Survey: Impact Of Jellyfish On Wireless Ad-Hoc Network

Ms Hetal P. Patel[*1], *Prof. Minubhai. B. Chaudhari*[#2]

** C.S.E. Department, Government College of Engineering, Gandhinagar
Gujarat Technology University, Gujarat, India.*

*# C.S.E. Department, Government College of Engineering, Gandhinagar
Gujarat Technology University, Gujarat, India*

Abstract

Recently, Wireless Ad Hoc networks become a hot research topic among researchers due to their flexibility and independence of network infrastructures. Wireless Ad Hoc networks are vulnerable to many attacks due to its unique characteristics such as open network architecture, stringent resource constraints, shared wireless media and highly dynamic topology. The attacks can be of different types out of which denial of service is one of the most difficult attacks in detail and Jellyfish attack in detail with its impact. Jellyfish attack is a new denial of service attack that exploits the end to end congestion control mechanism of TCP (Transmission Control Protocol). The main goal of the Jellyfish nodes is to reduce the goodput of all the flows by either reordering the packets or dropping a small fraction of packets.

Keywords: Wireless Ad hoc Networks, Security, Type of attack and Jellyfish Attacks.

1. Introduction

In recent years, wireless ad hoc networks (WANETs) have become very popular due to their wide range of applications and their ability to be deployed under normal and harsh conditions while supporting high data rates. So that's why Wireless Ad Hoc networks to become one of the fastest growing areas of research. This new type of self-deploying network may combine wireless communication with high degree node mobility.

Ad-hoc means "for a particular purpose without consideration of huge application". The Wireless ad-hoc network is a self-configuring infrastructure-less network of mobile devices connected by wireless links A Wireless Ad-hoc Network is a group of nodes without any existing infrastructure and forms a temporary network. These networks are used in emergency search, disaster management, electronic class rooms, military operations, conferences etc. An ad-hoc network does not contain any centralized administration. Since the nodes communicate with each other without any infrastructure, the connection establishment is done by forwarding packets over themselves. To support this connectivity, nodes use some routing protocols such as AODV (Ad-hoc On-Demand Distance Vector), DSR (Dynamic Source Routing) and DSDV (Destination- Sequenced Distance-Vector).

Security is a necessary need for both wired and wireless network communications [1]. Unlike wired networks, wireless networks pose a number of challenges to security solutions due to their unpredictable topology; wireless shared medium, heterogeneous resources and stringent resource constraints etc. There are a wide variety of attacks that target the weakness of this kind of network. In this type of network, security is not a single layer issue but a multilayered issue. There are five major security goals that need to be addressed in order to maintain a reliable and secure ad-hoc network environment. They are mainly:

- Confidentiality
- Availability
- Authentication
- Integrity
- Non-repudiation

The main characteristics of Wireless Ad-Hoc network such as dynamic topology, resource constraints, lack of network infrastructure or centralized administration, make it more susceptible to a number of attacks and thus the vulnerability in this networks will be more. As this technology is increasing day by day and will be widely used in the years to come, providing the security to this type of networks is a major issue. Denial of service attack is one of the major threats to the Wireless Ad-Hoc networks, in which Protocol-compliant Denial of Service attacks are the most difficult to defend against [2], Aad et al. refer to such attacks as Jellyfish attacks.

Our methodology is to study Denial of service resilience via a new and general class of *protocol compliant* denial-of-service attacks, which we refer to as *Jellyfish* (JF) [2]. Previously studied attackers *disobey* protocol rules; on the contrary, Jellyfish conform to all routing and forwarding protocol specifications, and moreover, as implied by the name, are passive and difficult to detect until after the "sting." Jellyfish target *closed-loop* flows that are responsive to network conditions such as delay and loss. [4] Examples include TCP flows and congestion-controlled UDP flows employing a TFRC-like algorithm.

2. An Overview of Security in Ad-Hoc Network

2.1 Issues in Ad Hoc Wireless Networks

Ad hoc networks come into some of the traditional problems of wireless communication and wireless networking is Follow:

- The wireless medium does not have proper boundaries outside of which nodes are known to be unable to receive network frames.
- The wireless channel is weak, unreliable, and unprotected from outside signals, which may cause lots of problems to the nodes in the network.
- The wireless channel has time-varying and asymmetric propagation properties.
- Hidden-node and exposed-node problems may occur.

2.2 Types of Security Attacks

The security attacks in wireless Ad-Hoc can be roughly classified into two major categories, namely passive attacks and active attacks are as described in the figure 1.The active attacks further divided according to the layers [1].

Attacks can be classified according to network protocol stacks. Figure 1 shows an example of classification of security attacks based on protocol stack. Some attacks could be launched at multiple layers also.

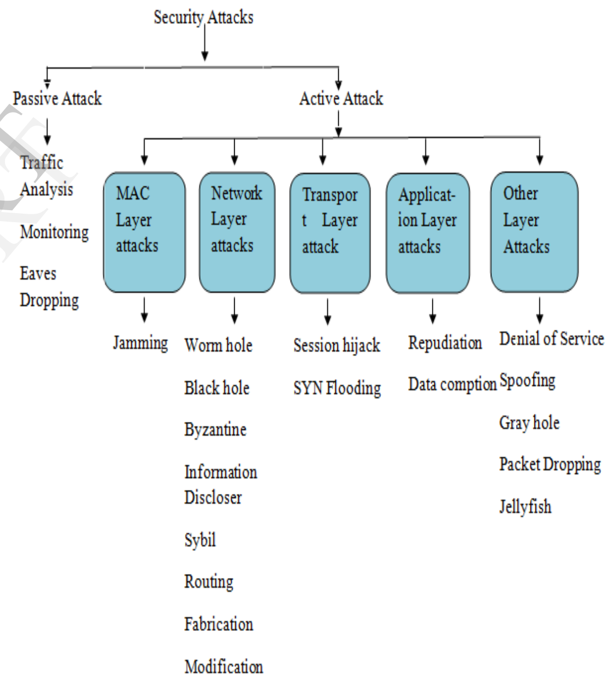

Fig. 1. Different types of attack.

Ad hoc networks are prone to different Denial-of-Service attacks because of its dynamic topology, remote location and services. The different types of Denial of service attacks in Adhoc networks are jamming, exhaustion and integration, selective forwarding, tampering, misdirection, sinkholes, Sybil, wormholes, and flooding [2]. A very common attack in wireless networks is Jellyfish attack. It targets TCP's congestion control mechanism. The main goal of the Jellyfish nodes is to reduce the goodput of all the flows to near-zero by either reordering the packets or dropping a small fraction of packets. [4] These forwarding mechanisms are variants of Jellyfish attack.

3. Jellyfish Attack

Significant progress has been made in securing ad hoc networks by developing secure routing protocols that ensure different security concepts such as authentication and data integrity. Moreover, intrusion detection and trust-based systems have been developed to protect MANETs against misbehaviors such as rushing attack, queryflood attacks, and selfish behaviors. Yet, most of the defense mechanisms are not able to detect a set of protocol compliant attacks called jellyfish (JF) attacks. [2]

When a malicious nodes launches forwarding rejection attacks it also may comply with all routing procedures. A malicious node launching Jellyfish attacks may keep active in both route discovering and packet forwarding in order to prevent it from detection and diagnosis, but the malicious node can attack the traffic The Jellyfish attack is especially harmful to TCP traffic in that supportive nodes can hardly differentiate these attacks from the network congestion. Reference also described that malicious nodes may even abuse directional antenna and dynamic power techniques to avoid upstream nodes to detect their misbehaviours of dropping packets. [3]

As shown in Figure-2, node JF is a Jellyfish, and node S starts to communicate with node D after a path via the Jellyfish node is established. Then the Denial of service attacks launched by node JF will cause packet loss and break off the communications between nodes S and D eventually. [3]

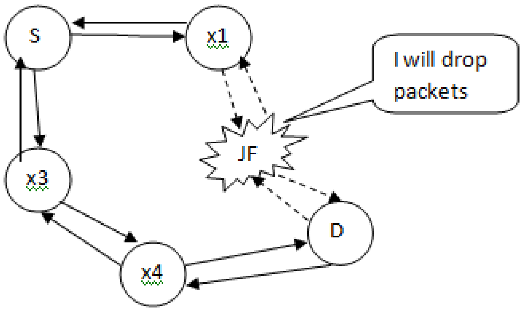

Fig. 2. Jellyfish attack scenario

Many of the attacks disobey the protocol rules, but the Jellyfish attack obeys all the protocol rules. The main strength of this attack is that it is compliance with all the data plane and control plane protocols, so that the detection and diagnosis of the attack becomes difficult and time consuming. [2]

This attack mainly targets closed-loop flows as such flows respond to network conditions like packet loss and packet delay. These attacks are passive and are difficult to detect.

In particular, many applications such as file transfer, messaging, and web will require reliable, congestion-controlled delivery as provided by protocols such as TCP. Moreover, TFRC-controlled real-time applications such as interactive video must also adapt their rates to available bandwidth and hence also employ end-to-end congestion control. The dual role of hosts as routers in ad hoc networks introduces a critical vulnerability for congestion control: specifically, there are a number of *forwarding* behaviors that routers (ad hoc relay nodes) can employ that will severely degrade the end-to-end throughput of congestion-controlled traffic. We refer to these behaviors as variants of the Jellyfish attack, which we describe as follows.

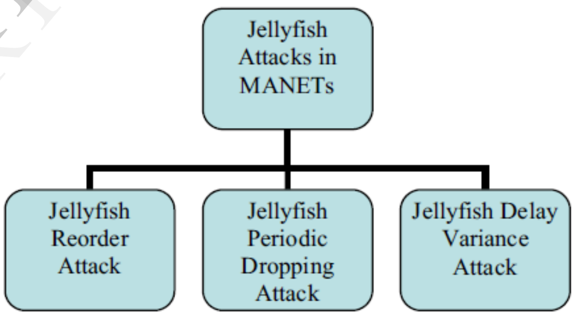
Jellyfish Attacks in MANETs
Jellyfish Reorder Attack
Jellyfish Periodic Dropping Attack
Jellyfish Delay Variance Attack

Fig. 3. Jellyfish Type

We next present simulation experiments that illustrate the effects of JF on end-to-end goodput. To study these effects in isolation, we consider a simple "chain" scenario with a sequence of nodes between the sender and receiver, one of which is a JF. We use TCP Sack, the default IEEE 802.11MAC at 2 Mb/s, and show the 95% confidence intervals over 10 simulation runs.

3.1 JF Reorder with Impact

TCP has a well-known vulnerability to reordered packets due to factors such as route changes or the use of multi-path routing, and a number of TCP modifications have been proposed to improve robustness to misordering [9]. However, *no* TCP variant is robust to *malicious* and persistent reordering as employed by the JF misordering attack.

TCP's use of cumulative acknowledgements defines the message "ACK-N" to indicate that *all* segments "1,...,N" have been received. Consequently, receipt of duplicate ACKs is used to infer loss. Yet, because duplicate ACKs can also indicate an out-of-order packet receipt, TCP has a number of mechanisms to increase its robustness to out-of order packets, including TCP Sack and reorder robust TCP. Yet, all such TCP variants assume that reordering events are rare, short-lived, and due to network events such as route changes. In contrast, we consider JF nodes to maliciously re-order packets. In this attack, JF deliver *all* packets, yet after placing them in a re-ordering buffer rather than a FIFO buffer. Consequently, we will show that such persistent re-ordering of packets will result in near zero goodput, despite having all transmitted packets delivered. [3, 12]

Fig. 4 shows the impact of the JF-reorder attack on the TCP-Sack flow for different re-ordering. This experiment has a scheduler that is a FIFO queue, except that it selects randomly among the first packets in the queue. The figure depicts performance as a function of the re-ordering buffer size expressed in packets.

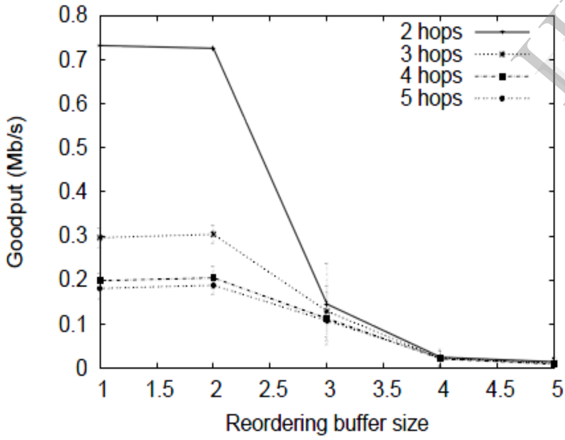

Fig. 4. JF-reorder effect on throughput [3, 12].

The figure indicates that TCP is robust to moderate reordering with a reordering buffer of 2 packets. Whereas, when the reordering buffer is larger and the reordering is performed in this persistent and malicious way, TCP throughput collapses. For example, consider the curve with 3 nodes and a 2-hop chain, i.e., a source, destination, and a single relay node. Without an attack (a reordering buffer of 1), the flow obtains a throughput of 710 kb/s. Yet, with a reordering buffer of 3 or more packets, the throughput decreases to approximately 1% of the peak value indicating a successful attack and near starvation of the flow. That is, if the scheduler selects the next packet to service randomly among the first 3 or more queued, the resulting reordering cannot be overcome by TCP. We note that solutions to TCP reordering such as TCP-PR use only timers to detect loss versus duplicate ACKs. Thus, attackers would need to either use other JF variants for TCP-PR flows or use larger reorder buffers to force TCP-PR timeouts.

3.2 JF Periodic Dropping with Impact.

This attack is inspired by the Shrew attack in which an endpoint sends maliciously spaced periodic pulses in order to force flows into repeated timeout phases. The JF periodic dropping attack utilizes the same principles but realizes the attack via periodic dropping at relay nodes. In particular, suppose that congestion losses force a node to drop % of packets. if these losses occur periodically at the retransmission time out timescale (approximately 1 second), TCP throughput is reduced to near zero even for small values of . Thus, a JF periodic-dropping node can drop no more packets than neighboring congested nodes, but inflict near-zero throughputs on all TCP flows traversing it. Losses due to buffer overflow are predictable in congested environments.

Kuzmanovic and Knightly show that if such losses occur periodically near the retransmission time out (RTxTO) timescale (in the 1s range as RTxTO is intended to address severe congestion), then end-to-end throughput is nearly zero. An *endpoint* attack is described in which a malicious node transmits periodic pulses into the network. As the RTxTO spaced pulses can force all flows sharing the bottleneck link to enter repeated timeout phases, the attack results in all such flows obtaining near-zero throughput while the attacker has a low average transmission rate. The study showed that the impact of the attack can be quite severe whether minimum RTxTO values are all set to 1 second or are randomized over a wide range. [3, 12]

Here, we utilize the same principle for the JF periodic dropping attack in which attacking nodes drop all packets for a short duration (e.g., tens of ms) once per RTxTO. Thus, unlike, JF are passive and generate no traffic themselves; like non-malicious nodes, JF drop for only a small fraction of time; yet, with this dropping pattern during a maliciously chosen period, the following behavior results. Upon encountering the JF's first loss duration, the victim flow will enter timeout as the JF chooses the dropping duration to be sufficiently long to result in multiple losses. When the flow attempts to exit timeout RTxTO seconds later, the

JF will immediately or soon after periodically drop again. Note that the JF knows when a flow enters timeout as the JF itself induced the loss. Thus, the JF can safely assume that by RTxTO seconds later, the flow will be attempting to exit and will be in the fragile slow-start state. [3, 12]

As shown in Fig. 5, depicts the results of simulation experiments with the JF periodic dropping attack. Consider first the upper curve in which the path consists of a source, a single relay node (a JF), and a destination. A time period of 0 indicates no attack and the flow again obtains a throughput of 710 kb/s. The degradation in throughput to the victim is highly non-linear as a function of the dropping period, with null frequencies near 0.5 and 1 second (the minimum RTxTO value). To obtain the null at 1 second, the JF drops packets for 90 ms every 1 second, which results in dropping 9% of the time, and forwarding 91% percent of the time, values easily incurred by a congested node. [3, 12]

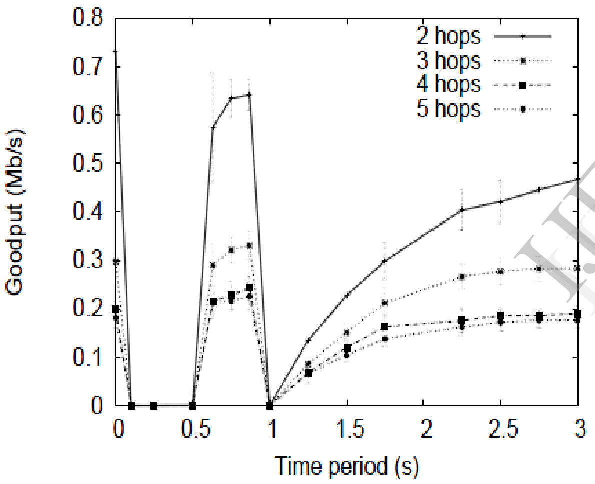

Fig. 5. JF-drop effect on throughput [3, 12].

The attack is therefore successfully exploiting the slow-timescale congestion avoidance mechanism of TCP, namely, that flows must infer that multiple packet losses within a round-trip time are an indication of severe congestion, such that the flow must back off aggressively, and wait RTxTO seconds before entering slow start. Significantly reducing RTxTO or removing the mechanism all together would lead to significant spurious retransmissions and potentially congestion collapse, whereas increasing the value would make the attack even more devastating. Finally, we show how an intermediate JF can attenuate the TCP throughput by varying the RTT. [3, 12]

3.3 JF Delay Variance with Attack

Let, we consider a delay-variance attack in which the attacker delays packets (preserving order) in order to thwart TCP's timers and congestion inferences. This attack not only thwarts widely deployed TCP variants, but also can disrupt rate-based congestion control algorithms. Notice that JF nodes are *protocol compliant* in that IP's datagram service does not mandate loss-free service, in-order delivery, or bounded delay jitter.

Variable round-trip-times due to congestion are an inevitable component of TCP's operation. Yet, ensuring high performance in the presence of random and high delay variation due to an *attacker* was clearly not incorporated into TCP's design. Such a high delay variation can:

1) Cause TCP to send traffic in bursts due to "self-clocking," leading to increased collisions and loss;

2) Cause mis-estimations of available bandwidth for delay-based congestion control protocols such as TCP Westwood and Vegas;

3) Lead to an excessively high RTxTO value. Indeed, enhancing TCP to combat the effects of *non-malicious* delay variation to wireless links has been the focus of intense research, as has the development of tools for available bandwidth estimation. Consequently, *malicious* manipulation of packet delays by the JF delay variance attack has the potential to significantly reduce TCP throughput. Such attackers therefore wait for a variable amount of time before servicing each packet, maintaining FIFO order, but significantly increasing delay variance. [3, 12]

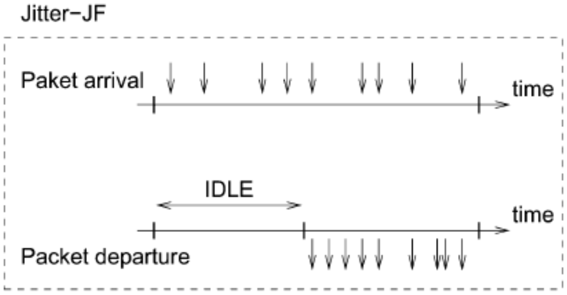

Fig. 6. Jitter implementation used [3,11].

The jitter implementation we used is shown in Fig. 6. In this scenario, the JF behaves as a server with vacations, alternating between periods of serving no packets (and queuing, but not dropping them) and serving packets at its maximum capacity. Both idle and active periods are of equal lengths. Packet departure times are proportional to their arrival times. We deploy this jitter-JF in a three node chain. [3, 12]

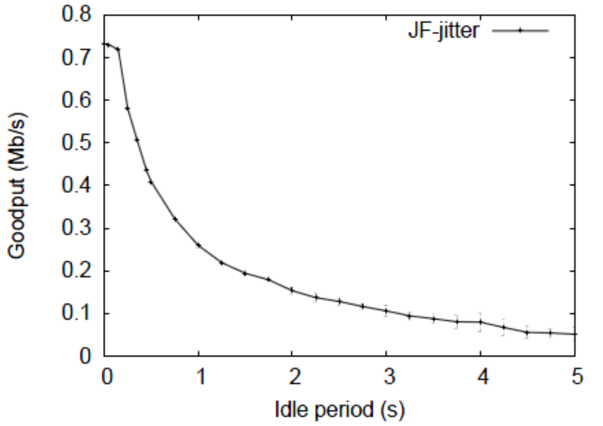

Fig. 7. JF-jitter effect on throughput [3,11].

As Show in Fig. 7 how TCP goodput decreases with increasing jitter (i.e., increasing idle and active periods). While this decreased throughput is also due to increased mean delay, the figure nonetheless indicates that the effects of this attack can be quite severe. [3, 12]

4 Related Works

Imran Raza et al.[5] proposed a solution by introducing two new states R_F and R_{W+F+0} in TCP Reno State transition diagram and uses TCP timestamp options to avoid fast retransmit and timeout problems. The proposed solution prevents TCP Reno to reduce its congestion window size unnecessarily when retransmissions are due to persistent packet reordering attack rather than packet loss; but this cannot consider the malicious node or the malicious route which causes the reordering.

Das, A et al. [6] proposed a novel security scheme for wireless ad-hoc network based on shared information. They proposed to keep redundancy in the number of shares to withstand loss of some shares due to transmission loss as well as due the presence of network layer security threats, but this scheme does not fully mitigate the Jellyfish reorder attack and it does not identify the malicious route.

Tarun Banka et al. [7] proposed a new metric, reorder density function (RD), to represent the reordering of packets in a stream.

Fahad Samad et al. [8] proposed to introduce a security scheme called JAM (Jellyfish Attacks Mitigator) which can be used to detect and mitigate Jellyfish attacks in ad hoc networks.

B. B. Jayasingh et al. [9] proposed to develop an algorithm and novel metric that detects the Jellyfish reorder attack at single node based on the Reorder Density which is a basis for developing a metric. The comparison table shows the effectiveness of novel metric.

There are several derived metrics to monitor packet reordering in network. There are some existing metrics for determining the reordering such as Percentage of Late Packets, Mean Displacement of Packets and the Reorder entropy [10][11].

5 Conclusion and Future Work

This paper, we studied Denial of service attack perpetrated by Jellyfish: relay nodes that stealthily misorder, delay, or periodically drop packets with its impact that they are expected to forward, in a way that leads astray end-to-end congestion control protocols.

Future work involves the study (including implement) of the Jellyfish attacks in Wireless networks. Experimental studies will be conducted in different Wireless Technologies studying the effect of Jellyfish attacks in different scenarios. Implement the way for prevention of jellyfish attack. And analysis of result and performance of the network increases using the proposed solution.

References

[1] Mahendra Kumar,Ajay Bhushan,Amit Kumar"A Study of wireless Ad-Hoc Network attack and Routing Protocol attack"International Journal of Advanced Research in Computer Science and Software Engineering(IJARCSSE),Volume 2, Issue 4, April 2012 ISSN: 2277 128X

[2] I. Aad and J.P. Hubaux, E.W. Knightly, “Impact of Denial of Service Attacks on Ad Hoc Networks”, IEEE/ACM Transactions on Netwroking,vol.16,pp.791-802,Aug.2008.

[3] Nidhi Purohit,Richa Sinha,Hiteishi Diwanji"Simulation Study of Black Hole and Jellyfish attack on MANET Using NS3"Special Issue of International Journal of Computer Applications (0975 – 8887) on Wireless Communication and Mobile Networks, No.9. Jan.2012

[4] Syed Atiya Begum, L.Mohan, B.Ranjitha"Techniques for Resilience of Denial of Service Attacks in Mobile Ad Hoc Networks" International Journal of Electronics Communication and Computer Engineering 2012 Volume 3, Issue (1) NCRTCST, ISSN 2249 –071X

[5] Imran Raza, S. A. Hussain, Amjad Ali , Muhammad Hassan Raza, "Persistent Packet Reordering Attack in TCP Based Ad Hoc Wireless Networks", International Conference on Information and Emerging Technologies (ICIET), Karachi ,pp. 1-6, June 2010.

[6] Das, A.;Basu,S.S.;Chaudhuri, A.; "A novel security scheme for wireless ad-hoc network", 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronics Systems Technology (Wireless VITAE), Chennai,pp.1-4, March 2011.

[7] T. Banka A. A. Bare A. P. Jayasumana, "Reorder Density (RD): A Formal, Comprehensive Metric for Packet Reordering", Lecture Notes in Networking Technologies, Services, and Protocols, Springer, 2005.

[8] Fahad Samad,Qassem Abu Ahmed,Asadullah Shaikh and Abdul Aziz"JAM: Mitigating Jellyfish Attacks in Wireless Ad Hoc Networks"Springer-Verlag Berlin Heidelberg 2012, CCIS 281, pp. 432–444, 2012.

[9] B. B. Jayasingh,B. Swathi"A Novel Metric For Detection of Jellyfish Reorder Attack on Ad Hoc Network"BVICAM'S International Journal of Information Technology (BIJIT)2010 Vol. 2 No. 1 ISSN 0973 - 5658

[10] B. Ye, A. P. Jayasumana and N. Piratla, "On Monitoring of End-to-End Packet Reordering over the Internet," Proc.Int. Conference on Networking and Services (ICNS'06), Santa Clara, CA, July 2006.

[11] Banka, T., Bare, A. and Jayasumana, A., "Metrics for Degree of Reordering in Packet Sequences," Proc. IEEE 27 Local Computer Networks Conf, Nov. 2001, pp. 333-342.

[12] Imad Aad, JeanPierre Hubaux, Edward W. Knightly "Denial of Service Resilience in Ad Hoc Networks" MobiCom'04, Sept. 26Oct.1, 2004, Philadelphia, Pennsylvania, USA.Copyright 2004 ACM 1581138687/04/0009