Volume 3, Issue 8, August 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Mitigate the Impact of DoS Attacks by Verifying Packet Structure

Shiv Kumar, Ritika Singal, Priyadarshni
ECE Department LCET Katani, Kalan
India

Abstract—In this Paper, A mechanism for reducing the DoS (Denial of service) attack has been explored. DoS has become the major threat to the Internet. It is malicious efforts of attackers to deny the legitimate user to access the website or any web service through internet. This is an attack in which attackers tries to cripple the services of internet. Protecting the legitimate traffic from the denial of service attack not only for availability of services but also for effective utilization of network and local resources such as bandwidth , CPU uses ,routers switches etc is necessary. If network implementation is not proper or there are faults in standard specification of network protocols then this results in gaps that allow various types of network attacks including DoS and DDoS attacks. The mechanism is based on the routers that can distinguish the attack packets from the packets of legitimate users and filters the most of traffic to reach the victim. First history of incoming packets is recorded. The source IP address, destination IP address, length of packet to particular destination, number of packets per unit time, sequence no etc has been recorded. The filtration is done according to the parameters of packet structure. The simulations have been done using NS2 (Network Simulator). Graphs of delay, bandwidth and traffic rate are compared with normal traffic, with traffic of DoS attack and after filtration of traffic.

Keywords— DoS , DDoS, NS2, Internet Protocol, Packets

I. INTRODUCTION

A denial of service, or DoS, is a very basic category of attack in the world of network engineering, one which can be used in several scenarios. In this type of attack, an attacker attempts to prevent the use or delivery of a valued resource to its intended audience or customer. It can be implemented via multiple methods, physically and digitally. For instance, an attacker can deny access to telephone systems by cutting the major telecom cable feeding a building, repeatedly calling every available phone line, or distorting PBX. In all above mentioned instances, the attacker succeeds by denying the user’s access to the resource, as all incoming and outgoing calls would fail. The DoS concept can be easily initiated to the world of networks. Networking equipments (Routers, Switches, servers etc) can handle a finite amount of traffic at any given time based on factors such as hardware performance, memory and bandwidth. If this limit or rate is reached, new requests will be rejected. As a result, legitimate traffic will be ignored and the users will be denied access. So, an attacker who wants to disrupt a specific service or device can do so by simply overwhelming the target with packets designed to consume all available resources. A DoS is not a traditional "crack", in which the goal of the attacker is to gain unauthorized privileged access, but it can be just as malicious. The target of DoS is disruption and inconvenience. Success is measured by how long the chaos lasts. When turned against crucial targets, such as root DNS servers, the attacks can be very serious in nature. DoS threats are often among the first topics that come up when discussing the concept of information warfare. They are simple to set up, difficult to stop, and very efficient. Denial of service is about without permission knocking off services. Denial of service can be considered as someone don't get what they paid for. Denial of service attacks are easy to launch and it is hard to protect a system against them.

II. Implementation of Present Work

A. Introduction to NS2:

NS2 is an open-source event-driven simulator designed specifically for research in communication networks. Since its inception in 1989, NS2 has continuously gained tremendous interest from industry, academia, and government. NS2 now contains modules for numerous network components such as routing, transport layer protocol, application, etc due to being under constant investigation and research. To verify network performance, researchers can simply use an easy-to-use scripting language to configure a network, and observe results generated by NS2. NS2 has become the most widely used open source network simulator.

NS simulator is based on two languages: An object oriented simulator written in C++ and a OTCL (an object oriented extension of Tcl) interpreter , used to execute user’s command scripts. It has a wide library of protocol and network objects. There are two class hierarchies: the compiled C++ hierarchy and interpreted OTCL one, with one to one correspondence between them. The compiled C++ hierarchy allows us to achieve efficiency in simulation and faster execution time. This is in particular useful for detailed definition and operation of protocols. This allows one to reduce the packet and event processing time. Then in OTCL script provided by the user, a particular network topology can be defined, the specific protocol and applications that is wished to simulate (whose behaviour is already defined in compiled hierarchy) . The OTCL can make use of objects complied in C++ through an OTCL linkage that creates a matching of OTCL object for each of the C++.

Network Simulator (Version 2), widely known as NS2, is very helpful in understanding the dynamic nature of communication networks. Simulation of wired as well as wireless network functions and protocols (e.g., routing algorithms, TCP, UDP) can be done using NS2. In general, NS2 provides users with a way of specifying such network protocols and simulating their corresponding behaviours. Due to its flexibility and modular nature, NS2 has gained constant popularity in the networking research community since its birth .Several revolutions and revisions have been made since its birth resulted growing maturity of the tool. Among the players responsible for its growth are the University of California and Cornell University who developed the REAL network simulator,1 the foundation which NS is based on.

Virtual Internetwork Testbed (VINT) project was initiated and funded by the Defence Advanced Research Projects Agency (DARPA) in 1995 to support the development of NS. Currently the National Science Foundation (NSF) has joined the ride in development. Last but not the least, the group of researchers and developers in the community are constantly working to keep NS2 strong and versatile. Following figure shows the basic architecture of NS2. NS2 provides users with an executable command “ns” which takes on input argument, the name of a Tcl simulation scripting file. The name of a Tcl simulation script (which sets up a simulation) as an input argument of an NS2 executable command ns are entered by users. In most cases, a simulation trace file is created, and is used to plot graph and/or to create animation. C++ defines the internal mechanism (i.e., a backend) of the simulation objects, the OTcl sets up simulation by assembling and configuring the objects as well as scheduling discrete events (i.e., a frontend). The C++ and the OTcl are linked together using TclCL. Mapped to a C++ object, variables in the OTcl domains are sometimes referred to as handles. Conceptually, a handle (e.g., n as a Node handle) is just a string (e.g._o10) in the OTcl domain, and does not contain any functionality. Instead, the Functionality (e.g., receiving a packet) is defined in the mapped C++ object (e.g., of class Connector). In the OTcl domain, a handle acts as a frontend which interacts with users and other OTcl objects. It may define its own procedures and variables to facilitate the interaction. Note that the member procedures and variables in the OTcl domain are called instance procedures (instprocs) and instance variables (instvars), respectively.NS2 provides a large number of built-in C++ objects. It is advisable to use these C++ objects to set up a simulation using a Tcl simulation script. After simulation, NS2 outputs either text-based or animation-based simulation results. To interpret these results graphically and interactively, tools such as NAM (Network Animator) and Xgraph are used. To analyze a particular behavior of the network, users can extract a relevant subset of text-based data and transform it to a more conceivable presentation.

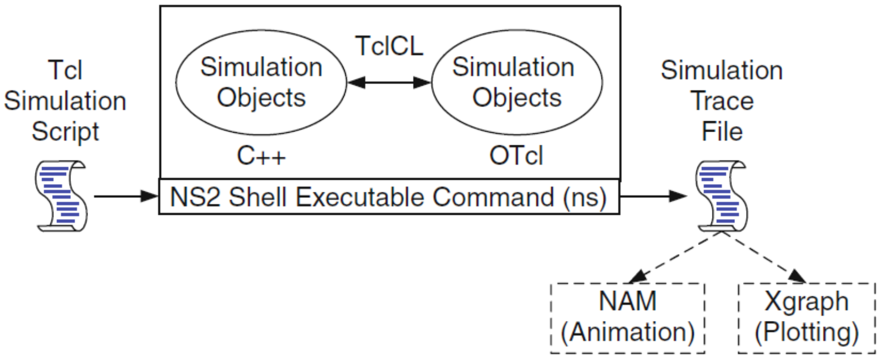

Fig 2.1.Basic Architecture of NS2

B. System Implementation

Seven nodes have been taken for problem formation. Node number 0, 1, 2 and 3 are wired nodes. Node number 4 and 5 are base station nodes. Base station nodes are those nodes which have capability to connect wired as well as wireless mobile nodes. All these six nodes are connected through wires. Normal traffic is moving between these nodes. Node 6 is taken as movable node.

Node 6 moves towards the base station nodes and create the unnecessary traffic causing the denial of service attack. Three Scenarios has been taken into consideration. In first Scenario there is no attack, normal traffic is passing through the network. In 2nd Scenario Node 6 moves towards the base station nodes 4 and 5 and sends the malicious traffic causing the DoS attack. In 3rd Scenario dos attack is detected and new mechanism is implemented to stop the DoS attack. All these scenarios were implemented in NS2 simulator. First normal traffic is monitored, IP address history (Source IP address and Destination IP address), IP length number of packets arriving in normal case when there was no DoS attack is stored. There is counter which simply counts the packets per second, when value of packets per second is increased from threshold value, comparison with IP history starts. First source IP address is checked, if it is same then packet is allowed to pass, if it is different from the IP stored in the history, packet is dropped. There may be IP spoofing. IP spoofing means if attacker knows the source IP address, he will send the malicious traffic with the same source IP address. In this case DoS attack cannot be avoided. Here a new mechanism of checking the length of packet has been entered. Lengths of packets are stored in previous IP history. If any spoofed IP crosses the first barrier then packet length of this IP packet is checked. It will be dropped here. All the attacked packets having different length will be dropped completely. Most of DoS effect will be reduced . Comparison for bandwidth, traffic and delay is done which shows the reduction of DoS attack after implementation of new mechanism. Flow chart shown below explains step by step procedure for implementation of proposed work. For above mentioned three scenarios coding is done in NS2.

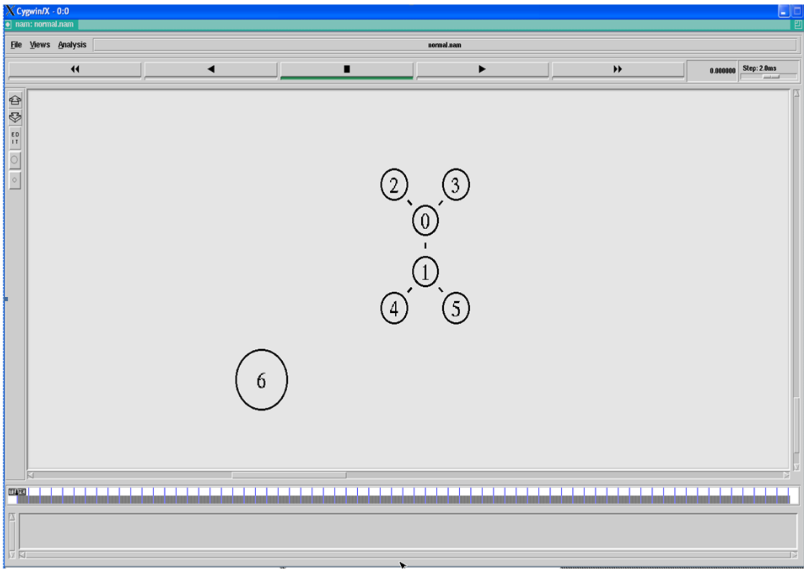

Fig 2.2 Seven nodes used in different scenarios

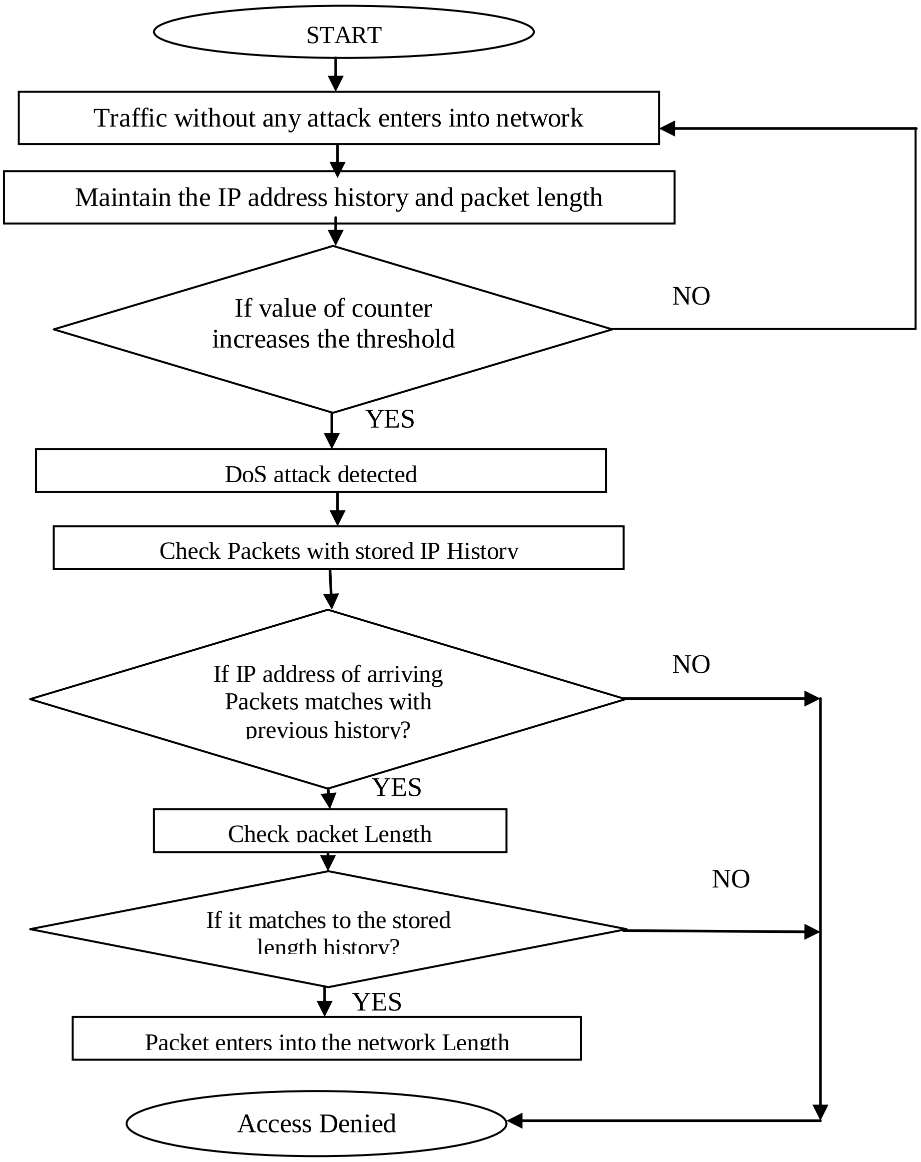
START
Traffic without any attack enters into network
Maintain the IP address history and packet length
If value of counter increases the threshold
NO
YES
DoS attack detected
Check Packets with stored IP History
If IP address of arriving Packets matches with previous history?
NO
YES
Check packet Length
NO
If it matches to the stored length history?
YES
Packet enters into the network Length
Access Denied

Fig 2.2: Flow Chart of Present Work

III. RESULT & DISCUSSION

After implementation of three scenarios results of Bandwidth, delay and traffic are shown below

TABLE I Comparison of Three Scenarios for Different Parameters

Performance Parameters	**SCENARIO 1 (With Normal traffic)**	**SCENARIO 2 (With DoS attack)**	**SCENARIO 3 (After Implementation of New mechanism)**
Max Bandwidth Utilization	3,800 bytes/sec	50,000 bytes/sec	8,400 bytes/sec
Max Delay	12 mSec	36 mSec	22.5 mSec
Max traffic (number of packet/sec)	19.1	33.6	25.2

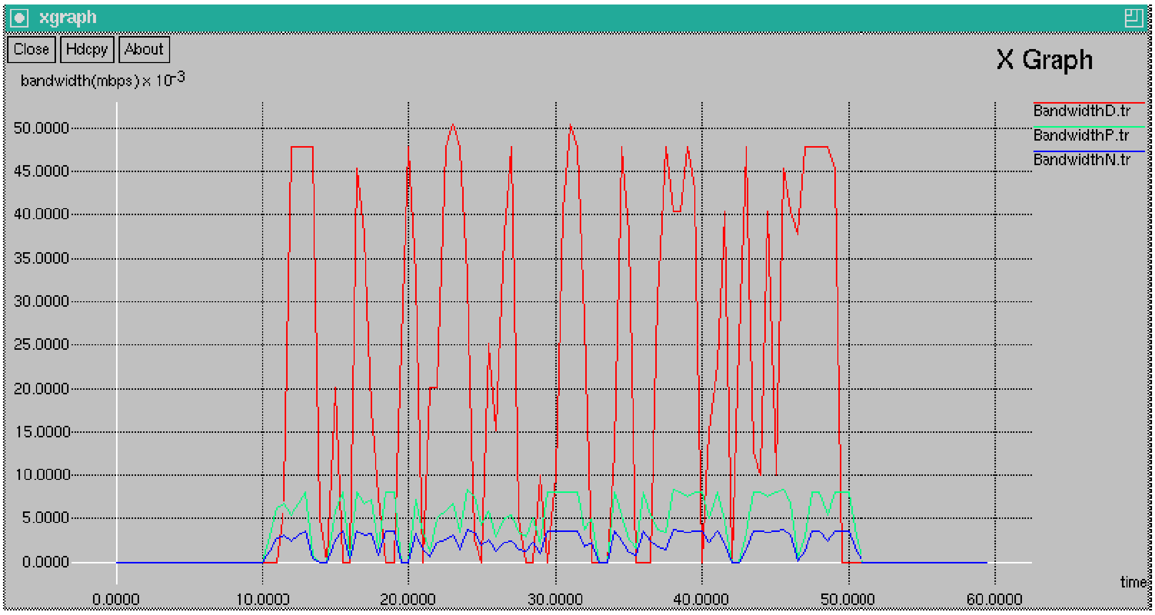

Fig 3.1: Graph shows Comparison between bandwidth

Blue colour represents scenario without DoS attack .Red color represents the scenario with DoS attack. Green color represents the scenario after reducing the DoS attack.

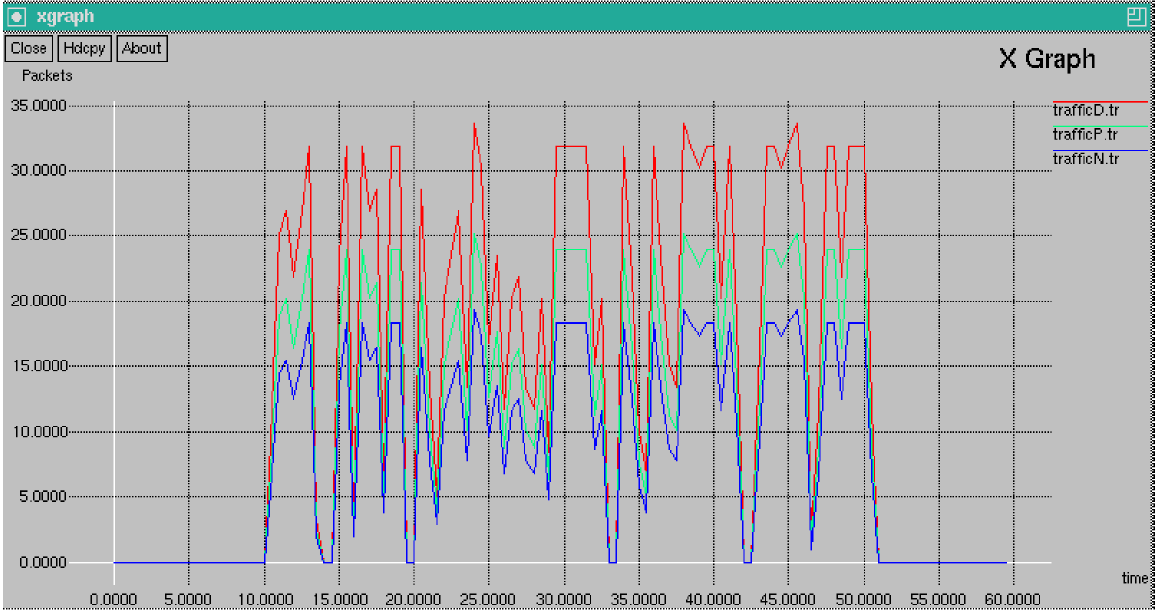

Figure 3.2: Graph shows comparison between traffic

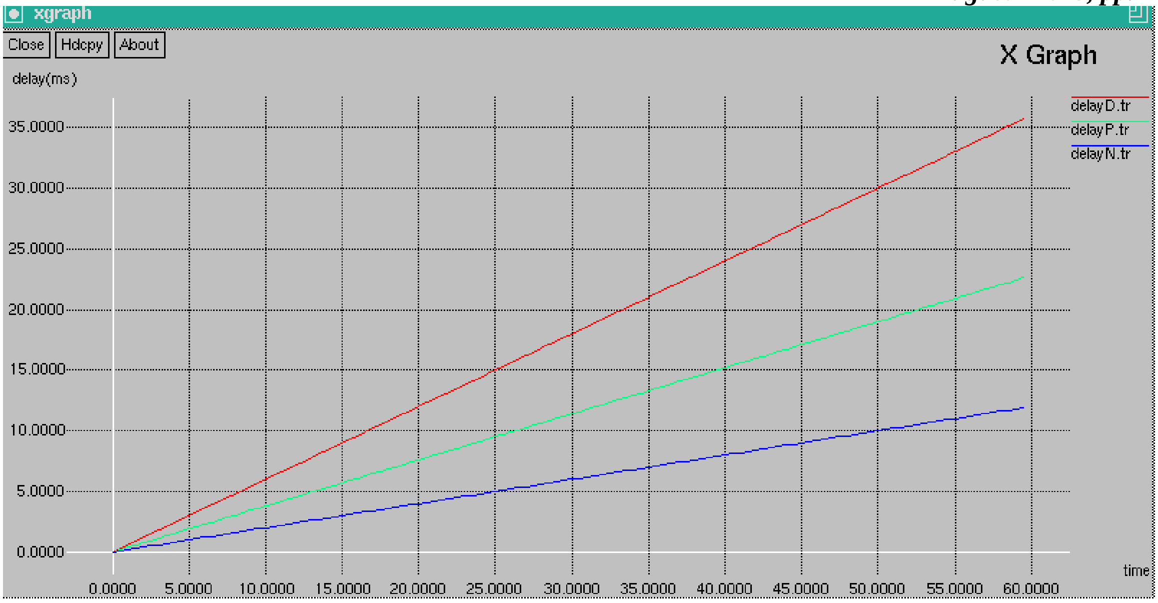

Figure 3.2: Graph shows comparison between delay parameters

IV. Conclusion and Future Scope

In the presence of DoS attack, the average delay of packet travelling through the network is more than that of without attack. After implementation of DoS reducing technique, the value of delay reduces as compared to the value of delay during DoS attack. Bandwidth utilization in case of DoS attack is 50000 bytes/s while bandwidth utilization, after the implementation of DoS mitigating technique, reduces to 8000 bytes/sec. also traffic rate decreases from 33.6 packets/sec to 25.2 packets/sec after using the DoS mitigation technique. It can be concluded from the work presented that DoS attack can be reduced by verifying the packet structure. Here, length of packet has been considered. There are more aspects of packet structure e.g flow id, different flags, time to live, etc. that can be considered to reduce the DoS attack in future.

References

[1] Garg, A., Narasimha Reddy A.L. "Mitigation of DoS attacks through QoS regulation" *in Proc. Quality of Service*, 2002. pp: 45 - 53

[2] Li .J, Mirkovic. J, Wang.M, Reiher.P, and Zhang.L "SAVE: source address validity enforcement protocol"*in Proc. INFOCOM* 2002, vol. 3, pp 1557-1566

[3] Sherr M. , Greenwald M. , Gunter C.A. , Khanna S., Venkatesh S.S. "Mitigating DoS attack through selective bin verification" *in Proc. Secure Network Protocols,* 2005. (NPSec),pp 7-12 .

[4] Badishi G., Herzberg A., Keidar I., Romanov O., Yachin A "An Empirical Study of Denial of Service Mitigation Techniques" *in Proc. Reliable Distributed Systems* 2008. pp 115 - 124

[5] Jayashree. P, Easwara kumar. K.S, Gokul B, Hari shankar S. "Providing QoS as a Means for Defending DoS Attacks in Active Networks" *in Proc. ADCOM 2008* , pp: 406 – 409, 2008

[6] Gupta B. B, Joshi R. C, Mishra. M "Distributed Denial of Service Prevention Techniques" *International Journal of Computer and Electrical Engineering,* Vol. 2 No 2, pp: 1793-8163, 2010

[7] Zhang Fu "Mitigating Distributed Denial-of-Service Attacks: Application-Defense and Network-Defense Methods" *in Proc. Computer Network Defense (EC2ND),* 2011, p 59

[8] Singh. R., Sharma T.P. "Detecting and reducing the denial of Service attacks in WLANs" *in Proc. Information and Communication Technologies (WICT),* 2011, pp 968 – 973

[9] Barna, C., Shtern, M., Smit, M., Tzerpos V., Litoiu M "Model-based adaptive DoS attack mitigation" *in Proc. Software Engineering for adaptive and Self managing systems (SEAMS 2012)*2012 p: 119 - 128

[10] Yu Ming **"**Mitigating Flooding-Based DDoS Attacks by Stochastic Fairness Queuing" *Academic Journal of Advances in Information Sciences & Service Sciences*, vol. 4 Issue 6, p 145 ,2012

[11] N. Jeyanthi and. Sriman Narayana Iyengar N. Ch "An Entropy Based Approach to Detect and Distinguish DDoS Attacks from Flash Crowds in VoIP Networks*" International Journal of Network Security*, vol.14, No.5, pp.257-269,2012

[12] Priyadharshini V, Kuppusamy .K "Prevention of DDOS Attacks using New Cracking Algorithm", *International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com* Vol. 2, Issue 3, pp. 2263-2267, 2012

[13] Bhange,A, Syad A, Thakur. S.S "DDoS Attacks Impact on Network Traffic and its Detection Approach" *International Journal of Computer Applications, December 2012. Published by Foundation of Computer Science, New York, USA,* Vol 40– No.11 pp: 36-40, 2012

[14] Nagarjun , P.M.D, Kumar V.A, Kumar C.A "Simulation and analysis of RTS/CTS DoS attack variants in 802.11 networks" *in Proc PRIME International conference* 2013, pp 258-263

[15] DARPA "*NS Manual from VINT (virtual internetwork testbed) Project*" http://www.isi.edu/nsnam/ns/ ,Oct 1997

[16] Johanna Antila "*TCP Performance Simulations using NS2*" M.Eng. Thesis (March 2002)

[17] Eitan Altman and Tania Jimenez "*NS simulator for beginners*" University De Los Andes , Merida and Venezuela France ,December 2003

[18] Paul Meeneghan and Declan Delaney "*An introduction of NS, NAM and OTCL scripting*" Department of computer science, university of Ireland NUIM-CS-TR, 2004-05

[19] Ke Liu "*Network Simulator 2 Introduction*" Department of computer science SUNY Binghamton, 2006

[20] Teerawat Issariyakul and Ekram Hossain "*Introduction to network simulator NS2*" Springer Science & Business Media ,Edition 2009