7975.
[16]. The Network Simulator (NS2), [online] Available: http://www.isi.edu/nsnam/ns/.