Entropy Based Detection for DDoS Attack

Project Report submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

In

Computer Science & Engineering

Under the Supervision of

Ms. Ramanpreet Kaur

By

Gena Bansal

Roll No: 111308

Jaypee University of Information Technology

Waknaghat, Solan – 173234

Himachal Pradesh

Certificate

This is to certify that project report entitled “Entropy Based Detection for DDoS Attack”, submitted by Gena Bansal in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science & Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision. This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date:

Ms Ramanpreet Kaur
Assistant Professor
Dept. of Computer Science & Engineering
Jaypee University of Information Technology

Acknowledgement

I wish to express my profound gratitude and indebtedness to Ms. Ramanpreet Kaur, for introducing the present topic and for her inspiring guidance, constructive criticism and valuable suggestion throughout the project work. Last but not least, my sincere thanks to all my friends who have patiently extended all sorts of help for accomplishing this undertaking.

Date:

Gena Bansal
111308
Computer Science & Engineering

Table of Content

List of Figures

List of Tables

ABSTRACT

Distributed Denial of Service (DDoS) attacks have emerged a popular means of causing mass targeted service disruptions, sometime for extended periods of time. The relative ease and low cost of launching such attacks, supplemented by the current inadequate defense mechanism, have made them one of the top threats to the Internet community today. Since the increasing popularity of web-based applications has led to several critical services being provided over the Internet, it is necessary to monitor the network traffic so as to prevent malicious attackers from depleting the resources of the network and denying services to legitimate users. Due to increase in sophistication of attacks and large complex networks have made the defense mechanism challenging. Although a number of techniques have been proposed to defeat DDoS attacks but still it is very hard to detect and respond to DDoS. An important method for DDoS defense is to effectively detect the attack. This report provides an overview of existing DDoS attacks along with the current state of art of detection mechanism. My work focuses on simulating UDP flood based DDoS attack on dumb-bell topology in NS2 environment and analyzing the effect of UDP flooding on various performance metrics. Throughput and packet drop rate are analyzed with and without DDoS attacks. And later, implementing the existing statistical techniques: Entropy Based and Chi-Square approach for DDoS attack detection. The detection mechanism continuously monitors incoming traffic to the server and any abnormal rise in the inbound traffic is detected using Entropy Variation technique. Secondly, Chi- square approach is used to test traffic data with specific distributions. Chi-square goodness-of fit test is that it can be applied to binned data (i.e., traffic data put into classes) and chi-square test is defined for the hypothesis whether the traffic data follow a specified distribution or not. Simulation environment consists of two scenarios: first with the continuous and constant attack traffic and second with the attack varying in timing intervals like attack occurring between time intervals 10-20 ms and 40-50ms.Attack is comparatively easy detect in first scenario than in the second because of the frequent variation. Performance Evaluation considers the detection rate and false positive alarm rate.

1. INTRODUCTION

In computing, Distributed Denial of Service attack (DDoS) is an attempt to make a machine or network resource unavailable to its intended users. Although the means to carry out, motives for, and targets of a DoS attack may vary, it generally consists of the efforts of one or more people to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet. DDoS attacks typically target sites or services hosted on high profile web servers such as banks, credit card payment gateways, and even root name servers. The term is generally used relating to computer networks, but is not limited to this field; for example, it is also used in reference to CPU resource management. One common method of attack involves saturating the target machine with external communications requests, so much so that it cannot respond to legitimate traffic or responds so slowly as to be rendered essentially unavailable. Such attacks usually lead to a server overload. In general terms, DDoS attacks are implemented by either forcing the targeted computers to reset, or consuming its resources so that it can no longer provide its intended service or obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately. DDoS attacks are considered violations of the Internet proper use policy.

1.1 Definition

A Denial of Service (DoS) attack is an explicit attempt by a malicious user to consume the resources of a server or a network, thereby preventing legitimate users from availing the services provided by the server or the network. When multiple attackers coordinate together to consume the resources of a network or a server then it is referred to as Distributed Denial of Service (DDoS) attack. Some examples of DDoS attack are: SYN flooding, UDP flooding, DNS-based flooding, ICMP directed broadcast, Ping flood attack, IP fragmentation, and CGI attack. Owing to the distributed nature, the DDoS attacks are very difficult to detect.

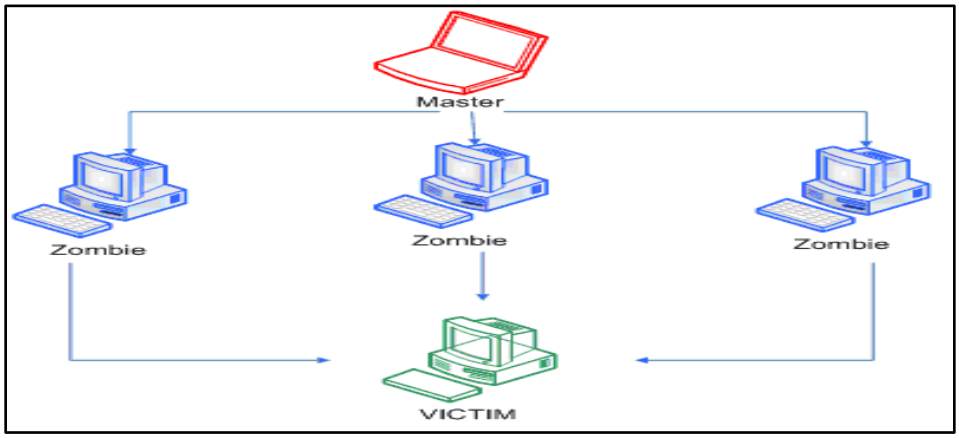

Fig 1.1 A conceptual diagram of DoS Attack

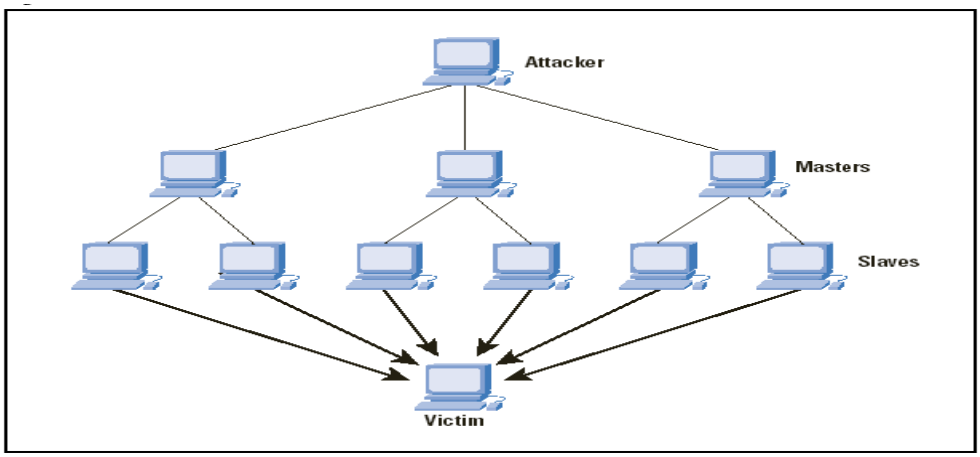

Fig 1.2 A conceptual diagram of DDoS Attack

1.2 Manifestation of DDoS Attack

Symptoms of denial-of-service attacks to include:

• Unusually slow network performance.

• Unavailability of a particular web site.

• Inability to access any web site.

• Dramatic increase in the number of spam emails received.

• Disconnection of internet connection.

• The term "hit offline" being used and then the target may disconnect from the internet.

2. Literature Review

2.1 HISTORY

Denial-of-service attacks under a number of guises have been around for decades. They started in 90's, at the first stage they were quite "primitive", involving only one attacker exploiting maximum bandwidth from the victim, denying others the ability to be served. Distributed DoS attacks, first being seen in late June and early July of 1999. The first well-documented DDoS attack appears to have occurred in August 1999, when a DDoS tool called Trinoo was deployed in at least 227 systems, of which at least 114 were on Internet, to flood a single University of Minnesota computer; this system was knocked off the air for more than two days. The first well-publicized DDoS attack in the public press was in February 2000. On February 7, Yahoo! was the victim of a DDoS during which its Internet portal was inaccessible for three hours. On February 8, Amazon, Buy.com, CNN, and eBay were all hit by DDoS attacks that caused them to either stop functioning completely or slowed them down significantly. And, on February 9, E*Trade and ZDNet both suffered DDoS attacks. Analysts estimated that during the three hours Yahoo was down, it suffered a loss of e-commerce and advertising revenue that amounted to about $500,000. According to book seller Amazon.com, its widely publicized attack resulted in a loss of $600,000 during the 10 hours it was down.

In late 2012 to 2013, 46 financial institutions in the United States were hit with over 200 coordinated and timed DDoS attacks. Recent massive attack ever in 2014 targeting the content delivery network Cloudflare reached new highs, striking at the company's data centers in Europe and the US. According to Cloudflare CEO, the full volume of the attack exceeded 400 gps. Not only DDoS attacks increased in number, but more tools are becoming available that make them easier to pull off. Now, attacker possess a fair degree of skill and recruit an army of computers into a botnets in order to create enough computing power to launch an attack, recent attack methods require considerably fewer resources and less skill. DDoS attack kits like "HPING", making the job of a relatively unskilled hacktivists much easier, and DDoS-as-a-service attacks are an increasingly common phenomenon, whereby attackers hire themselves and their botnets out to those

wishing to launch attacks. Another recent development is the use of network time protocol amplification attacks, which use publicly available network time protocol servers. Recently, there has also been a dramatic rise in mobile applications used in DDoS attacks, driven by the ease with which mobile apps can be downloaded. These apps allow any mobile user to join a DDoS attack if he or she wishes—for example, for an ideological cause with which he or she sympathizes. It is predicted that such attacks will increase dramatically.

According to Huawei Cloud Security Center, the top three IDC service attack targets are e-commerce, online gaming, and DNS services. Attacks on online financial service systems are usually motivated by political intentions, blackmail, and obscuring unauthorized operations.

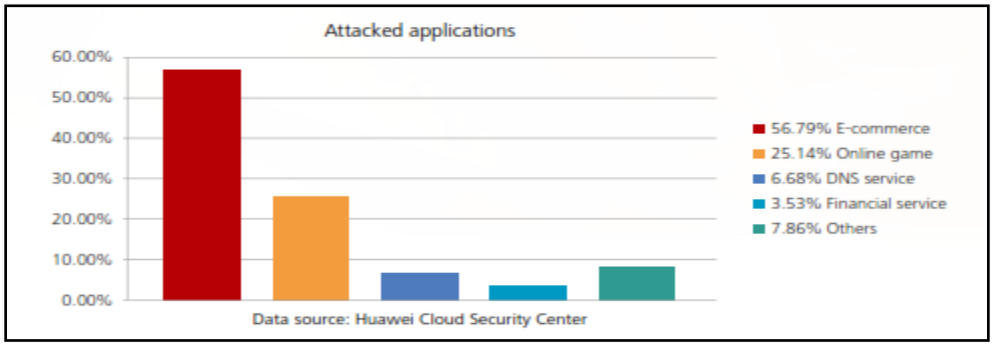

Fig 2.1 Attacked Applications

According to Huawei cloud security center statistics, botnets are small-scale and specialized globally, targeting at a part instead of the whole. Botnets with less than 1000 hosts are common since they are easily controlled.

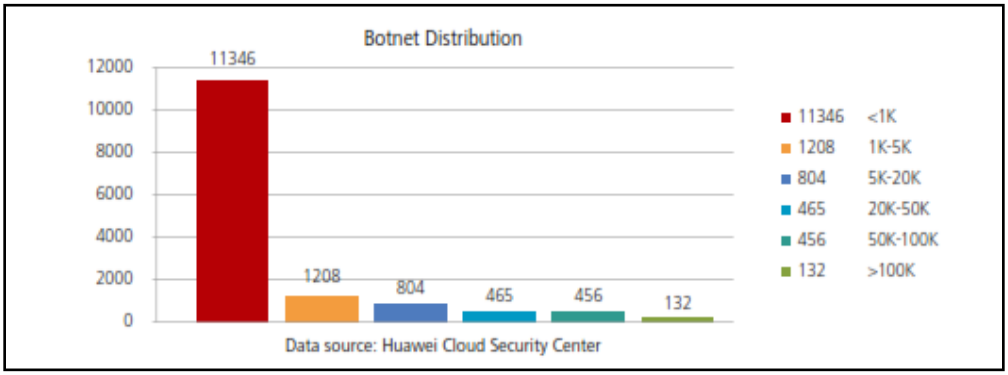

Fig 2.2 Botnet Distribution

2.2 Classification OF DDoS Attack

In general, DDoS attacks can be broadly divided into three types:

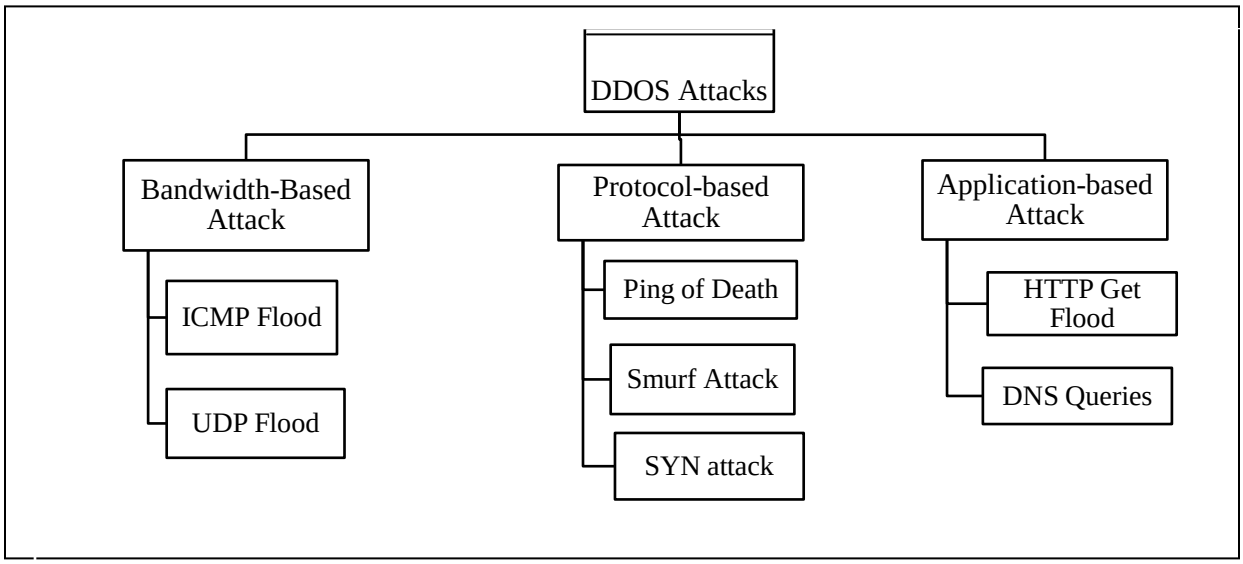

Fig 2.3 Classification of DDoS Attack

Bandwidth-based attacks

Attacks of this type send mass junk data messages to cause an overload, leading to the depletion of network bandwidth or equipment resources. Often the attacked routers, servers and firewalls processing resources are limited. Overload attacks lead to their failure in handling normal legal access, resulting in either a sharp decline in the quality of service or a complete denial of service - in either case it means your customers, users, etc cannot access the systems they need to.

Protocol-based attacks

Attacks of this type exploit a specific feature or implementation bug of some protocol installed at the victim in order to consume excess amounts of its resources. It includes SYN floods, fragmented packet attacks, Ping of Death, Smurf DDoS and more.

Application-based attacks

Attacks of this type often send application-layer data messages according to business-specific features resulting in the depletion of certain resources in the application layer

(such as the number of users, connections, etc.) and the system's services are no longer available. Such attacks are usually not particularly large in volume; but even such low-rate traffic can often lead to a serious declination or even paralysis of business system performance.

2.3 Specific DDoS Attacks

UDP Flood

UDP is a sessionless networking protocol. One common DDoS attack method is referred to as a UDP flood. Random ports on the target machine are flooded with packets that cause it to listen for applications on that those ports and report back with a ICMP packet.

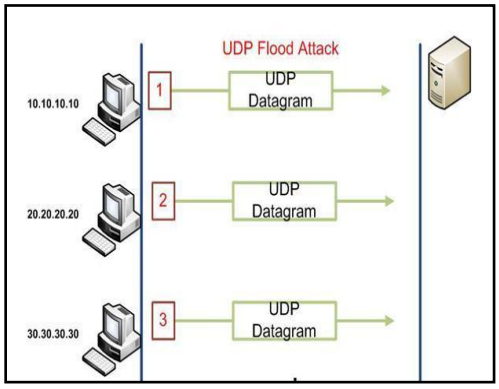

Fig 2.4 UDP Flood Attack

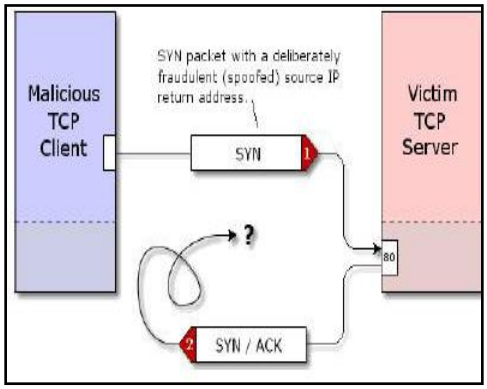

Fig 2.5 Syn Attack

SYN Flood

A SYN flood attack will send repeated spoofed requests from a variety of sources at a target server. The server will respond with an ACK packet to complete the TCP connection, but instead of closing the connection the connection is allowed to timeout. Eventually, and with a strong enough attack, the host resources will be exhausted and the server will go offline.

Ping of Death

It is an attack that was prevalent in legacy systems. PoD is conducted using the ping command. Within IPv4 based networks, a ping command's total payload size is 84 bytes

and the maximum size of network packet a computer can handle is 65,536 bytes. To initiate a PoD, the attacker sends a ping packet larger than 65,536 bytes, which makes the system unstable, crash or reboot.

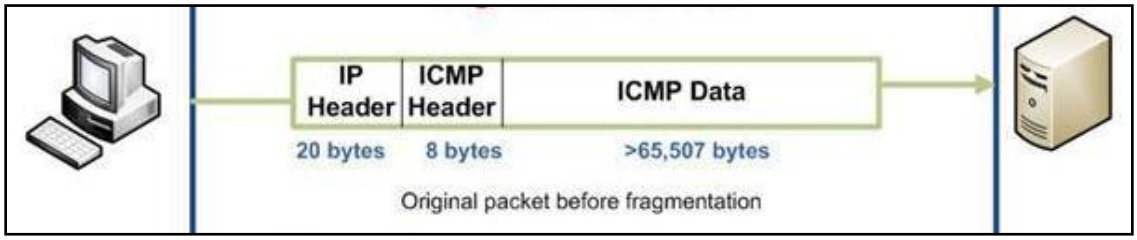

Fig 2.6 Ping of Death Attack

Smurf Attack

It is a attack in which large numbers of ICMP packets with the intended victim's spoofed source IP are broadcast to a computer network using an IP Broadcast address. Most devices will respond to this by sending a reply to the source IP address. If the number of machines that receive and respond to these packets is very large, the victim's computer will be flooded with traffic and it slow down the victim's computer.

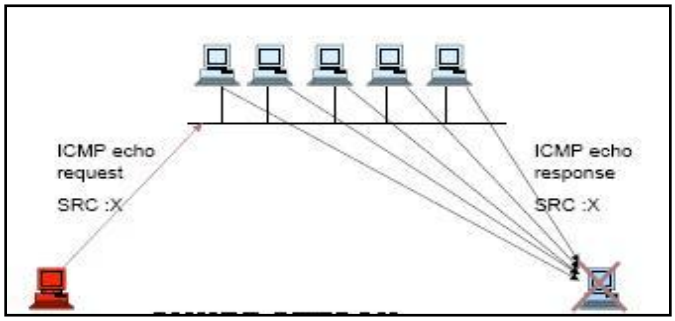

Fig 2.7 Smurf Attack

Unintentional DDoS

Unintended distributed denial of service happens when a spike in web traffic causes a server to not be able to handle all of the incoming requests. The more traffic that occurs, the more resources are used. This causes pages to timeout when loading and eventually the server will fail to respond and go offline.

2.4 Impacts of DDoS Attack

Besides the obvious downtime that a DDoS attack generates, there might be other impacts to the victim. Of course, downtime and the issues caused in all of the dependent systems can be devastating enough, but the attack may generate additional costs, as some hosting solutions, including DDoS-protected ones, bill the user for the traffic used by its clients. Therefore, even if the DDoS attack was not successful in bringing down the website, the victim may still have to pay a substantial amount of money for the used bandwidth. This is in addition to the maintenance and operational costs of the site.

Depending on the sector of business that the victim is in, DDoS attacks might lead to customer frustration. If, for example, the online store goes down, customers may lose confidence in the brand and go somewhere else to shop. This could result in a loss of revenue or in some cases, even repression, depending on the granted service-level agreements (SLAs). This loss is very hard to measure. Some servers and network devices might start to reboot when they exhaust their resources.

2.5 Defense Mechanism

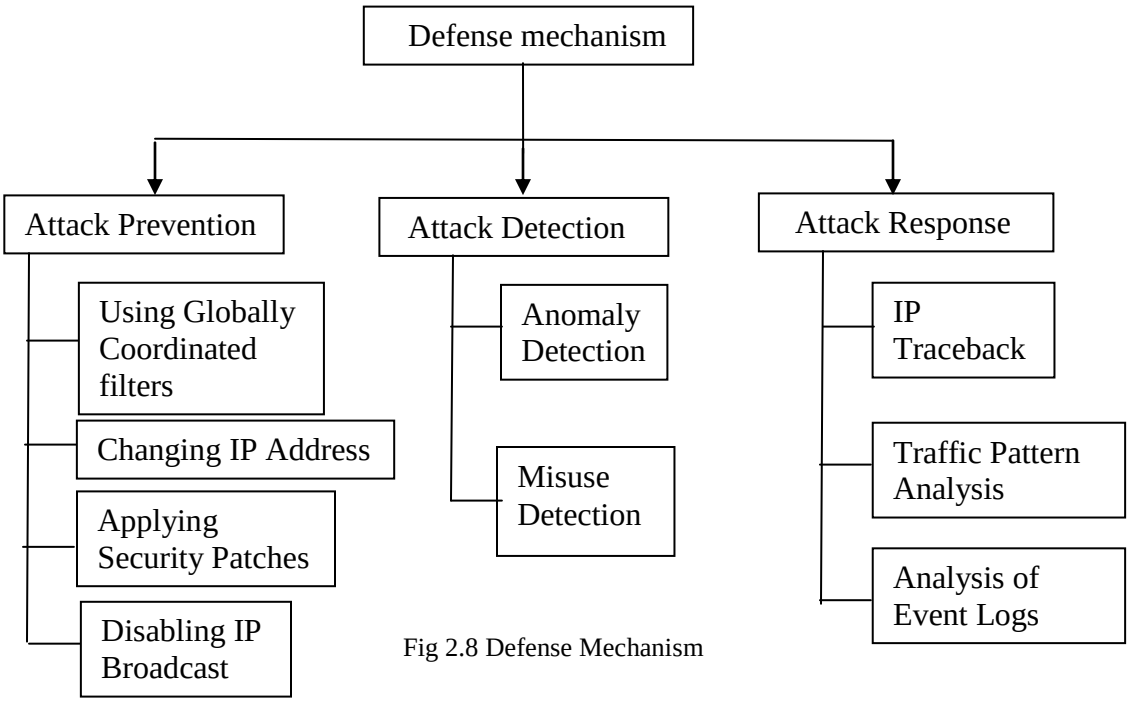

Fig 2.8 Defense Mechanism

2.5.1 Attack Prevention

The best mitigation strategy against any attack is to completely prevent the attack. In prevention stage DDoS attacks are stopped from being launched in the first place. Prevention approaches [9] offer increased security but can never completely remove the threat of DDoS attacks because they are always vulnerable to new attacks for which signatures and patches do not exist in the database. Few mechanisms are:

- Using Globally Coordinated Filters

Filtering mechanisms can be divided into the following categories:

Ingress filtering is an approach to set up a router such that to disallow incoming packets with illegitimate source addresses into the network. This mechanism can drastically reduce the DoS attack by IP spoofing if all domains use it. Sometimes legitimate traffic can be discarded by an ingress filtering when Mobile IP is used to attach a mobile node to a foreign network.

Egress filtering is an outbound filter, which ensures that only assigned or allocated IP address space leaves the network. Egress filters do not help to save resource wastage of the domain where the packet is originated but it protects other domains from possible attacks.

- Applying Security Patches

Applying security patches, can armor the hosts against DDoS attacks. The host computers should update themselves with the latest security patches for the bugs present and use the latest techniques available to minimize the effect of DDoS attack.

- Changing IP Address

Changing IP address [46], is another simple solution to a DDoS attack in order to invalidate the victim computer's IP address by changing it with a new one. This is called moving target defense.Once the IP address change is completed all Internet routers will have been informed, and edge routers will drop the attacking packets.

2.5.2 Attack Detection

It has been a very active research area. By performing detection, a host computer and a network can guard themselves against being a source of network attack as well as being a victim of DDoS attack.

- Anomaly Detection: Detects behaviors that are abnormal with respect to some normal standard. Many anomaly detection systems and approaches have been developed to detect the faint signs of DDoS attacks. It includes various statistical approaches [3] like Chi-Square Statistics [8] and Entropy variation [8] are few schemes.

2.5.2.1 Statistical Approaches

Statistical properties of normal and attack patterns can be exploited for detection of DDoS attacks. Generally a statistical model for normal traffic is calculated and then a statistical inference test is applied to determine if a new instance of the traffic or flow belongs to this model. Instances that do not follow the normal model, based on the applied test statistics results, traffic or flows are then classified as anomalies.

1. Chi-Square Statistic

Pearson's chi-square ($\chi 2$) test is used to compare distributions. In the chi-square DDoS detector, the current distribution of values for some packet attribute(source address) is compared against a baseline measurement representing typical traffic seen in that detector's environment. When the chi-square statistic indicates a substantial discrepancy between the baseline and current distributions, the detector concludes that a DDoS attack may have begun. The test works best when the number of possible values is small. In particular, a rule of thumb is that the expected number of packets in a sample having each possible value be at least five. However, this can often be achieved through "binning", that is combining a set or range of possible values and treating them as one. Packet lengths can be binned into ranges such as 0-64 bytes, 65-128 bytes, 129-255 bytes, etc. In practice, it is found that defining bins dynamically based on the frequency-sorted distribution of values is often best. For example, five bins for the IP source address attribute might be defined as follows: The most frequent address, the next four most frequent addresses, the next 16, the next 64, and the remainder. This Statistical approach is memory intensive.

For a sample of N packets, let B be the number of available bins. Define N as the number of packets whose value falls in the i^{th} bin and n_i as the expected number of packets in

the i^{th} bin under the typical distribution. Then the chi-square statistic is computed as follows:

$$\chi^2 = \sum_{i=1}^{B} \frac{(n_i + N_i)^2}{n_i}$$

2. Entropy

In information theory, entropy is a measure of the uncertainty associated with a random variable. The entropy detection method is mainly used to calculate the distribution randomness of some attributes in the network packets' headers. These attributes could be the packet's source IP address, or some other values indicating the packet's properties. After analyzing the characteristics of DDoS attack, we know that, when the attack comes out, there is large number of data packets, high volume of traffic flow, and many incomplete connection requests. The attackers always fabricate a lot of data packets, and the IP addresses of these packets are generally different and randomly distributed. The analysis of these characteristics could help us to detect the DDoS attack better. The formula of entropy calculation is as follows:

$$\boldsymbol{H = -\sum_{i=1}^{n} (p_i \log_2 p_i)}$$

Where,

$\boldsymbol{p_i}$ is the emergence probability of each distinct source IP address.

n is the total number of packets being analyzed, and

H is the entropy.

- Misuse Detection: Identify well-defined patterns of known exploits and then looks out for the occurrences of such patterns. These patterns are defined as attack signatures. Several popular network monitors perform signature-based detection, such as CISCO_s NetRanger [14], NID, SecureNet PRO, RealSecure and Snort [15].

2.5.3 Attack Response

Once an attack is identified, the immediate response is to identify the attack source and block its traffic accordingly. The blocking part is usually performed under manual control (e.g. by contacting the administrators of upstream routers and enabling access control lists) since an automated response system might cause further service degradation in response to a false alarm. Automated intrusion response systems do exist, but they are deployed only after a period of self-learning. Some techniques are:

- IP Traceback[6]

  IP traceback traces the attacks back towards their origin, so one can find out the true identity of the attacker and achieve detection of asymmetric routes, as well as path characterization. Some factors that render IP traceback difficult is the stateless nature of Internet routing and the lack of source accountability in the TCP/IP protocol.

- Traffic Pattern Analysis

  Traffic Pattern Analysis is another method in order to response to DDoS attacks. During a DDoS attack, traffic pattern data can be stored and then analyzed after the attack in order to find the specific characteristics and features that may indicate an attack.

- Analysis Of event logs

  Analysis of event logs is another good approach that targets the response to DDoS attacks. The selection of event logs that occurred during the setup and the execution of the attack can be used, in order to discover the type of DDoS attacks that has been used and do a forensic analysis. Network equipment such as firewall can be used in the selection of event logs.

3. **Introduction to NS2**

3.1 Overview of NS-2

NS (Version 2) is an open source network simulation tool. It was developed as part of the VINT project (Virtual Internet Testbed).This was a collaboration of many institutes including UC Berkeley, AT&T, XEROX PARC and ETH. Version 1 of NS was developed in 1995 and with version 2 released in 1996. It is an object oriented, discrete event driven simulator written in C++ and Otcl. The primary use of NS is in network researches to simulate various types of wired/wireless local and wide area networks. It is used to implement network protocols such as TCP and UPD, traffic source behavior such as FTP, Telnet, Web, CBR and VBR and router queue management mechanism such as Drop Tail, RED and CBQ and routing algorithms. Ns2 is written in C++ and Otcl to separate the control and data path implementations.

The reason why ns2 uses two languages is that different tasks have different requirements: For example simulation of protocols requires efficient manipulation of bytes and packet headers making the run-time speed very important. On the other hand, in network studies where the aim is to vary some parameters and to quickly examine a number of scenarios the time to change the model and run it again is more important.

In ns2, C++ is used for detailed protocol implementation and in general for such cases where every packet of a flow has to be processed. For instance, if you want to implement a new queuing discipline, then C++ is the language of choice. Otcl, on the other hand, is suitable for configuration and setup. Otcl runs quite slowly, but it can be changed very quickly making the construction of simulations easier.

NS-2 has many and expanding uses including:

- To evaluate the performance of existing network protocols.
- To evaluate new network protocols before use.
- To run large scale experiments not possible in real experiments.
- To simulate a variety of ip networks.

3.2 Design of NS-2

NS-2 is an Object-oriented Tcl (OTcl) script interpreter that has a simulation event scheduler and network component object libraries.

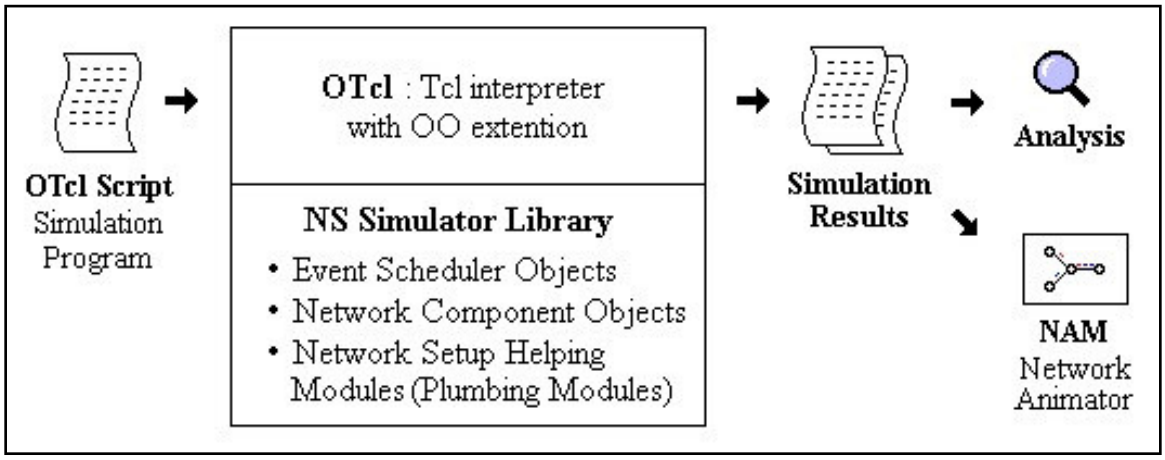

Fig 3.1 Simplified view of NS-2

An OTcl script will do the following.

- Initiates an event scheduler.
- Sets up the network topology using the network objects.
- Tells traffic sources when to start/stop transmitting packets through the event scheduler.

Depending on the purpose for an OTcl simulation script, simulation results are stored as trace files, which can be loaded for analysis by an external application:

- A NAM trace file (file.nam) for use with the Network Animator Tool
- A Trace file (file.tr) for use with XGraph or TraceGraph .

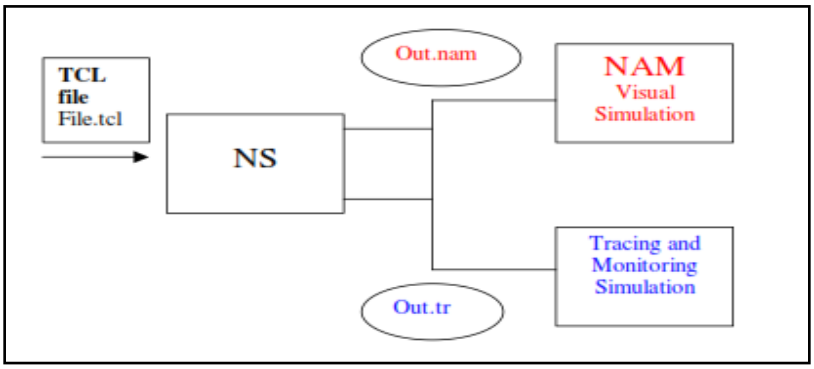

Fig 3.2 Flow event for a TCL file

3.3 Software tools used with NS-2

3.3.1 NAM

NAM provides a visual interpretation of the network topology created Figure displays the NAM application and its components. It

- Provides a visual interpretation of the network created
- Can be executed directly from a Tcl script
- Controls include play, stop ff, rw, pause, a display speed controller and a packet monitor facility.
- Presents information such as throughput, number packets on each link.
- Provides a drag and drop interface for creating topologies.

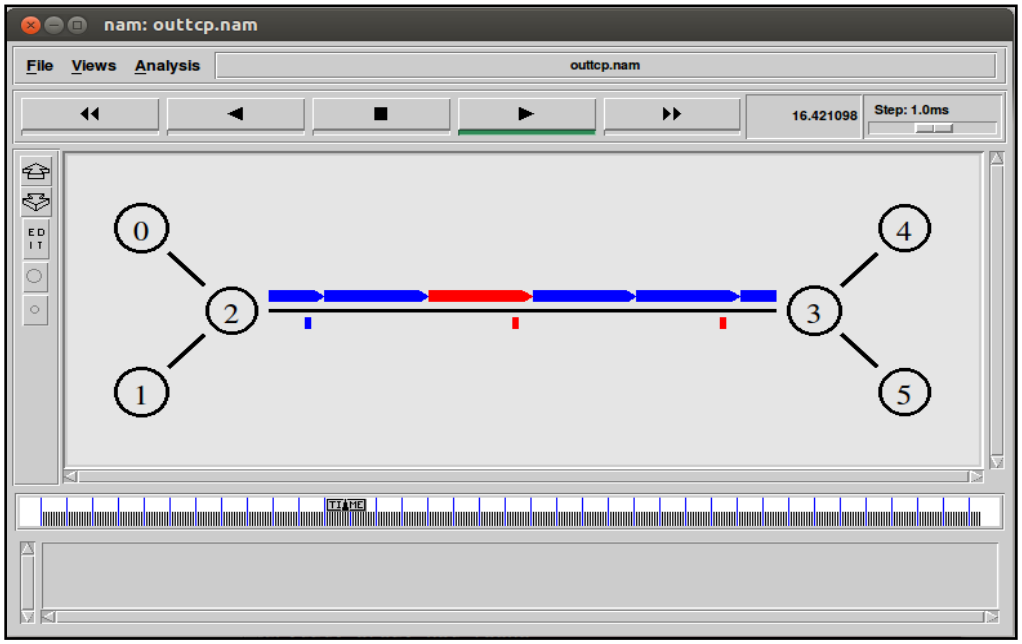

Fig 3.3: NAM Editor

3.3.2 Xgraph

XGraph is an X-Windows application that includes:

- Interactive plotting and graphing
- Animation and derivatives

To use XGraph in NS-2 the executable can be called within a TCL Script. This will then

load a graph displaying the information visually displaying the information of the trace file produced from the simulation

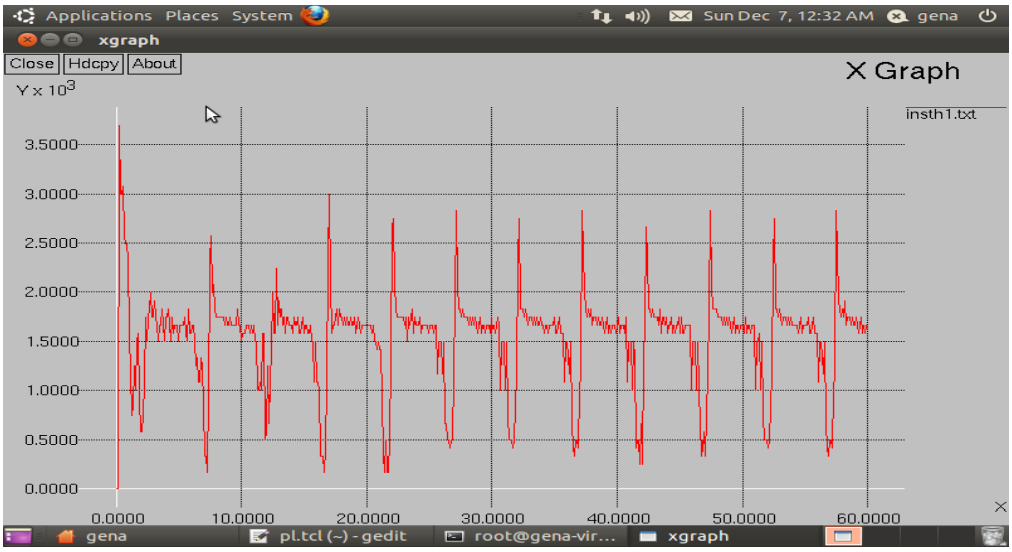

Fig 3.4: XGraph running for any file

3.3.3 Topology Generators

Topology Generators are used with NS-2 to create a network topology to simulate a certain network model. Each topology generator provides a Graphical User Interface. The user can then choose the structure of the topology e.g. number of nodes. When this is complete the generator can be run to produce TCL code depicting the topology for use with NS-2. The four most common topology generators are as follows.

GT-ITM

This generator focuses on reproducing the hierarchical structure of the topology of the Internet based on the TS (Transit Stub).

Tiers

This is based on a three level hierarchy aimed at reproducing the differentiation between WAN, MAN and LAN compromising the Internet.

Brite

This is a single generation model providing several degrees of freedom with respect to how nodes are placed in the plane.

Inet

This generator initially assumes node degrees from a power law distribution.

3.4 OTcl Scripting with NS-2

Topology Creation

To set up the topology, a new simulator object must be created at the beginning of the script with the command:

```
set ns [new Simulator]
```

The simulator object has member functions that enable creating the nodes and the links, connecting agents etc. All these basic functions can be found from the class Simulator. When using functions belonging to this class, the command begins with "$ns", since ns was defined to be a handle to the Simulator object.

Node Creation

New node objects can be created with the command:

```
set n0 [$ns node]
set n1 [$ns node]
```

The member function of the Simulator class, called "node" creates the nodes and assigns them to the handles n0 and n1. These handles can later be used when referring to the nodes. If the node is not a router but an end system, traffic agents (TCP, UDP etc.) and traffic sources (FTP,CBR etc.) can be set up, that is sources need to be attached to the agents and the agents to the nodes, respectively.

Node Links

A unidirectional link between the two nodes is created as follows

- A Simplex link (one way) - $ns simplex-link $n0 $n1 <bandwidth><delay> <queue_type>

A bi-directional link between the two nodes is created as follows

- A duplex link (both ways) - $ns duplex-link $n0 $n1 <bandwidth> <delay> <queue_type>

NS-2 supports numerous queue types including FIFO, RED (Random Early Detection), Drop Tail, FQ (Fair Queuing), SFO (Stochastic)

Network Agent, application and traffic sources

The most common agents used in ns2 are UDP and TCP agents. In case of a TCP agent, several types are available. The most common agent types are:

- Agent/TCP – a Tahoe TCP sender
- Agent/TCP/Reno – a Reno TCP sender
- Agent/TCP/Sack1 – TCP with selective acknowledgement

The most common applications and traffic sources provided by ns2 are:

- Application/FTP – produces bulk data that TCP will send
- Application/Traffic/CBR – generates packets with a constant bit rate
- Application/Traffic/Exponential – during off-periods, no traffic is sent. During on - periods, packets are generated with a constant rate. The length of both on and off-periods is exponentially distributed.
- Application/Traffic/Trace – Traffic is generated from a trace file, where the sizes and interarrival times of the packets are defined.

Creation of CBR traffic source using UDP as transport protocol and attach it to node say, n0:

```
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set packet_size_ 1000
$udp0 set packet_size_ 1000
$cbr0 set rate_ 1000000
```

An FTP application using TCP as a transport protocol can be created and attached to node n1 in much the same way:

```
set tcp1 [new Agent/TCP]
```

$ns attach-agent $n1 $tcp1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$tcp1 set packet_size_ 1000

The UDP and TCP classes are both child-classes of the class Agent. With the expressions [new Agent/TCP] and [new Agent/UDP] the properties of these classes can be combined to the new objects udp0 and tcp1. These objects are then attached to nodes n0 and n1. Next, the application is defined and attached to the transport protocol. Finally, the configuration parameters of the traffic source are set. In case of CBR, the traffic can be defined by parameters rate_ (or equivalently interval_, determining the interarrival time of the packets), packetSize_ and random_ . With the random_ parameter it is possible to add some randomness in the interarrival times of the packets. The default value is 0, meaning that no randomness is added.

Tracing

A Trace file contains all information needed for animation purposes- both on a static network layout an on dynamic events such as packet arrivals, departures, drops and link failures. Tracing in NS-2 is implemented with the following OTcl code.
To Trace packets on all links following commands are used:
set trace_file [open out.tr w]
$ns trace-all $trace_file
$ns flush-trace
close $trace_file

4. Phase I

To simulate UDP flood attack on dumb-bell topology in NS2 environment and analyzing performance of the network by observing throughput and packet drop in the no attack scenario as well as in the scenario with DDoS attack.

4.1 Flowchart

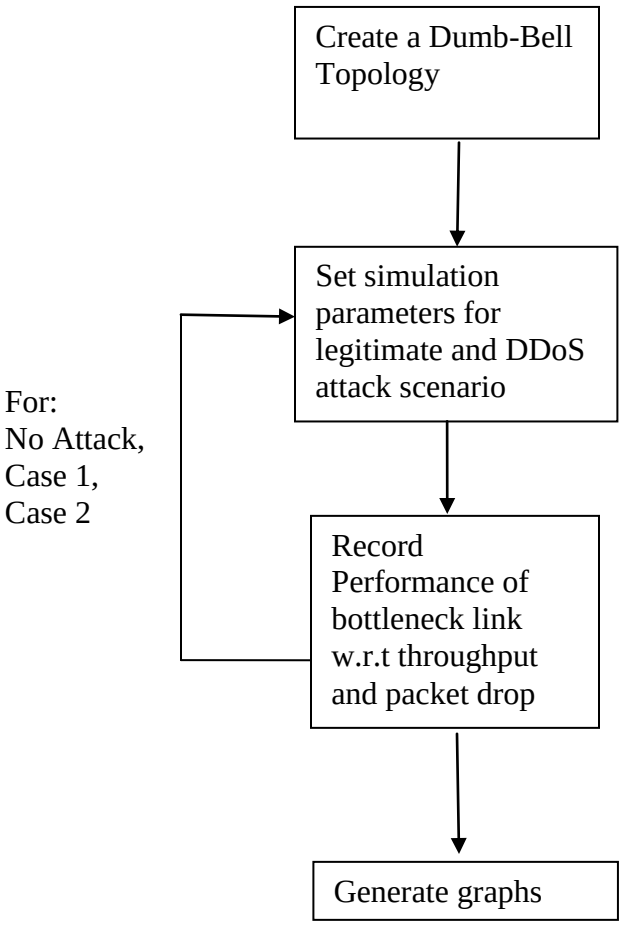

Fig 4.1 Flowchart for simulating DDoS Attack

4.2 Software Requirements:

Software : VMware Player

Simulator : NS version 2.34

Language : OTCL, AWK Scripting

Operating System : Ubuntu 10.10

First Phase

The first phase of this project is to build a topology for DDoS attack network and simulate the DDoS attack .We have used dumb-bell topology(as shown in fig) for creating traffic. In the topology there are few router nodes numbered as 0, 6, 14, 17, 15, 20, 26. Node numbered 18 is the Server and 8 to12, 21 to 25, 27 to 31 are clients. They send legitimate request to server (FTP traffic) .The bandwidth of all links is set to be 100Mbps and the bandwidth of bottleneck link (14-17) is 1.5 Mbps. Attacking nodes are 1 to 5 and they send attack traffic to server .The link between14-17 is bottleneck link. The purpose of attack node is to congest the bandwidth of bottleneck link so that legitimate traffic could not get accessed by the server.

Parameters	Value	
Simulation Time	60 seconds	
No of legitimate clients	15	
No of attack sources	5	
Access Bandwidth	100 Mbps	
Bottleneck Bandwidth	1.5 Mbps	
Attack type	Udp flood	
Attack intervals	Case 1	15-30ms
	Case 2	10-20ms and 40-50ms

Table 4.1 Basic Parameters

4.3 Simulation Result

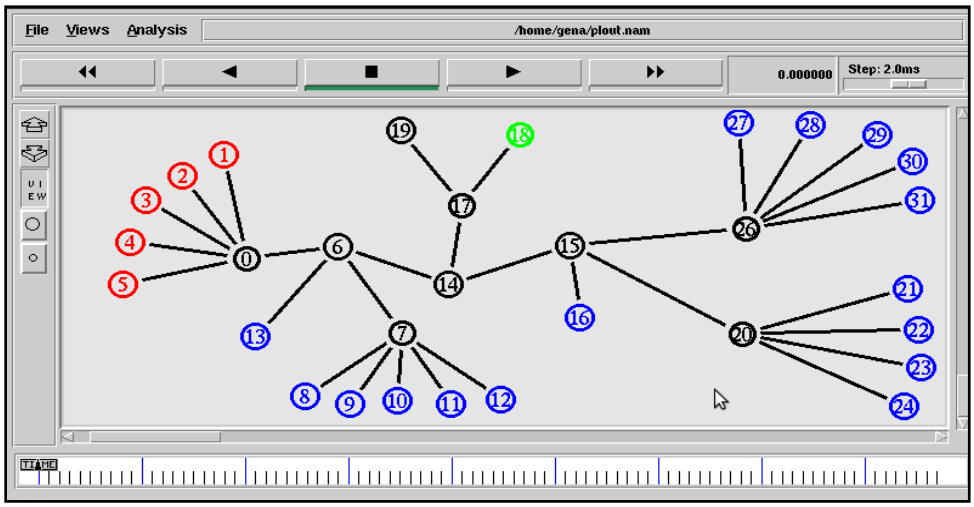

Fig 4.2 Topology

Client node

Router node

Attack node

Server node

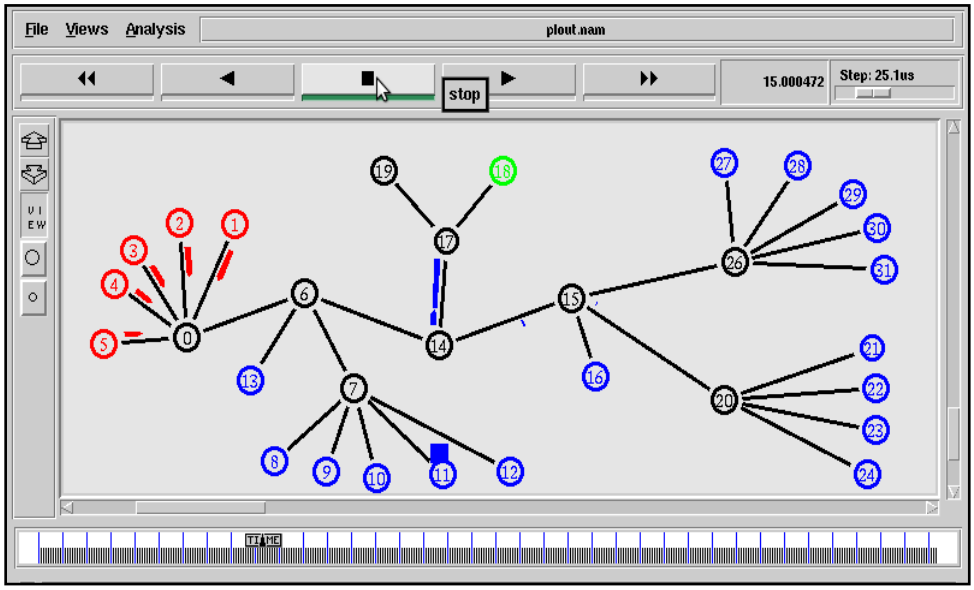

Fig 4.3 Traffic Generation

Trace File Format:

Type Identifier	Time	Source Node	Destn. Node	Packet Name	Packet Size	Flags	Flow ID	Source Addr	Destn. Addr	Seq.no	Packet unique ID

Table 4.2 Trace File Format

4.3.1 Packet Loss

Packet loss is the failure of one or more transmitted packets to arrive at their destination. In the no attack scenario, the number of legitimate packet drop is very less as compared the other two attack scenarios. The number of legitimate packet drop increases in the attack scenario because the link is congested by the attack packet. Hence, dropping out the legitimate packets.

Scenario	No of legitimate packets arriving at bottleneck	No. of TCP packets dropped	% packet drop
No attack	109040	479	0.4392
Scenario Case 1	85676	555	0.6477
Scenario Case 2	80607	750	0.9304

Table 4.3: Packet Drop

4.3.2 Throughput

Throughput is the number of packets successfully delivered per unit time. Throughput is controlled by the available bandwidth and the queue length. Here, in the simulation overall throughput of the link is considered, ignoring the goodput and the badput.

No attack Scenario

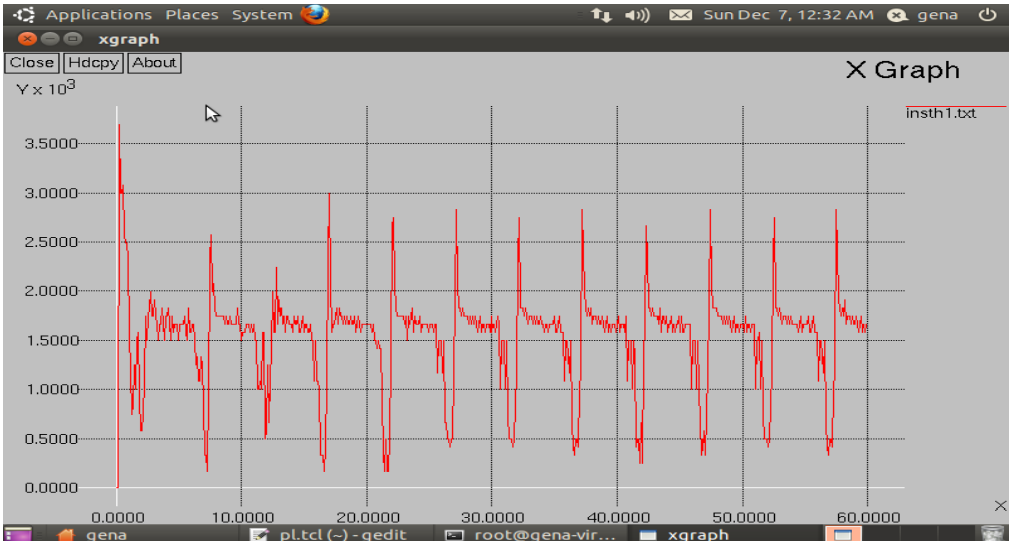

Fig 4.4: Tcp Throughput for No Attack

Scenario Case I

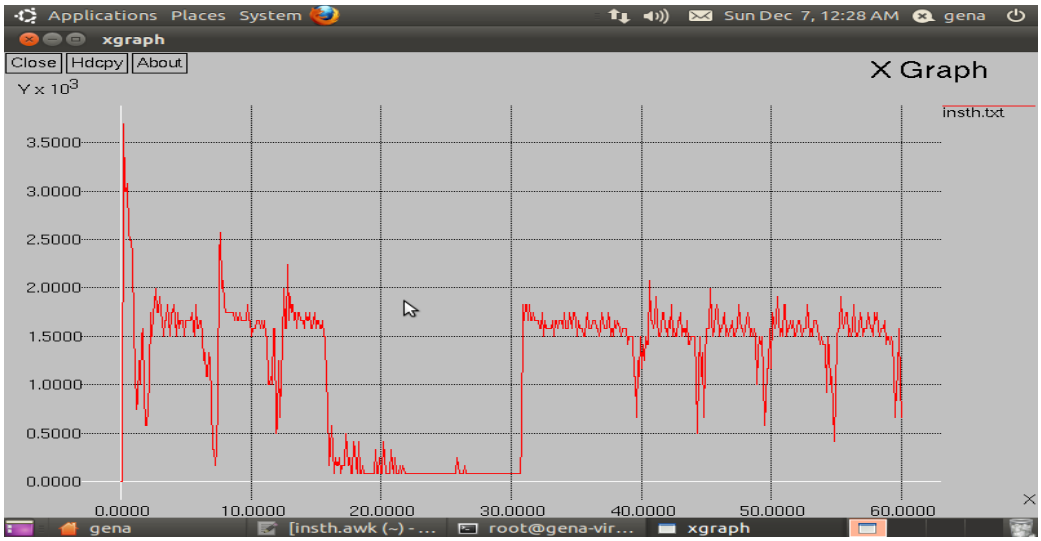

Fig 4.5: Tcp Throughput for Scenario Case I

Tcp Throughput for Scenario Case I decrease in the time interval of 15-30 ms as the UDP packets (malicious packets) are flooded in the network and the server is busy in responding to the attack packets and dropping out the legitimate packets.

Scenario Case II

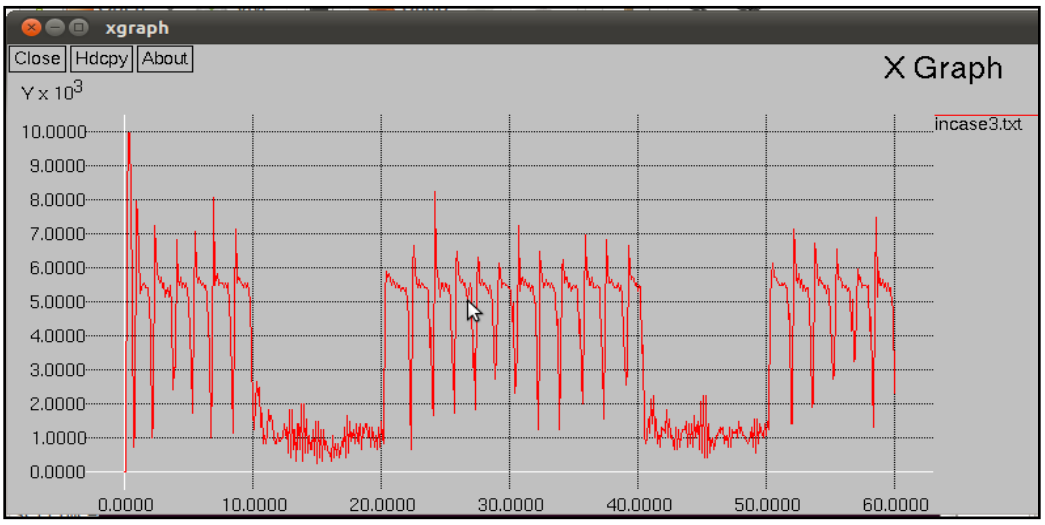

Fig 4.6: Tcp Throughput for Scenario Case II

Tcp Throughput for Scenario Case II decrease in the time interval of 10-20 ms and 40-50 ms as the UDP packets (malicious packets) are flooded in the network and the server is busy in responding to the attack packets and dropping out the legitimate packets .Whereas in the time interval 0-10 ms, 20-40 ms and 50-60 ms the throughput is in accordance with the no attack scenario.

From above results it can be concluded that during the attack although there is a sudden increase in the total throughput of the network but throughput of the legitimate packets decreases drastically .And as soon as attack interval is over, there is a downfall in the throughput for an instant.

Among the above two attack cases, DDoS attack with varying time interval will be more difficult to detect as the change in the values of entropy is very frequent.

5. Phase II

Implementation of detection technique based on Entropy Variation for DDoS attack on the existing dumb-bell topology

5.1 Flowchart

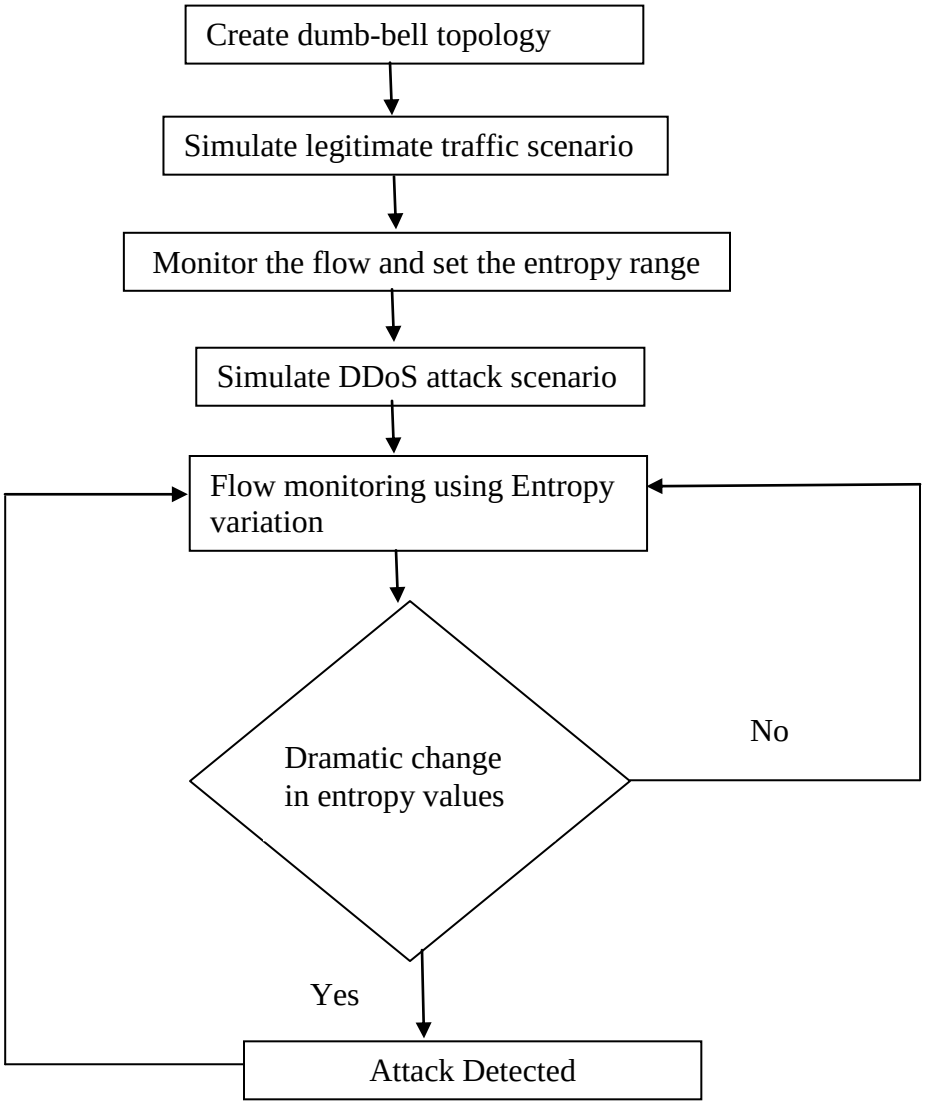

Fig 5.1 Flowchart for simulating DDoS Attack

5.2 Detection Algorithm Based on Entropy

Detection algorithms measure statistical properties of specific fields in the packet headers. For instance, if a detector captures 1000 consecutive packets and computes the frequency of occurrence of each unique source IP address in those 1000 packets, then the detector will have a model of the distribution of the source address allow us to measure the randomness of the addresses.

If the entropy detector determines that the current entropy for some attribute is below the normal range, it suggests that traffic with a relatively small number of values for that attribute is dominating. Since the entropy detector tracks value frequency, it can identify which values are the most common and are likely candidates for rate limiting. For finer targeting, the detector could watch for specific values with dramatic increases in frequency and treat those as suspicious. Conversely, an unusually high entropy value suggests that the low-frequency values are causing trouble, so the detector might suggest that packets having high frequency values be given preferential treatment.

The formula of entropy calculation is as follows:

$$\boldsymbol{H = -\sum_{i=1}^{n} (p_i \ \log_2 \mathrm{p_i})}$$

Where,

$\boldsymbol{p_i}$ is the emergence probability of each distinct source IP address.

n is the total number of packets being analyzed, and

H is the entropy.

The process of computing entropy of *W* packets is as follows:

1. Compute the entropy of the first *W* packets with reference to a specific header parameter (e.g. source IP address).

2. Slide the window so the new first term was previously the second term and the next W_{i-1} consecutive terms are contained in the window.

3. Isolate the term in the summation corresponding to the probability of the symbol acquired from shifting the window.

4. Recompute the affected probabilities for the current window of data.

5. Repeat steps 2-7 to determine subsequent entropy values.

6. Compute threshold from the above calculated entropy values using the equation:

Threshold=Mean ± ß *Standard Deviation

7. If entropy value lies outside the calculated threshold range then there is possibility of suspected attack.

5.3 Simulation Results

5.3.1 Entropy

Entropy for the three scenarios is calculated using the packet window size as one thousand. In figure, x-axis represents the time and y-axis represents the entropy value with respect to packet.

Fig 5.2 Entropy with packet window 1000

Scenario	Range
No Attack	3.35-3.59
Case 1	2.68-4.08
Case 2	2.91-4.09

Table 5.1 Entropy range

While a network is not under attack the value of Entropy falls in a narrow range whereas in case of attack value of Entropy has a wide range. While a network is under DDOS attack the value of Entropy first increases and then decreases in a detectable manner.

5.3.2 Entropy w.r.t. packet window size

In figure, packet window is taken as 10, 100, 500 and 1,000 packets i.e. entropy is computed after every n number of packets. The window size, *W*, is a tunable parameter that controls how much smoothing of short-term fluctuations the detector will do. Increasing *W* will reduce the variation in entropy and may reduce the rate of false positives resulting from brief and presumably insignificant anomalies. With the increase in the packet window size the graph becomes less congested and the entropy values lie in a narrow range. However, *W* should be kept small enough that attacks are detected quickly. For the study, window size of 1,000 packets is a found to be the reasonable compromise in the network environments.

No Attack Scenario

Fig 5.3 Entropy with varying packet window size for no attack

Case1 Attack Scenario

Fig 5.4 Entropy with varying packet window size for case1

Case2 Attack Scenario

Fig 5.5 Entropy with varying packet window size for case2

Table below summarizes the range of entropy values under different window size.

Scenario	Entropy Value Range		
	Window Size 10	Window Size 100	Window Size 1,000
No Attack	0-2.62	2.49-3.51	3.35-3.59
Case1	0-2.49	1.61-4.10	2.68-4.08
Case2	0-3.00	2.60-4.10	2.91-4.09

Table 5.2: Entropy range under different window size

5.3.3 Threshold

Threshold is calculated from the no attack scenario using the equation:

Threshold=Mean ± ß *Standard Deviation, ß is defined to be in the range [3, 6] since it is an observed fact that if ß is over 6, the detector couldn't detect almost any anomalies. On the other hand, the detector is too sensitive to detect an attack precisely producing many false negatives if ß is (0, 2].In the graphs below, the value of 1 on y-axis shows the anomaly detected corresponding to the time on x-axis.

Case1 Attack Scenario

For ß = 4:

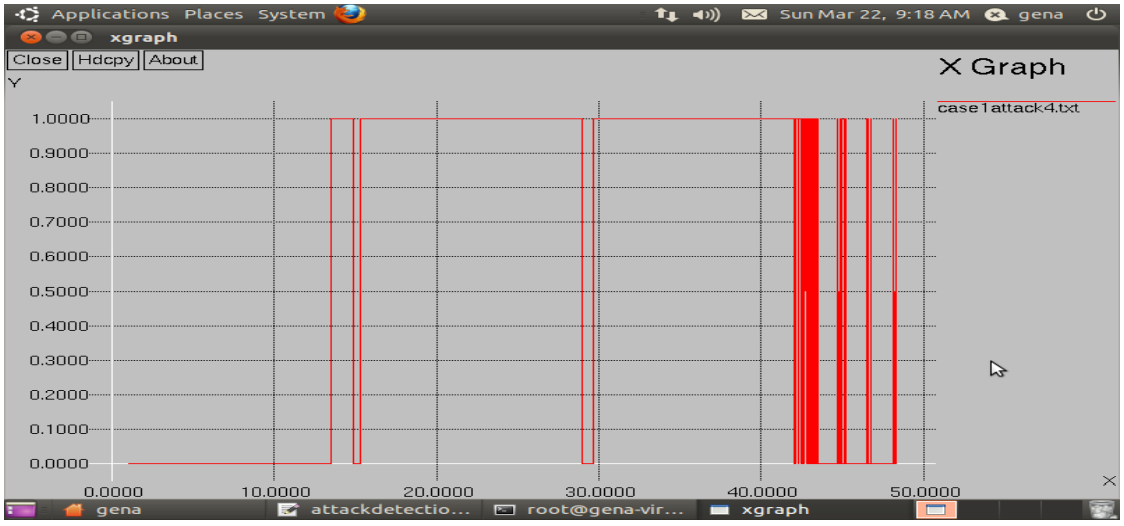

Fig 5.6: Attack detection case 1(ß = 4)

In this case, attack is detected in the time interval 13-28 ms, 30-42 ms and fluctuations in the time interval from 42-48 ms. With ß = 4 there are many false positive alarm raised by the detector which is not a good attribute.

For ß = 6:

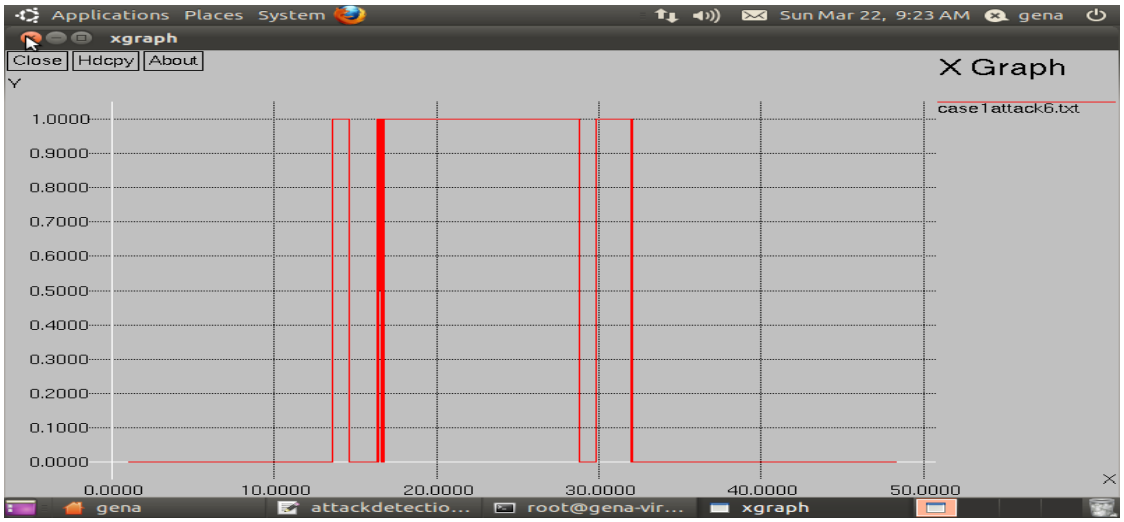

Fig 5.7: Attack detection case 1(ß = 6)

In this case, attack is detected in the time interval 13-28 ms, 30-32 ms. With ß = 6 the number of false positives is comparatively less.

Case2 Attack Scenario

For ß = 4:

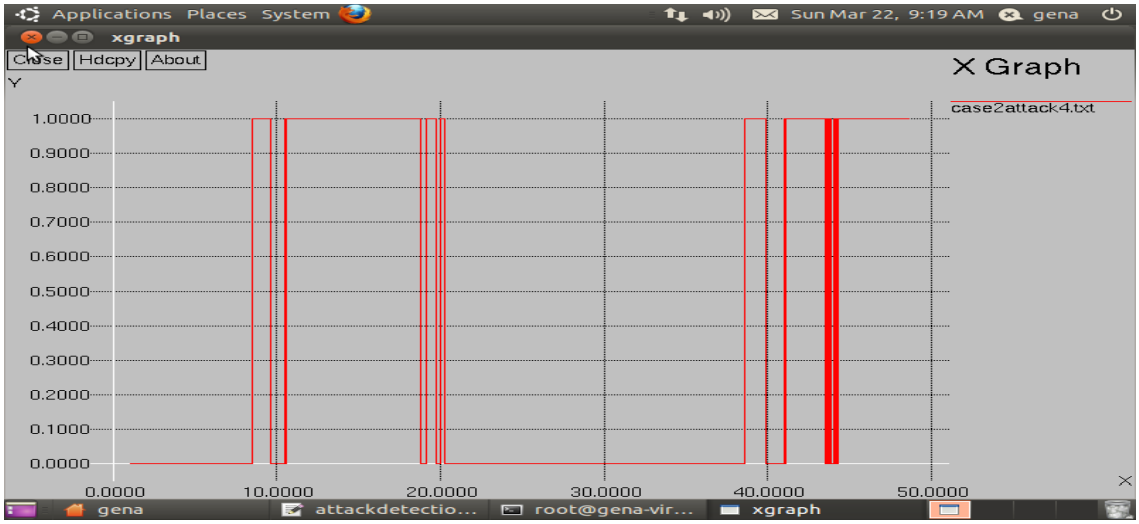

Fig 5.8: Attack detection case 2(ß = 4)

In this case, attack is detected in the time interval 10-20 ms, 39-50 ms with few fluctuations. With ß = 4 there the number of false positives is very less.

For ß = 6:

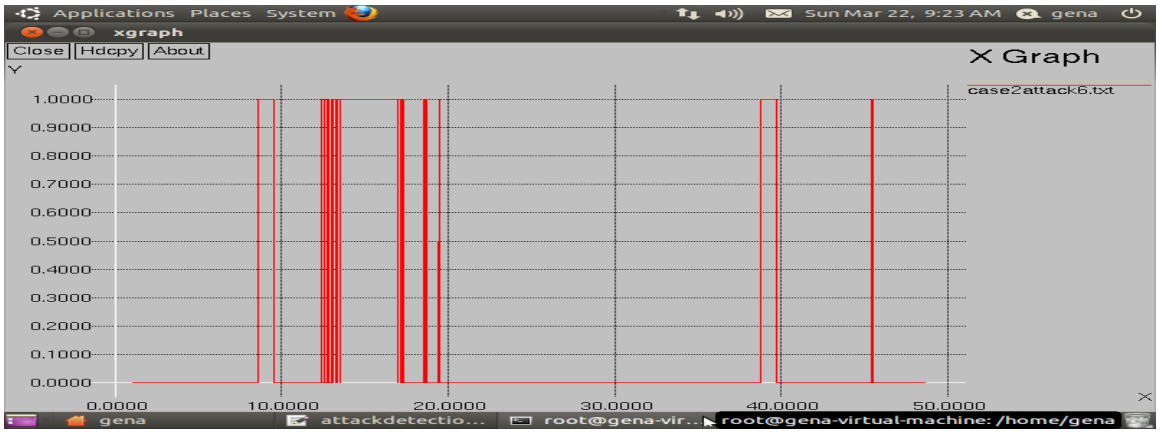

Fig 5.9: Attack detection case 2(ß = 6)

In this case, attack is detected in the time interval 10-19 ms, 39-40 ms and at 45 ms with some fluctuations in between. With ß = 6 there are some false negatives.

5.4 Performance Evaluation

The metrics used to evaluate performance of DDoS detection approach, namely, detection rate (R_d) and false positive alarm rate (R_{fp}). The detection rate (R_d) is the measure of percentage of attacks detected among all actual attacks performed. The detection rate (R_d) is defined as follows:

R_d = d / n

where, d is the number of DDoS detected attacks, and n is the total number of actual attacks generated during the simulation.

The false positive alarm rate (R_{fp}) is the measure of percentage of false positives among all normal traffic event. It is defined as follows:

R_{fp} = f / m

where, f is the number of false positive alarm raised by attack detection mechanism, and n is the total number of normal traffic flow events during the simulation.

Scenario	Performance Evaluation Metric	
	R_d	R_{fp}
Case1($ß$ = 4)	1.71	0.44
Case1($ß$ = 6)	0.99	0.09
Case2($ß$ = 4)	0.95	0.09
Case2($ß$ = 6)	0.54	0.06

Table 5.3: Performance Evaluation II

As $ß$ is defined to be in the range [3, 6] for efficient detection of attacks by the detector. For the present scenario for the value of $ß$ as 6 results are better than the value of $ß$ as 4.

5.5 Adaptive DDOS detector based on entropy approach

Implementing adaptive DDOS detector on the existing dumb-bell topology using the concept of moving averages.

The detection reliability depends on how a system selects a threshold value of adjusting size of moving packet window. In general, the threshold value is heuristically chosen based on the result of passed instruction pattern analysis for non attack scenario. Thus, the using static threshold value calculated from the no attack scenario might be less practical in the real network environment since it is impossible to adjust threshold values whenever a conspicuous flow is detected. For an instance, if the difference between the average of previous monitoring interval and new entropy value is decreased, the detector can't detect the attacks with high threshold value because of the steady channel condition and stealthy attack pattern. In this case threshold value is needed to be decreased. Meanwhile, if the channel is burst but the detector has relatively small threshold value, the detector works very sensitively in this situation. As a result, the detector yields many false positives, a bad characteristic of a detector. In that case, threshold should be increased accordingly. To provide the high reliable and efficient DDoS attack detection

in real network environment, an adaptive DDoS attack detector which can dynamically select the threshold value of moving average window size should be used**.**

5.5.1 Flowchart

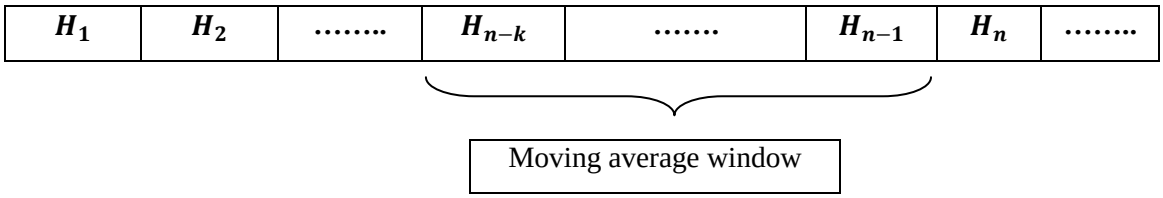

Fig 5.10: Attack monitoring system using moving average concept

μ: i-th average of Moving Average Window
s: Standard Deviation of H_{n-m} ~H_{n-1} with μ_i
D_i: Absolute value of difference between μ_i and H_n
ß: threshold multiplication factor, positive integer value
ω: threshold ($\omega = \beta * s$)

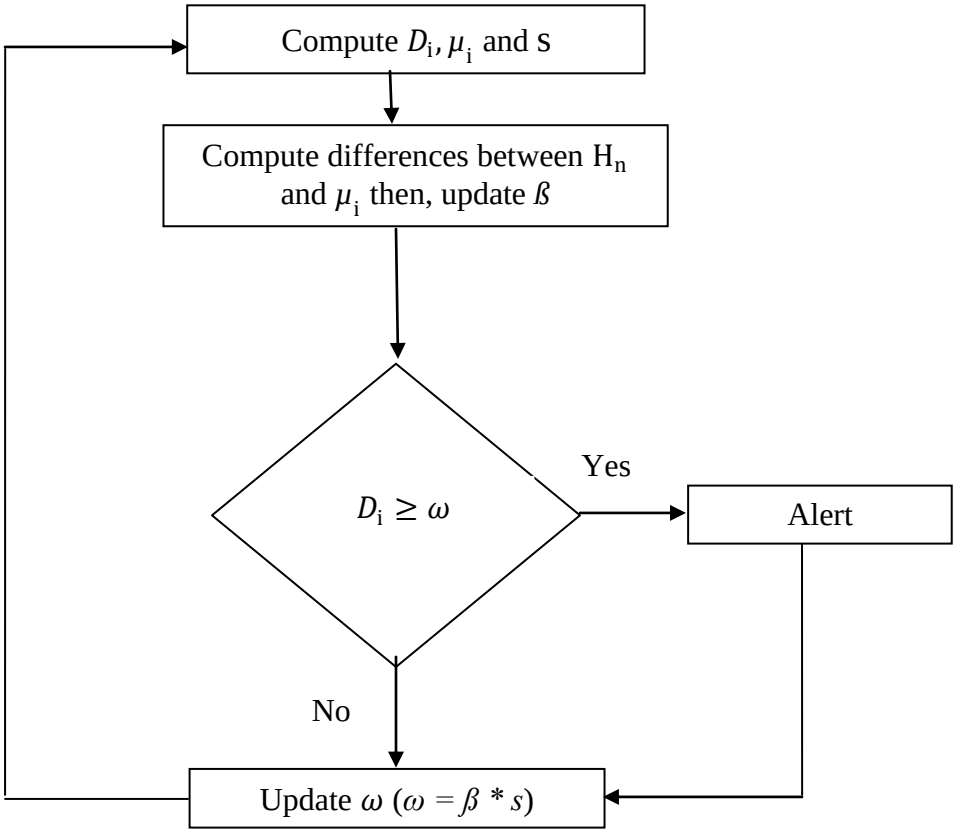

Fig 5.11 Flowchart for adaptive detector

5.5.2 Simulation Result:

Case 1 Attack Scenario

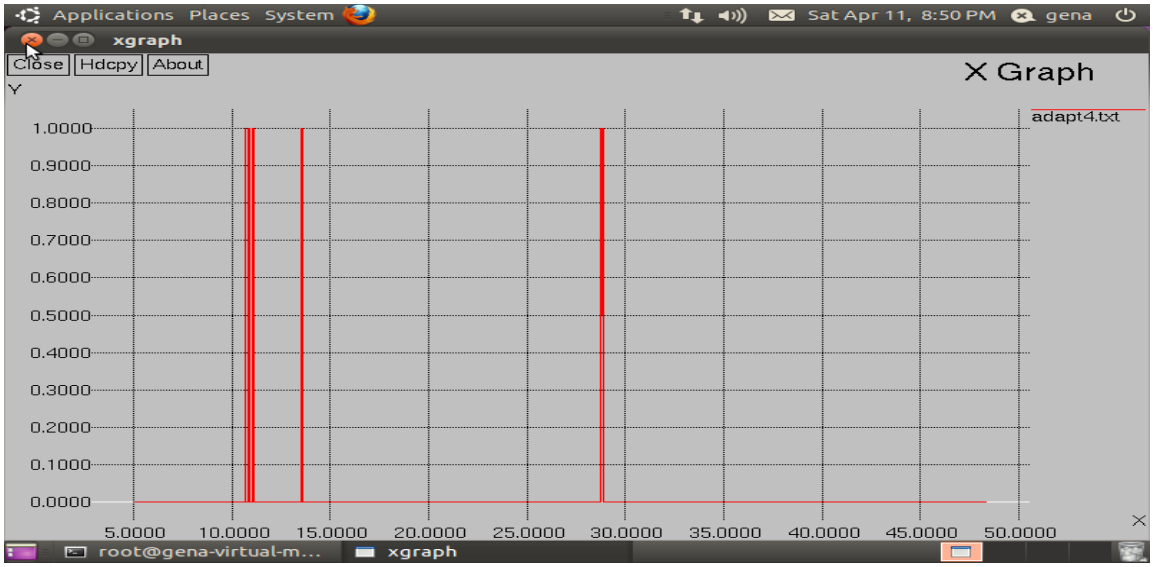

Fig 5.12: Attack detection by adaptive detector case 1

Case 2 Attack Scenario

Fig 5.13: Attack detection by adaptive detector case 2

As observed in above two cases, the result is not satisfactory. The detector raises many false positives as well as negative. For adaptive detector to perform efficiently a large data set should be used to give satisfactory results.

6. Phase 3

Implementing Chi-Square Statistical detection technique for DDOS attack on the existing dumb-bell topology.

6.1 Flowchart

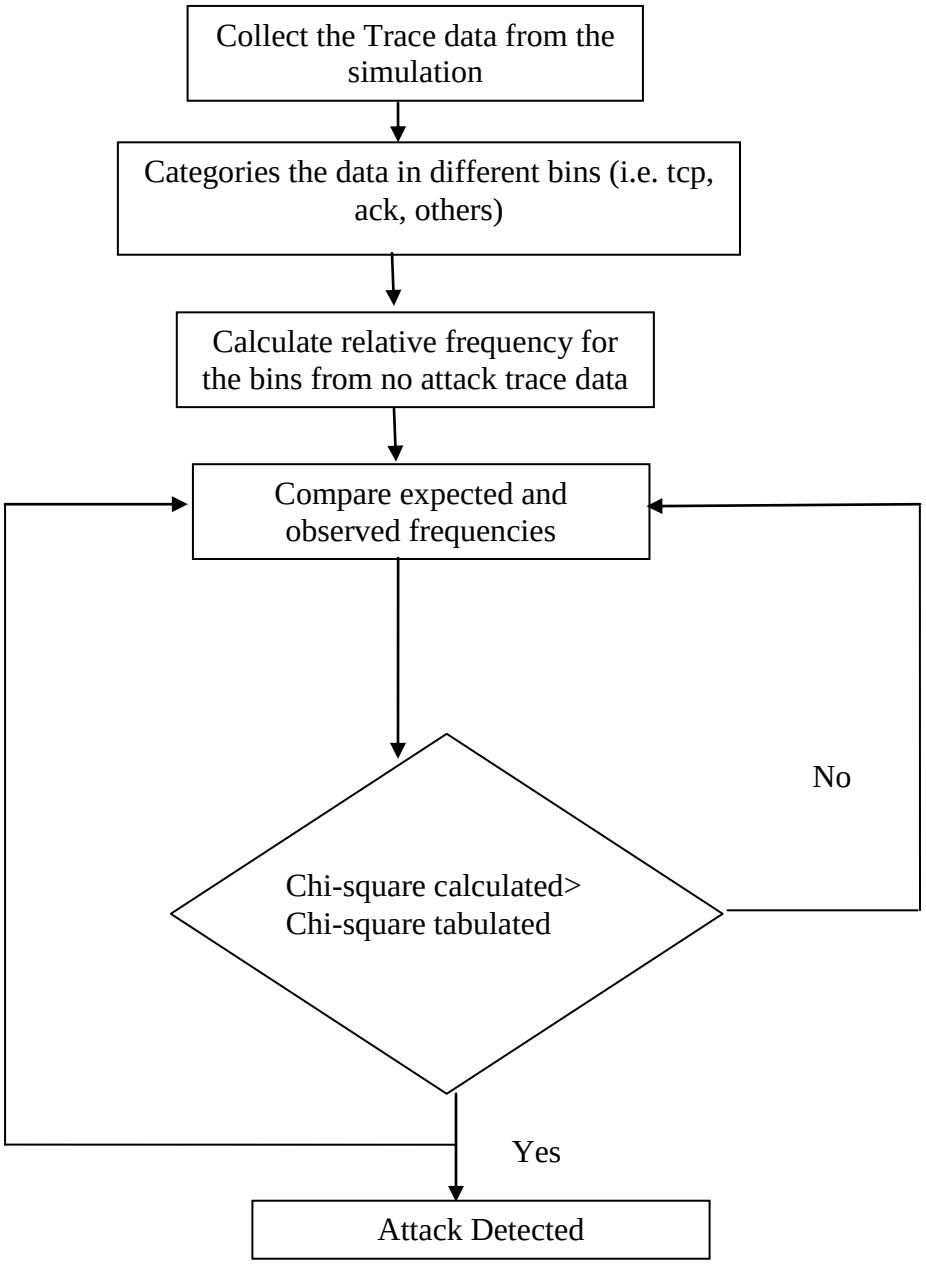

Fig 6.1 Flowchart for Chi-Square Statistic

6.2 Detection Algorithm Based on Chi-Square Statistic

The network trace data contains 60 ms continuous network traffic generated by simulating dumb-bell topology. This captured traffic goes as input to the system. To analyze whole data efficiently at a time, data packets are distributed into1 ms interval. This data is used as input to the next phase. Now packets are categorized in four categories (tcp, syn-ack, dropped packet and packets greater than specific size).Then the packets per microsecond are calculated for each category and relative frequency for each category is calculated. A distribution is made, the distribution is called sample distribution. This sample distribution is also called observed data entries in chi-square statistics. Once a sample distribution is made for the captured traffic, this sample distribution has passed to chi-square statistic for testing. In this testing chi-square calculation is performed on the sample distribution. A chi- square value is calculated. This chi-square value is passed to the decision phase. In decision phase the chi-square calculated value is compared with chi-square tabulated value, which is also called critical value. If the chi-square calculated value is greater than critical value then it means that attack has occurred.

Categories	No. of packets per ms	Categories	Relative Frequencies
TCP	8927	TCP	0.3322
Syn-Ack	9015	Syn-Ack	0.3355
Dropped packets	10	Dropped packets	0.0003
Packets smaller than 1050 bytes	8927	Packets greater than 1000 bytes	0.3322
Total	26,879	Total	1

Table 6.1: Packets Wise Distribution and Relative frequencies

To check anomaly in the data for any time duration say in time slot number 5and 20, T5 and T20 . Following hypothesis for the this test as;

H0: The T5 or T20 has the specified distribution or there is no attack in T5, and
H1: The T5 or T20 does not have specified distribution means there is an attack.

The chi-square(χ^2) test calculation for the case1 T5 is defined in table below :

Categories	Relative Frequencies(f)	Observed Frequencies(O)	Expected Frequencies(E=n*f)	$((O-E)^2$/E
TCP	0.3322	8942	8927	0.25
Syn-Ack	0.3355	9000	9015	0.024
Dropped packets	0.0003	1	8	6
Packets greater than 1000 bytes	0.3322	8927	8927	0
Total	1	N=26870		6.274

Table 6.2: Computation of Chi-Square Test Statistic for the Test of T5

The chi-square(χ^2) test calculation for the case1 T20 is defined in table below :

Categories	Relative Frequencies(f)	Observed Frequencies(O)	Expected Frequencies(E=n*f)	$((O-E)^2$/E
TCP	0.3322	1318	1289	0.65
Syn-Ack	0.3355	1140	1302	20.15

Dropped packets	0.0003	95	2	4324
Packets greater than 1000 bytes	0.3322	1328	1289	1.799
Total	1	N=3,881		4,346.6

Table 6.3: Computation of Chi-Square Test Statistic for the Test of T5

From above table, the χ^2 for the goodness-of-test statistic calculated is:

$\chi^2(T5) = (O – E)^2/E = 6.274$
$\chi^2(T20) = (O – E)^2/E = 4326.6$

Let the hypothesis test is performed at 5% significance level so (alpha=0.05). There are 4 types of categories in the test so k= 4 and the degree of freedom (df) is 4 -1 =3, so checking the chi square table, with alpha=0.05 and df=3 :

Chi-square tabulated value of T5= χ^2 (0.05) = 7.82
Chi-square Calculated Value of T5= χ^2 =6.274
Chi-square tabulated value of T20= χ^2 (0.05) = 7.82
Chi-square Calculated Value of T20= χ^2 =4326.6

Hence the chi-square calculated value is smaller than chi-square tabulated value for T5, so the null hypothesis H0 is accepted and H1is rejected means that there is no attack or anomaly in the network traffic at time slot T5. In others words we can also say that there is no denial-of-service attack. But if calculated value is greater, as for T20 it implies that there was an attack in that interval so the null hypothesis H0 is rejected and H1 is accepted.

6.3 Simulation Result

Case 1 Attack Scenario

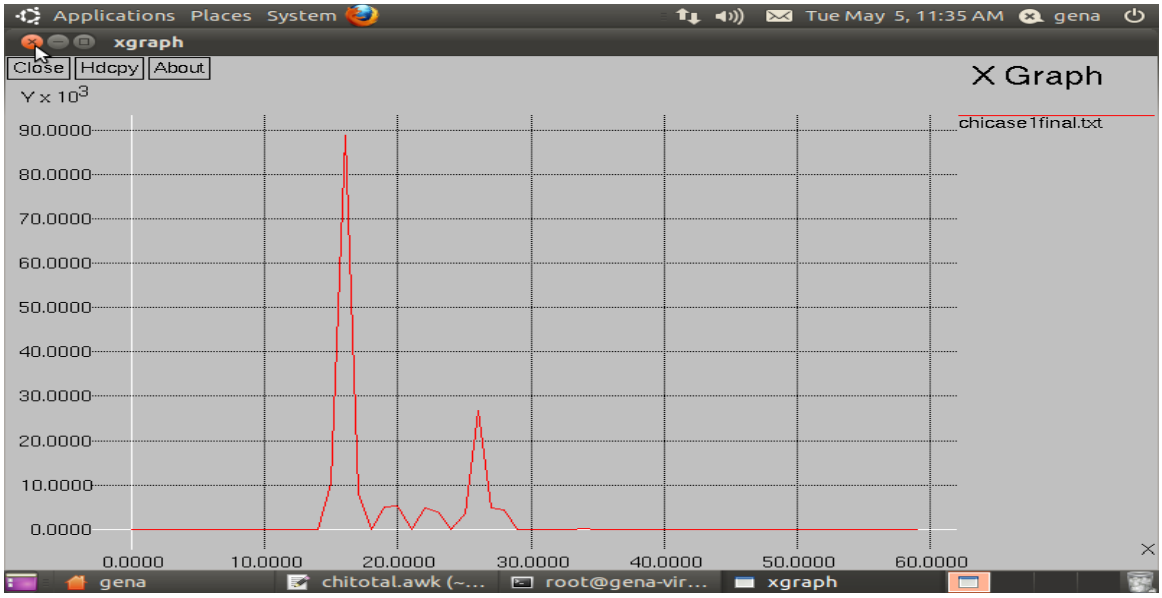

Fig 6.2: Attack detection using Chi-Square statistic(case1)

Case 2 Attack Scenario

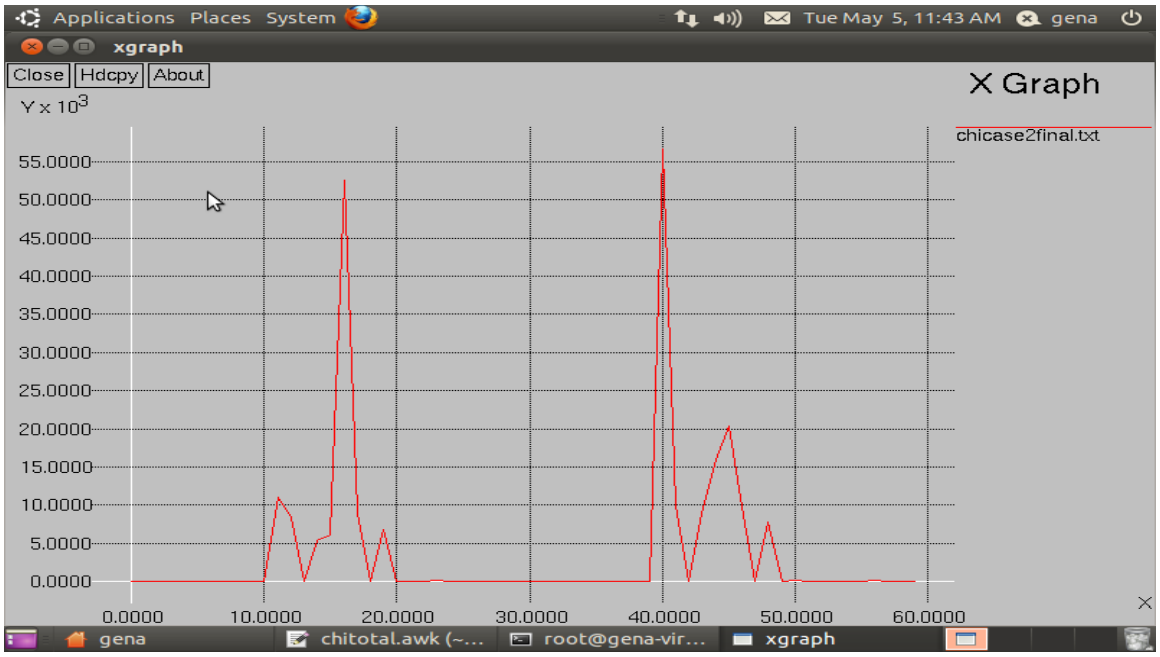

Fig 6.3: Attack detection using Chi-Square statistic(case2)

From the above two graphs, it can be concluded that there was an anomaly in the time interval 15-30ms in case first and 10-20ms and 40-50ms in case second as the chi-square value is comparatively large than the tabulated values.

6.4 Performance Evaluation

The metrics used to evaluate performance of Chi-Square Statistics detection approach, namely, detection rate (R_d) and false positive alarm rate (R_{fp}). The detection rate (R_d) is defined as follows:

R_d = d / n

where, d is the number of DDoS detected attacks, and n is the total number of actual attacks generated during the simulation.

The false positive alarm rate (R_{fp}) is defined as follows:

R_{fp} = f / m

Where , f is the number of false positive alarm raised by attack detection mechanism, and n is the total number of normal traffic flow events during the simulation.

Scenario	Performance Evaluation Metric	
	R_d	R_{fp}
Case1	1.5	0.46
Case2	3.18	0.56

Table 6.4 Performance Evaluation II

In Chi-Square statistical approach for DDOS detection the performance is comparatively low as compared to detection using Entropy Variation approach. Chi-Square distribution is of limited utility because when the expected and the observed values are large and total measurements are independent, this value follows the well-known chi-square distribution, with k-1 degrees of freedom. These assumptions (in particular, independence) do not typically hold for packet field values even under normal conditions. However, the chi-square statistic does provide a useful measure of the deviation of a current traffic profile from the baseline.

6.5 Comparison of Entropy And Chi-Square Approach

Performance in case 1

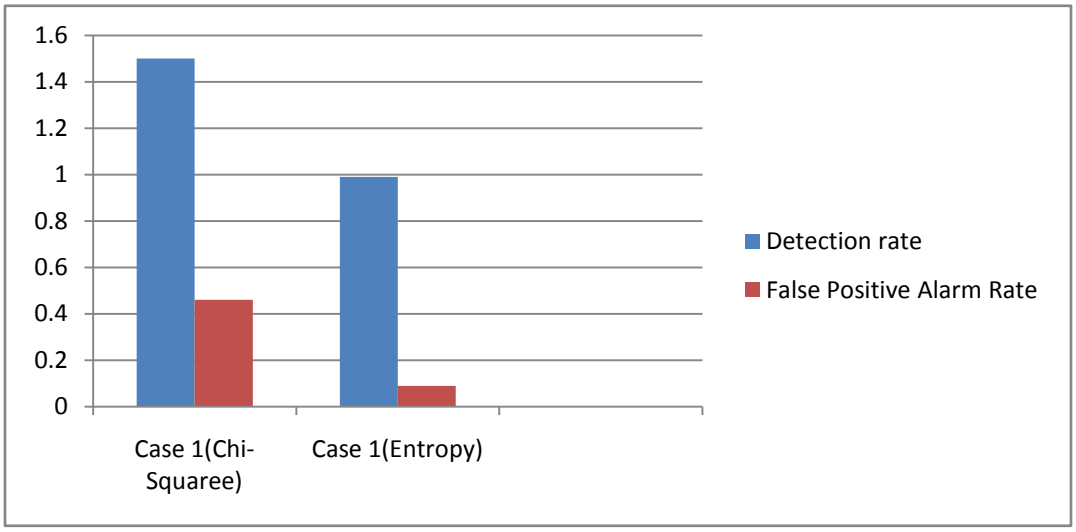

Fig 6.4 Performance of Chi-Square & Entropy I

Performance in case 2

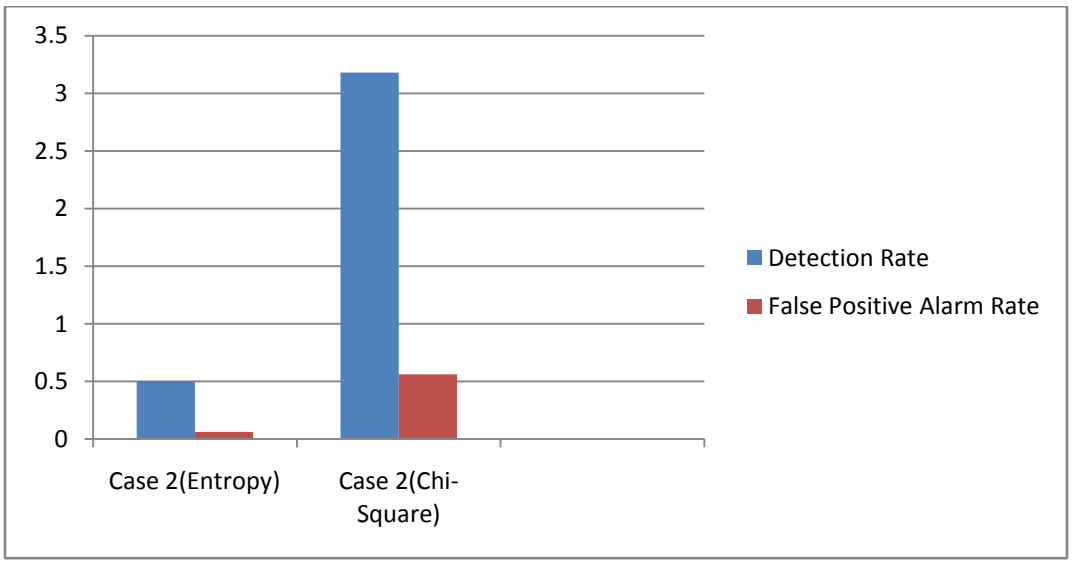

Fig 6.5 Performance of Chi-Square & Entropy II

From above two graphs, it can be observed that performance wise Entropy Variation approach is better than the Chi-Square Approach for detecting DDOS attack. The observed detection rate and false alarm rate in case of entropy variation is in accordance with that of the expected. The expected detection rate should be nearly 1 and the false positive alarm rate should be as minimum as possible. Whereas, in the case of chi-square there is a large deviation making it less efficient than the entropy. The reason for the deviation from the expected values can be the assumptions (in particular, independence of total number of packets) which do not typically hold for packet field values under normal conditions. Secondly, the very low baseline frequency value for a bin might have excessively influenced the chi-square value.

7. Conclusion

Global scenario of DDoS attacks can be clearly visualized from the study. Denial of service attacks are a real threat to the operation of any networked computer system. While they can be difficult to detect and react to, prudent planning and preparation can mean the difference between a total shut down of the organization and a slight inconvenience. Statistical detection mechanisms can provide the victim with a solid base for minimizing the impact of these potentially damaging attacks. During the attack although there is a sudden increase in the total throughput of the network but throughput of the legitimate packets decreases drastically .And as soon as attack interval is over, there is a downfall in the throughput for an instant. This implies denial of service to the legitimate clients. While a network is not under attack the value of Entropy falls in a narrow range whereas in case of attack value of Entropy has a wide range. And, window size (W) is a tunable parameter; which may reduce the rate of false positives resulting from brief and presumably insignificant anomalies. Entropy Variation is a better detection technique as compared to the Chi-Square. But, in the case of ON-OFF type of attack the detection rate of Entropy Variation technique is less as compared to the continuous attack scenario. In this type of scenario, Adaptive technique can be proved to be effective.

8. Future Work

- The focus so far has been on detection algorithms and the implementation of these algorithms on the smaller network using dumbbell topology in NS2 continuing, the implementation of same algorithms on the larger network and to check for scalability.
- Secondly, increasing the time duration and the checking whether these algorithms can reliably show the same result when implemented for larger duration.
- While our initial goal was to provide effective statistical detection mechanism against DDoS attacks, and will be further continuing to explore techniques for better defense against future stealthy attacks. Making the threshold calculation adaptive such that that the value of ‘beta’ used for calculating threshold is automatically adjusted according to the traffic stream, i.e. making detector adaptable.
- Another future task will be to make these algorithms computationally less expensive in terms of time and space. So that a detector with such capabilities could more effectively allocate its limited computational resources effectively and perform efficiently and fastly.

9. REFERENCES

1) Shaveta Gupta, Dinesh Grover, Abhinav Bhandari, "*Detection Techniques against DDoS Attacks: A Comprehensive Review*", International Journal of Computer Applications (0975 – 8887) Volume 96– No.5, June 2014
2) Reddybathini Durga Siva Prasad "*A Robust Mechanism to Mitigate DDoS Attack Using Entropy Variation*." December – January 2014, 762 - 766.
3) K. Munivara Prasad, A. Rama Mohan Reddy & K.Venugopal Rao, "*DoS and DDoS Attacks: Defense, Detection and Traceback Mechanisms*" Global Journal of Computer Science and Technology: E Network, Web & Security, Volume 14 Issue 7 Version 1.0 Year 2014
4) Daljeet Kaur and Monika Sachdeva, "*Study Of Flooding Based DDoS Attacks And Their Effect Using Deter Testbed*", IJRET Volume: 02 May-2013.
5) Monowar H. Bhuyan, H. J. Kashyap and J. K. Kalita, D. K. Bhattacharyya "*Detecting Distributed Denial of Service Attacks: Methods, Tools and Future Directions*", December 2012.
6) Yogesh Kumar Meena, Aditya Trivedi, "*A Novel Protocol for IP Traceback to Detect DDoS Attack*", IJCSI, July 2012
7) Brajesh Kashyap "*DDoS Attack Detection and Attacker Identification*." International Journal of Computer Applications (0975 – 8887) Volume 42– No.1, March 2012.
8) Rahul Rastogi, Zubair Khan, M.H. Khan "*Network Anomalies Detection Using Statistical Technique : A Chi-Square approach*" IJCSI (1694-0814) Volume 9, March 2012
9) Altyeb Altaher, Sureswaran Ramadass and Ammar Almomani "*Real Time Network Anomaly Detection Using Relative Entropy*" , IEEE 2011.
10) B.B. Gupta, Manoj Mishra and R.C. Joshi "*An ISP Level Solution to Combat DDOS Attacks Using Combined Statistical Based Approach*", Journal of Information Assurance and Security, 2008.
11) Laura Feinstein, Dan Schnackenberg, Ravindra Balupari, Darrell Kindred, *Statistical Approaches to DDoS Attack Detection and Response*", IEEE 2003.

12) Christos Douligeris, Aikaterini Mitrokotsa "*DDoS attacks and defense mechanisms: classification and state-of-the-art*", October 2003.

13) Mohd. Jameel Hashmi, Manish Saxena and Dr. Rajesh Saini, "*Classification of DDoS Attacks and their Defense Techniques using Intrusion Prevention System*" International Journal of Computer Science & Communication Networks,Vol 2(5), pp 607-614

14) J Mirkovic, P Reiher "*A taxonomy of DDoS attack and DDoS defense mechanisms*".

15) Sihan Qing, Hideki Imai, Guilin Wang, "*Information and communications Security*" 9th International Conference, ICICS 2007 Proceedings, 452-461.

16) Arrems Hua, Shih-Liang Chang, "*Algorithms and Architectures for Parallel Processing*", 9th International Conference, ICA3PP 2009 Proceedings, 266-270.

17) The ns Manual: http://www.isi.edu/nsnam/ns/doc/index.html

18) Cisco, NetRanger Overview http://www.cisco.com/univercd/cc/td/doc/product/iaabu/csids/csids1/csidsug/oveview.htm

19) Cisco, "Strategies to Protect Against Distributed Denial of Service Attacks." http://www.cisco.com/c/en/us/support/docs/security-vpn/kerberos/13634-newsflash.html

20) Computer Incident Advisory Capability, "Network Intrusion Detector Overview." http://www.cert.org/flocon/2008/presentations/balland_flocon2008.pdf

21) The Open Source Network Intrusion Detection System: Snort, http://www.snort.org

22) CERT Coordination Center, Denial of Service Tools. http://www.cert.org/advisories/CA-1999-17.html

23) CERT Coordination Center, IP Denial-of-Service Attacks. http://www.cert.org/advisories/CA-1997-28.html

24) CERT Coordination Center, Trends in Denial of Service Attack Technology. http://www.cert.org/archive/pdf/DoS_trends.pdf

25) CERT Coordination Center, Type of attacks.

http://www.cert.org/advisories/CA-1998-01.html

26) Defending Against The Next Generation Distributed Denial of Service DDoS attacks: DDoS Defense Reference Architecture http://www.drchaos.com/defending-against-the-next-generation-distributed-denial-of-service-ddos-attacks-ddos-defense-reference-architecture/

27) Chi-Square Goodness-of-Fit Test http://www.itl.nist.gov/div898/handbook/eda/section3/eda35f.html

25) “2013 Botnets And DDOS Attack Report” by Huawei Data Centre.

10. APPENDIX

Chi-Square Distribution Table

df	$\chi^2_{.995}$	$\chi^2_{.990}$	$\chi^2_{.975}$	$\chi^2_{.950}$	$\chi^2_{.900}$	$\chi^2_{.100}$	$\chi^2_{.050}$	$\chi^2_{.025}$	$\chi^2_{.010}$	$\chi^2_{.005}$
1	0.000	0.000	0.001	0.004	0.016	2.706	3.841	5.024	6.635	7.879
2	0.010	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210	10.597
3	0.072	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345	12.838
4	0.207	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277	14.860
5	0.412	0.554	0.831	1.145	1.610	9.236	11.070	12.833	15.086	16.750
6	0.676	0.872	1.237	1.635	2.204	10.645	12.592	14.449	16.812	18.548
7	0.989	1.239	1.690	2.167	2.833	12.017	14.067	16.013	18.475	20.278
8	1.344	1.646	2.180	2.733	3.490	13.362	15.507	17.535	20.090	21.955
9	1.735	2.088	2.700	3.325	4.168	14.684	16.919	19.023	21.666	23.589
10	2.156	2.558	3.247	3.940	4.865	15.987	18.307	20.483	23.209	25.188
11	2.603	3.053	3.816	4.575	5.578	17.275	19.675	21.920	24.725	26.757
12	3.074	3.571	4.404	5.226	6.304	18.549	21.026	23.337	26.217	28.300
13	3.565	4.107	5.009	5.892	7.042	19.812	22.362	24.736	27.688	29.819
14	4.075	4.660	5.629	6.571	7.790	21.064	23.685	26.119	29.141	31.319
15	4.601	5.229	6.262	7.261	8.547	22.307	24.996	27.488	30.578	32.801
16	5.142	5.812	6.908	7.962	9.312	23.542	26.296	28.845	32.000	34.267
17	5.697	6.408	7.564	8.672	10.085	24.769	27.587	30.191	33.409	35.718
18	6.265	7.015	8.231	9.390	10.865	25.989	28.869	31.526	34.805	37.156
19	6.844	7.633	8.907	10.117	11.651	27.204	30.144	32.852	36.191	38.582
20	7.434	8.260	9.591	10.851	12.443	28.412	31.410	34.170	37.566	39.997
21	8.034	8.897	10.283	11.591	13.240	29.615	32.671	35.479	38.932	41.401
22	8.643	9.542	10.982	12.338	14.041	30.813	33.924	36.781	40.289	42.796
23	9.260	10.196	11.689	13.091	14.848	32.007	35.172	38.076	41.638	44.181
24	9.886	10.856	12.401	13.848	15.659	33.196	36.415	39.364	42.980	45.559
25	10.520	11.524	13.120	14.611	16.473	34.382	37.652	40.646	44.314	46.928
26	11.160	12.198	13.844	15.379	17.292	35.563	38.885	41.923	45.642	48.290
27	11.808	12.879	14.573	16.151	18.114	36.741	40.113	43.195	46.963	49.645
28	12.461	13.565	15.308	16.928	18.939	37.916	41.337	44.461	48.278	50.993
29	13.121	14.256	16.047	17.708	19.768	39.087	42.557	45.722	49.588	52.336
30	13.787	14.953	16.791	18.493	20.599	40.256	43.773	46.979	50.892	53.672
40	20.707	22.164	24.433	26.509	29.051	51.805	55.758	59.342	63.691	66.766
50	27.991	29.707	32.357	34.764	37.689	63.167	67.505	71.420	76.154	79.490
60	35.534	37.485	40.482	43.188	46.459	74.397	79.082	83.298	88.379	91.952
70	43.275	45.442	48.758	51.739	55.329	85.527	90.531	95.023	100.425	104.215
80	51.172	53.540	57.153	60.391	64.278	96.578	101.879	106.629	112.329	116.321
90	59.196	61.754	65.647	69.126	73.291	107.565	113.145	118.136	124.116	128.299
100	67.328	70.065	74.222	77.929	82.358	118.498	124.342	129.561	135.807	140.169