ANALYSIS AND AVOIDANCE OF JELLYFISH ATTACK

[1]Devanshu Mishra, [2]Anuradha More, [3]Priyanka Maru, [4]Anish Dandekar, [5]Nayana Vaity.
[1,2,3,4,5] Department of Computer Science.
Terna Engineering College, Navi Mumbai.
[1]mishradevanshu95@gmail.com, [2]more.anuradha18@gmail.com,
[3]marupriyanka288@gmail.com, [4]dandekaranish@gmail.com, [5]nayana.vaity17@gmail.com

***Abstract*- MANET can be defined as a collection of mobile nodes wirelessly connected which has no infrastructure and can be self-configured. Due to its various characteristics like Dynamic topology, no centralized controller, no infrastructure, security limitations, energy constraint there are a few drawbacks and the major one is the security concern. If the limited security problem are not solved MANET will becomes prone to many attacks. Various protocols are available to overcome these drawback and one of the protocols is Adhoc on-demand Distance vector routing protocol (AODV) which is popular because of its reactive nature. Attacks can take place if any mobile node behaves maliciously which drops, delays or reorders the packets exchanged between intermediate mobile nodes during communication and such nodes are called jellyfish attacking nodes. The limited security problem might increase if these maliciously behaving mobile nodes are not handled properly. Thus, In our paper we are using trust based parameters and perceptron logic in order to avoid such maliciously behaving nodes using NS2 as the simulator.**

Keyword: MANET, AODV, Jellyfish, reactive, trust.

I. INTRODUCTION

In the past few years, various technologies have been established and among them MANET is the most popular. A network that is characterized as infrastructure less, spontaneous, dynamic is called adhoc network or Mobile Adhoc wireless network (MANET).It is a network with requirement of less financial demand and thus is applicable in various areas. Some of them are military applications, Search and rescue operations, Disaster relief operations, law enforcement, Commercial use, wireless mesh. This type of network has a basic principle that nodes are free to join and leave the network that is topology is dynamic[1]. Despite of the various applications MANET, it has certain characteristics like dynamic topology, openness medium, decentralized administration due to which providing security has become the prime concern.

Due to the following reasons: MANET is vulnerable to many attacks. The first reason is no-central administration so it becomes difficult to detect maliciously behaving nodes. The second reason is dynamic topology which can lead to disturbance in the network as the nodes are mobile in nature and can come and go as and when required. Due to such behaviour it is easy for maliciously behaving nodes to stay hidden. The next reason being packet losses due to the mobile nature and node interference which dramatically increases packet loss and the drawback being power life of MANET as the nodes require more energy as and when transmission and reception of packets takes place. These are the various limitations which makes MANET prone to many attacks. This attack is relevant for open-loop flows that do not respond to congestion, loss, or delay information, and hence cannot be thwarted by JellyFish[2].

MANET being vulnerable to many attacks and many protocols are available to defend itself against these attacks and one of the most popular is Adhoc On demand Distance vector routing (AODV) protocol. AODV protocol is an extension of DSR protocol .It is a reactive protocol wherein the route is found only when there is a need for it to transfer packets from source to destination. This is done by using route request RREQ message as shown in Fig1, route reply RREP message and route error RRER message as shown in Fig2. Whenever any node stops working or moves out of the network then an RRER message is sent to the source to inform it about that particular node.

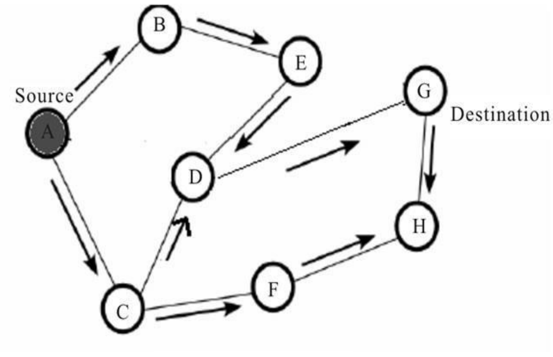

Fig1: RREQ Broadcast

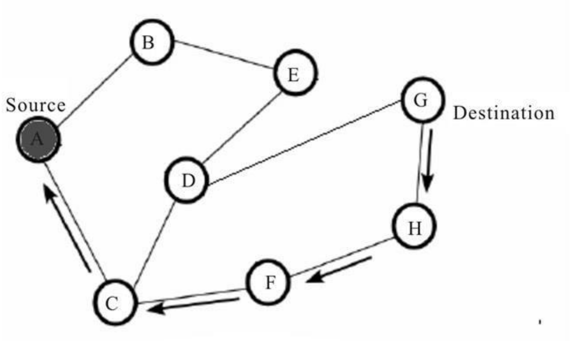

Fig 2: RREP Forwarded Path

II. JELLYFISH ATTACK

MANETs are adhoc networks that have a networking environment which is routable on top of a link layer adhoc network. There are two kinds of attacks namely: Active attack and passive attack. An active attack is an attack which is characterized as an attempt to break into the system. Examples of active attack are Denial of service (Dos), Spoofing, Ping flood, Black hole attack, Man-in-the-middle, Arp poisoning, etc. Passive attacks can be defined as attacks in which the attackers monitors or scans the system to find open ports and vulnerabilities. Unlike active attackers who cause physical harm to the system, Passive attackers observe the system and then attack. Thus, passive attacks are difficult to detect. Few examples of attacks which are passive are traffic analysis, network analysis, eavesdropping, etc.

JellyFish attack is defined as a sort of Denial of service (DoS) attack, in which there is delay before the transmission and reception of information .A mobile node launching JellyFish attack will be referred as “JF-node” further. JellyFish Attack is such that it behaves according to the protocols and thus is harder to detect. The a closed loop protocol such as TCP is mainly the target of the JF-node whose main aim is to exploit the working mechanism to degrade the communication performance. Analysis of effects of three JF attacks variants namely : (1)JF-reorder, (2) JF-delay,(3)JF-drop is done over TCP-SACK , TCP-Reno, TCP-new Reno, TCP Tahoe and amongst them TCP-SACK has been the most robust as compared to the others in terms of handling packet losses and retransmission timeouts[3]. JellyFish attack works in accordance with the data and control protocols to make itself difficult to detect as well as to prevent. An intermediate node can produce a critical vulnerability for TCP congestion control mechanism because there is no functional distinction among mobile nodes in MANET. Such a maliciously behaving node alters its forwarding behaviour as described in the following JF variants.

1. Jellyfish reordering attack:

In this variant of JellyFish attack, the JF-node, before forwarding the packets reorder them. Since acknowledgement of some reordered packets are not received in time those packets are retransmitted by the sender again. From the receiver’s point of view, it generates an acknowledgement each time a packet is received. In such a case where packets are reorders the sender might receive duplicate Acknowledgement. The JF node thus creates a buffer in its input queue. The attacker reorders the packets in this buffer before forwarding it.

2. Jellyfish periodic dropping attack

In this variant of JellyFish attack, during communication process, a JF node randomly drops some packets over a specified period of time. Thus, incorrect route congestion has taken place such information is conveyed to TCP. Thus the dropping of nodes is misunderstood as congestion in the route. The JF-node either chooses to discard a fraction of packets (Example:10 packets are dropped out of 1000 packets) or may discard all the packets received during that period of time(Example: discard data packets for some milliseconds, every second near the TCP sender timeout).Thus, the TCP is forced to enter the retransmission timeout (RTO) and increase its RTO. This leads to decrease in the throughput as the attacker increases the frequency of dropping of packets.

3. Jellyfish delay variance attack:

In this kind of TCP variant, the JF-node are delaying the packets selfishly which results in increase in the RTT, which misleads the sender TCP increasing the congestion window size and sends traffic in bursts. It will therefore, lead to more collisions.

III. LITERATURE SURVEY

Sr. No.	**Techniques / Solutions**	**Drawbacks**
01.	Checks the shared hops from RREP’s and maintains last packet sequence numbers that are sent and received	1. Time delay 2. Cannot detect JF nodes.
02.	Secured ETX metric (Expected Transmission Count)	1. Time delay 2. Overhead due to much calculation.

03.	Compares the RREP sequence numbers with threshold value using dynamic learning method	1. Time delay 2. Cannot detect JF nodes.
04.	Fidelity table based on the acknowledgements received by the source node.	1. Time delay
05.	Using SRREQ and SRREP based on the random numbers generation	1. Time delay 2. Cannot detect JF nodes. 3. Network overhead
06.	Compare RREPs and discards the high destination seq - number RREP.	1. Time delay 2. Cannot detect JF nodes.
07.	Enhance Route Discovery for AODV(ERDA)	1. Cannot detect JF nodes.
08.	Compares the RREP sequence numbers. with threshold value and selects the routes	1. Cannot detect JF nodes.
09.	IN node generates SREQ to the destination for fresh SN.	1. Time delay 2. Cannot detect JF nodes.
10.	Behavioural analysis filters and trust values.	1. Time delay 2. Cannot detect JF nodes. 3. Network overhead
11.	Feedback solution based on the no. of packets sent from the nodes	1. Always it doesn't works i.e. when congestion occurs 2. Cannot detect JF nodes.
12.	Using Prior_ReceiveReply method	1. Time delay 2. Cannot detect JF nodes.
13.	Checking SN's of source node and first route reply.	1. Time delay 2. Cannot detect JF nodes.
14.	Compares SN's of more than one RREP's at source node	1. Cannot detect JF nodes.

IV. PROPOSED METHODOLOGY

A. Referring to "Avoidance of black hole node"

$T = \tanh(1+R_1)$[4]

End to end delay (E) and throughput can be related in the following manner.

If Throughput increases then End to end delay decreases.

Similarly, Throughput decreases then End to end delay might have increase.

Let us say,

E=processing time +queuing time + latency + transmission time + propagation time.

For Jellyfish delay attack (Type 2) is introduced in a processing time …………………… .(1)

Let's have a constant "a" for all other factor of E……………………………………… (2)

From (1) and (2)

E=processing time + a +a + a+ a ……. (3)

Let's have D as delay factor which is introduced by jellyfish attack

Thus, our equation becomes,

$E = D*a + a + a + a + a$

$E = Da + 4a$

$E = a(D+4)$ …………………………….. (4)

Assuming all the factors affecting E takes unit time, our equation becomes

$E = D + 4$ ………………………………… (5)

For jellyfish node to introduce delay in the network requires more than unit time.

E= (>=2) + 4……………………………… (6)

Thus, our lower bound for E becomes,

E = 6 ……………………………………... (7)

Hence, our proposed equation is

$T = \tanh(1+R_1+E)$

$T = \tanh(1+R_1+6)$

$T = \tanh(7+R_1)$.

V. RESULTS

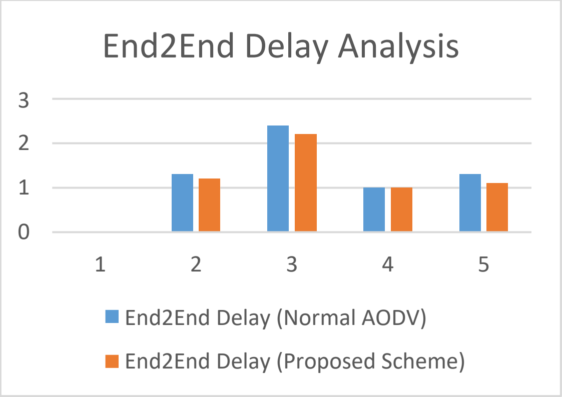

Fig3: End to End Delay Analysis

Fig1 shows the end to end delay comparison between normal AODV and proposed scheme which indicates slight improvement. Therefore end to end delay for proposed mechanism shows slight improvement.

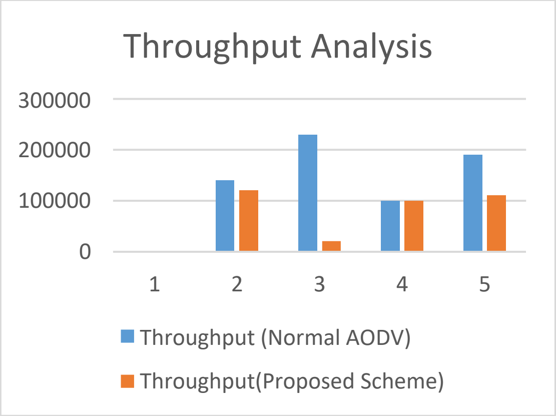

Fig4: Throughput Analysis

Fig2 shows throughput comparison between normal AODV and proposed scheme. We observed that proposed scheme shows improvement in throughput as compared to normal AODV.

CONCLUSION

As we know that MANET is prone to many attacks and thus we have suggested a methodology to overcome JellyFish attack using AODV protocol. For that we have studied and analysed the malicious behaviour of the JF mobile nodes. We observed that JF attack has three types of which we suggested methodology of avoiding two types of JF attacks (Type 1: Dropping of packets and Type 2: Delay variance). Trust counter is used to avoid both types of attacks. Improvement in AODV was observed by our proposed scheme.

REFERENCES

[1] E. Sam Prabhakar and Mr. K. Srinivasan, “An efficient detection and counter measure of a jellyfish attack using dmpd algorithm in manet”, Volume 21 Issue 4 pp 16-21 APRIL 2016.”

[2] Imad Aad, Jean-Pierre Hubaux, Senior Member, IEEE, and Edward W. Knightly, Senior Member, “Impact of Denial of Service Attacks on Ad Hoc Networks” IEEE, VOL. 16, NO. 4, AUGUST 2008, pg.719-802.

[3] Anjugam S and Muthupriya V,” Direct trust-based detection and recovery process of jellyfish attack in manet”, IJETCSE, Volume 22, pg.32-38 Issue 2 – MAY 2016.

[4] Lata Ragha, Jay Thakar and Jagruti Desai, “Avoidance of Co-operative black hole attack in AODV in MANET”, 15 September 2016.