Quantifying Severity of Flooding Denial of Service Attacks on Internet Enabled Network Performance Metrics

Elisha M. Nkuba
Dept. of ECE
Sri Venkateswara College of Engineering and Technology,
Chittoor, India

Ashwin. JS
Dept. of ECE
Sri Venkateswara College of Engineering and Technology,
Chittoor, India

***Abstract*— There are dozens of studies have been done related to networking security, in particular, network performance, but most of them focus on traffic generation, statistic's collection and graphical visualizations. In these, however, very few tends to quantify the network performance metrics, even though they quantify, they rely on computing an average or the total value. The value averaged over the long interval does not indicate at what particular time of the modeled attack, the effect is higher or less so as it can be intervened. In this paper, we present a new approach to quantify severity imposed by flooding attacks on network performance.**

We used network simulator version2 software to pre-process the object terminal command language scripts developed to generate, and statistic's collection of file transfer protocol and constant bit rate traffics. Simulation were done in two scenarios; attack-free and attack scenario. We further ported the trace file got from network simulator version2 onto the automated post - processing tools to obtain both graphical visualized details and quantified values of the instantaneous throughput and instantaneous delay.

Results indicated that the instantaneous throughput during attack is less compared against the values obtained before attack. On the other hand, we found that instantaneous delay during the attack scenario is higher than the delay experienced by the traffic before attack. We computed the intensity of severity caused by the attack on throughput and delay and concluded that, the severity is highly depended on throughput and delay. Therefore, denial of service is likely to occur at some points during attack on file transfer protocol and constant bit rate traffic. We believe that we have found an innovative benchmark approach to evaluation of network performance.

Keywords*— *Denial of service, Performance metrics, flooding, severity, TCP, UDP, ns2, APP

I. INTRODUCTION

Internet connectivity has already changed many aspects of the lives of individuals in developed economies: - creating new ways to communicate and socialize, new business models and industries, and more efficient ways for firms and workers to operate. Innovations such as social media, search engines and on-line market places as well as the convergence of these platforms with nearly all existing activities have permeated society and impacted productivity, economic growth and business creation. These platforms have also changed the way in which public services, healthcare and education are provided and shared. Many governments have recognized the role of internet access in enabling economic activities and social developments and have set out ambitious plans to promote investments in internet access Internet connected things are multiplying rapidly; but serious vulnerabilities are reported to attack cars, medical devices, and more others [50]. In the report it was recommended that manufacturers need to prioritize security to reduce risk of serious personal, economic and social consequences. The predicted consequences as the result of weak security agreed with the findings reported by [40] which contends that most of Denial of services are costly, in its report an average of $217,115 for the five years from 2011 to 2015 were incurred.

Denial of Service is the deliberate attempt to prevent or inhibits the normal use or management of communication facilities. The attempt may have a specific target; for example: to suppress all messages directed to a particular destination. Also can be of the form of denial of service by which there is a disruption of the entire network either by disabling the network or overloading it with messages aimed at degrading the performance.

A. Motivation

A survey on the taxonomies of Denial of service attacks [1], [14], [25] shows that there is no single means by which the Internet enabled network can be compromised. The malicious could focus on compromising the network through exploiting weakness, attack distribution, attack traffic dynamics and others as classified in the studies. Precisely; denial of service attacks exploited weaknesses in the Internet protocols, applications, operating systems, and protocol implementation in operating systems. Malicious activities that interrupt the normal operations on the network are becoming more sophisticated. These are evidenced in incidents, which led to taking down banking websites, trade market websites, e-commerce websites and others whose implication led to financial loss for both users and service providers' implication led to financial loss for both users and service providers.

A great deal of studies has been conducted regarding network security on denial of service attacks [21], [29], [53], [30],[61],[62] but most of them focus on traffic generation, statistic's collections and graphical visualization and few tended to quantify network performance Metrics, or the results obtained are not completely quantitative, which are difficult to interpret. This project intends to analyze quantitatively the severity of flooding based denial of service attacks on network performance metrics, hence severity imposed by the selected flooding based denial of service attacks, which exploit the weakness of Internet protocol.

B. Objectives

The principal goal of this paper was to quantify severity of flooding based denial of service attacks on network performance metrics. This foremost goal had the associated particular objectives whose accomplishment contributed in the field of network security, in particular, flooding denial of service attacks. These specific objectives especially were to:

- Examine Transport Control Protocol Synchronize (SYN) flooding Denial of Service Attacks
- Examine User Datagram Protocol flooding Denial of Service Attacks.
- Quantify the Network Instantaneous throughput and delay performance metrics of TCP Synchronize (SYN) and User Datagram Protocol flooding Denial of Service Attacks using simulation platform ns2.
- Determine the severity of flooding based denial of service attacks on network Instantaneous throughput and delay performance metrics.

I. INTERNET TRANSPORT LAYER PROTOCOLS

Today the Internet and World Wide Web (www) are familiar terms to millions of people all over the world. Many people depend on applications enabled by the Internet such as electronic mail and web access. In addition, addition, the increase in popularity of the business applications places additional emphasis on the Internet. Different applications substantially in the quality of service requirements, hence they require the use of different but related the Internet protocols. For this paper Transmission Control Protocol and User Datagram Protocol which belong to Transport layer, have been taken into account. Understanding the theoretical concepts and mechanisms built behind those protocols helps to figure out how the malicious activities can be launched to compromise the network.

Internet Protocol/ Transmission Control Protocol (IP/TCP)

Transmission Control Protocol is one of the basic protocol of the Internet protocol suite which is part of protocols operates at the Transport layer. This does provide various functions such as dependable in sequence delivery of a stream of bytes from a program on one computer to another program on the other computer, flow control and congestion control [27].The protocol employs a range of mechanisms, which are responsible in ensuring that the services satisfy the applications' requirements. To ensure the reliable delivery of the data over the network, the Transmission Control Protocol employs a mechanism where the sender maintains a buffer, called a sliding window, of data that has been sent to the receiver. A receiver replies to receive data by sending acknowledgement (ACK) packets.

Functions of Transmission Control Protocol

Transmission Control Protocol being an end- to- end protocol at the transport layer performs a number of functions as outlined below [27].

- Data transfer

The Transmission Control Protocol can transfer a continuous stream of data between the users in the form of segments for transmission through the network.

- Reliable delivery

The Transmission Control Protocol must have the ability to recover from data that may be damaged, lost or may be duplicated over the network. This is done by assigning a sequence number to each segment being transmitted over the network and receiving a positive acknowledgment (ACK) on successful delivery. If the ACK is not received within a specific time interval, the data is retransmitted. With the use of sequence numbers, the receivers end order segments in correct sequence, that may be received out of order and to avoid duplicate packets. Damage is handled in Transmission Control Protocol by adding a checksum to each segment, which is being forwarded lastly the checking is done at the receiver, and the damaged segments are then ultimately discarded.

- Flow control

Transmission Control Protocol provides a mechanism that helps the receiver to control the amount of data sent by the sender. This is done by returning a "window" with every ACK packet indicating the acceptable range of sequence numbers beyond the last segment successfully received.

- Multiplexing

Transmission Control Protocol provides a set of ports within each host so that many processes within a single host can use Transmission Control Protocol communication facilities simultaneously. When it is concatenated with the network and host addresses, this forms a socket. The pair of sockets uniquely identifies each connection. Thus, a socket is simultaneously used for multiple connections.

- Connections

A connection is a combination of sockets, sequence numbers, and window sizes. Every connection is uniquely specified by a pair of sockets identifying it’s both sides. Whenever the two processes want to communicate, their TCPs has to first establish a connection (initialize the status information on both sides). Once the communication is complete, the connection has to be terminated or closed.

Transmission Control Protocol segment

- Source and Destination port

The Transmission Control Protocol segment uses the source and destination port number for identification of the sending and receiving end respectively. In many cases, they are associated with the source and destination IP address by combining them. It results into a technical term known as socket, which is IP address plus port number equal to socket.

- Sequence number

The 32-bit long field which identifies a count of the byte in the stream of data from the sending Transmission Control Protocol to the receiving Transmission Control Protocol.

<table>
<tr><td colspan="8">Source port address (16 bits)</td><td>Destination port address(16bits)</td></tr>
<tr><td colspan="9">Sequence number (32 bits)</td></tr>
<tr><td colspan="9">Acknowledge number (32 bits)</td></tr>
<tr><td>Hlen
(4
bits)</td><td>Reserved
(6 bits)</td><td>U
R
G</td><td>A
C
K</td><td>P
S
H</td><td>R
S
T</td><td>S
Y
N</td><td>F
I
N</td><td>Window Size</td></tr>
<tr><td colspan="8">Check Sum</td><td>Urgent Pointer</td></tr>
</table>

Fig 1: TCP segment format

□ Acknowledgment number

It is of 32 bits, if an ACK control bit is set, and this field indicates the value of the next sequence number of the segment to be received. This is always required to be sent once a connection is established. The transmission Control Protocol header also contains six status flags as outlined below [6].

□ URG (Urgent): This prioritizes specified traffic.

□ ACK (Acknowledgment): It acknowledges an SYN or receipt of data.

□ PSH (Push): This forces an immediate to send even if a window is not full.

□ RST (Reset): It forcefully terminates an improper connection.

□ SYN (Synchronize): It initiates a connection.

□ FIN (Finish): It gracefully terminates a connection when there is further data to send.

User Datagram Protocol (UDP)

User Datagram Protocol is a standard protocol with STD number 6 as described by RFC 768 –User Datagram Protocol. Its status is standard, and almost every TCP/IP implementation intended for small data unit's transfer or those which can afford to lose a little amount of data (such as multimedia streaming) will include User Datagram Program. It is basically an application interface to IP. It adds no reliability, flow-control or error recovery to IP. It simply serves as a multiplexer/multiplexer for sending and receiving datagrams, using ports to direct the datagrams [6], [30].

UDP datagram format

Each User Datagram Protocol datagram is sent within a single.

IP datagram. Although, the IP datagram might be fragmented during transmission, the receiving IP implementation will reassemble it before presenting it to the User Datagram Program layer. All IP implementations are required to accept datagrams of 576 bytes, which means that, allowing for maximum-size IP header of 60 bytes, a UDP datagram of 516 bytes is acceptable to all implementations. Many implementations will accept larger datagrams, but this is not guaranteed [5].

Source Port	Destination Port
Length	Checksum
Data	

Fig 2: UDP datagram format

From fig 2, above, components of the UDP Datagram are described below:

□ *Source Port:* It indicates the port of the sending process. It is the port to which replies are addressed.

□ *Destination Port:* describes the port of the destination process on the destination host.

□ *Length:* The length (in bytes) of this user datagram, including the header.

□ *Checksum*

An optional 16-bit one's complemented of the ones complement sum of a pseudo-IP header, the User Datagram Protocol header, and the User Datagram Protocol data.

Basic operation of User Datagram Protocol

User Datagram Protocol operation involves atmost two main process when is invoked by an application [27].

1) User Datagram Protocol encapsulates the data of users into datagrams.

2) Finally forwards these datagrams to the IP layer for the transmission. On the other side, these datagrams are afterwards forwarded to User Datagram Protocol from the IP layer. User Datagram Protocol subsequently removes the data from the datagram and forwards to the upper application layer. In User Datagram Protocol, a port is a number that specifies the application which is using the User Datagram Protocol service. It can be assumed as an address of the applications. The port number is also being used by the User Datagram Protocol client on the receivers end so that it can know that to which application the user data has to be forwarded [27].

Denial of Service flooding Attacks in Internet Protocol

Denial of service attacks is the deliberate attempt to prevent or inhibits the normal use or management of communication facilities. The attempt may have a specific target; for example: to suppress all messages directed to a particular destination. It in addition, can be of the form of denial of service by which there is a disruption of the entire network either by disabling the network or overloading it with messages aimed at degrading the performance.

How Denial of Service flooding Attacks works

There are various means by which the Internet enabled network can be compromised. The malicious could focus on compromising the network through exploiting weakness, attack distribution, attack traffic dynamics, attack enhancing techniques and others as classified in the studies. Expressing in a more technical term's denial of service attacks exploits. weaknesses in the Internet protocol, applications, operating systems, and protocol implementation in operating systems [2], [25], [51].In this paper, however, it focuses on the class of exploited weakness in Internet protocol implementation to make the attacked computer or network slow or stop working as intended.

Flooding denial of service attack is a technique of compromising the network under consideration where by attacker requests existing or non-existing content to overload the distribution infrastructure. It can be implemented by sending interest packets, which are not resolved at all or not resolved fast enough and thus lead to malicious CPU or memory consumption of a transmission resource [60].

Control Protocol Synchronizing flooding Attacks.

Theory on Transmission Control Protocol as described above in this section, among of other things discussed is the TCP segment with its contents. The attacker uses the loophole in the operations of different contents of the segment such as flags, e.g. Synchronizing (SYN), Finish (FIN), reset (RST), Urgent (URG), Acknowledgment (ACK) and Push (PSH) to compromise the service to the legitimate traffic. One of the features by which Transmission Control Protocol can be characterized is a Connection, which is a combination of sockets, sequence numbers, and window sizes. Every connection is uniquely specified by a pair of sockets identifying it's both sides. Whenever the two processes want to communicate, their TCPs has to first; establish a connection (initialize the status information on both sides). Once the communication is complete, the connection has to be terminated or closed. In other words, TCP connection between the two end computers is established by three-way handshake mechanism illustrated below [34];

Step 1. Client A sends a SYN packet to Client B
Step 2. Client B sends a SYN/ACK packet to Client A
Step 3. Client A sends an ACK packet to Client B

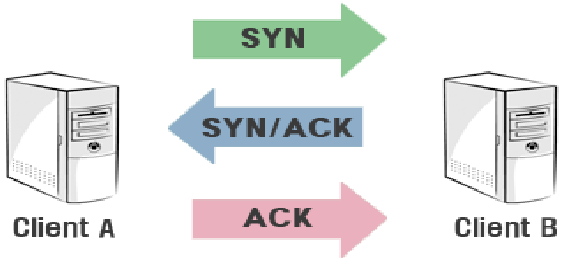


Fig 3: TCP three handshake

An SYN flood is a form of denial-of-service attack in which an attacker sends a succession of SYN requests to a target's system in an attempt to consume enough server resources to make the system unresponsive to legitimate traffic [60].

Transmission Control Protocol SYN flooding attack is an attack technique whereby the malicious conducts a denial-of-service (DoS) attack on a computer server by repeatedly sending SYN TCP segments to every port on the server using a fake IP address. The server responds to each such attempt with an SYN/ACK (a response segment whose SYN and ACK flag bits are set) segment from each open port and with a Reset (RST) segment from each closed port. Because the attacker intention is not to establish the connection; it never sends back the expected ACK segment. Moreover, as soon as a connection for a given port gets timed out, another SYN request arrives for the same port from the malicious [57].The manner in which the denial of service is performed in this kind of attack is also elaborated by [57] which contends that the connection is stored in the backlog queue on the server who counts a limited amount of time, then it removes the connection when timeout.

□ *User Datagram Protocol (UDP) flooding Attacks*

This is the kind of attack whereby the malicious continuously sends the UDP packet to a random port on the victims system. On receiving the UDP packet the victims system determines what application is waiting on the destination port, when it realizes that no application that is waiting in the port, it generates an ICMP packet to the destination unreachable to the spoofed source address. If the significant UDP packets are delivered to ports on the victim, the system will go down.

A. Network Performance Metrics

Performance metric is a quantitative measure of performance, specific to an application transported over an IETF-specific protocol [13].The IP Performance metric (IPPM) working group of the Internet Engineering Task Force defines Metrics, which are essential in measuring the Internet performance whose ultimate aim is to evaluate quality, performance and reliability of the Internet data delivery services [13],[39]. The group identifies a number of Metrics, but in this paper, three Metrics have been taken into account; throughput-related delay-related metrics and packet loss metric. The performance Metrics have been chosen on the ground that they are compatible with most of Post Processing trace analyzer tools associated with the ns-2. The metrics considered in this project is outlined below [45];

□ Throughput

This is the rate at which a network sends or receives data. It is evaluated in bits/second

$$\text{Throughput } Tp=Pa/Pf \qquad (1)$$

Where; Pa is the packet received and Pf are the amount of packets forwarded over the certain interval of time. Measurements of this network performance metric is a good indicator of denial of service for the higher-volume transactions, for example, transfer of large files, on the other hand, cannot capture the denial of service quality for short transactions [33].

□ Packet loss

It is the term which describes the phenomenon when the network traffic fails to reach the desired destination in a timely manner.

$$\text{Packet drop } (Pd) = Ps—Pa \qquad (2)$$

Where Ps is the amount of packets sent,
Pa is amount of Packet received.

The measurement done with respect to these metric captures impact of flooding denial of service attack, it cannot be applied to pulsing attacks.

□ *End to end delay:*

This refers to the time taken for a packet to be transmitted across the network from source to destination it defines all possible delays. Queuing delay, retransmission, latency and buffering during route's discovery are the factors which cause delay in the network. Lower delay means better performance [36].

$$D=Td—Ts \qquad (3)$$

Where: D= Experienced delay Td = Amount of delay experienced by the packets at the destination side side.
Ts=Amount of delay experienced by the packets at the sending side.

Techniques of Analyzing Network Performance Metrics

There are vast of techniques used in analyzing network entities [47]. Each technique had both merits and demerits, but their aim is to analyze the behavior of the network under the varied conditions. The techniques used to analyze network performances are; simulation, emulation, test-bed and real network analyzing techniques. The latter technique is rarely employed due to the high cost and availability issues.

Simulation Platforms

Simulation is the imitation of the operation of a real-world process or system over time. It requires first a model which represents the key characteristics, behavior and functions of the selected physical or abstract system or process [47].In this context of the project network topology along with parameters of interest is regarded as the model. Studies conducted using the simulation platform allows researchers to evaluate aspects of the network in a controlled manner. They can create different network topologies; it incorporates an operating systems behavior, and sends customized traffic patterns, change network scenarios and to collect data for analysis purposes. Currently, there exists a number of simulators for analysis of network performance. In the list frequently used simulators being ns2/ns3, OPNeT, OMNeT, GloMoSim and QualNeT [19, 35, 38, 42, 47, 49]. Some of these simulators are the open source, and others are commercially available. Therefore, the choice depends on the trade-off between the intended purpose and cost implications. Network Simulator version2 is most popular because it an open source and consists of prominent features suitable for modeling network performance[38].It's availability reduces the costs, so that attracts researchers in the network performance arena. OPNET and GloMoSim availability reduces the costs, and attracts researchers in the network performance arena. OPNET, GloMoSim and QualNeT are commercially available platforms, which offer excellent GUI and help facilities, but it is costly relative to ns2 and OMNeT [35].

Emulation platforms

In the context of network, emulation is the network virtualization which involves introducing a device, typically in a test environment, that alters packet flows in such a way to mimic the behavior of a real –world network. Emulation based research has a network scenario represented by a combination of one or more surrogate systems [47] allowing one to conduct experiments using some real components and combining them with simulation. Literatures on network emulators [11], [1], [44] figure out types, features and associated applications that allow one to select in accordance with purposes. Ignoring the in depth details of the simulators, the list includes: Hitbox, Ohio Network Emulator, dummy net NIST Net and trace–based mobile network emulator. Emulation technique retains most of the advantages of opting for simulation with an addition of other features that are implemented in Testbed environment.

Testbed Platforms

Testbed is the controlled experimentation platform that conforming to the IIC Reference Architecture where solutions can be employed and tested in an environment that resembles to the real-world conditions [26]. Contrary to simulation and emulation, testbed environment enables researchers to have full access on the various tools and equipment they intend to use in their experiments. There a range of testbed platforms in use today but the commonly used in analyzing network performance-related studies include; GENI [18], DETER [8], and ORBIT [43]. Experiments undertaken on the testbed platforms have a lot of benefits [11] but cost and accessibility issues have to be taken into account.

Quantifying Network Performance Metrics Existing Approach

In this subsection various flooding denial of service attacks on TCP and UDP; simulation, test bed and emulation based platforms are analyzed. The analysis is based on the quantitative description of the flood on network performance Metrics. The reviewed literature carried out and analyzed the network performance by generating traffic, collecting the statistics, visualizing graphically. Some quantified one or more network performance Metrics by post processing the trace file of ns2.
Fig. 4, shown below describes the conceptual frame work of the studies undertaken in relation to denial of service attacks. It uses terminal command line tools available in Unix or Linux environment or OTCL scripts, which serve as input or traffic generators to the network platform. The output of the platform is the trace file which shows events of the packets. This trace file can be further processed to visualize and quantify the network performance Metrics.
The surveyed literature for this study reveals there is an exhaustive use of tools such as Perl, AWK and Matlab as processors of the trace file. In paper [10] Trafil is in addition to Perl and AWK identified as a tool for automating network simulation and processing of tracing data files. Although the tool is based on ns2 trace files it can be extended to support a number of different other trace file formats. On the other hand, terminal command line tools such as wireshark, netstart, tcpdump, hping and others display the statistics of the simulate or emulated network.
Based on the working principles of ICMP and TCP protocol [21] demonstrated the effects of ICMP echo reply and TCP Synchronize (SYN) network attack on the network performance; the experimentation was done on open flow-enabled GENI test bed. Traffic generated, captured and monitored by using the command line tools such as hping3, wire shark and IPTraf. The conclusion drawn from this indicates that there was a denial of service. Concluding basing on observation of comparing statistics measurements or distributions may not be accurate, as it doesn't indicate what service was actually denied. This scenario merely shows how network traffic behaves differently under attack; it doesn't specify the application, quantify the denied service and how severely [34].

In a dump-bell, topology [29] analyzed the distributed denial of service attack on File Transfer Protocol services when the User Datagram Protocol and Transmission Control Protocol packet flooding attack launched using SEER GUI BETA6 on the DETER Testbed platform. In the study scripts for traffic generation and statistics, collection was developed. The signal behavior of both flat, pulse and ramp signals under attack and attack-free was visualized graphically.

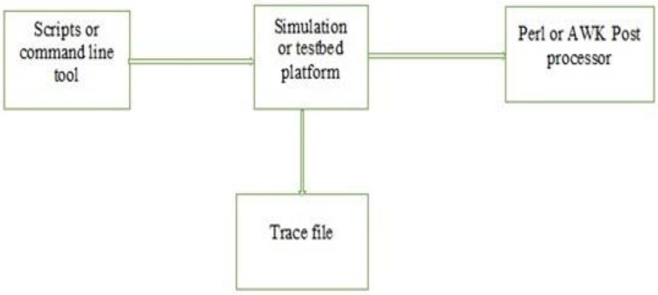

Scripts or command line tool
Simulation or testbed platform
Perl or AWK Post processor
Trace file

Fig 4: Existing approach

Selection of the application under which the experiment aimed at determining the denial of service when it is subjected to attacks is one of the approaches, which is acceptable. In this, however, it may serve the human perception as it doesn't quantify and indicate the level of severity.

Flooding denial of service attack on the bandwidth of wired network results into packet drops at the bottleneck resource [53]. The Xgraph used to depict the variation of throughput due to attack scenario indicates impact caused by an attack. This, however, in implementation it may result from unlike interpretations. Therefore, there is a need to improve the approach so that an exact value helps to avoid different interpretations.

Quantified values of TCP and UDP protocol's throughput was obtained through post processing ns2 trace files using PERL scripts [30].Under the same simulation parameters TCP throughput shows more reliability relative to UDP throughput. The obtained value can be easily interpreted for implementation's purposes, but in this scenario, it compares the average throughput of the protocol under free-attacks scenario only. Contribution of this paper is to go further in simulating the TCP and UDP protocols without/with attack scenarios and quantifies more than one network performance parameters using the automated post processing tools. Automated post processing tool is more reliable than other tools previously used in a sense that the quantified value of a metric can be moreover, validated within the platform. Take example the instantaneous throughput read from numerical data can be validated by simply simulating the same but an average throughput metric is enabled in the tool's graphical user interface.

The author of paper [7] predicted the impact of denial of service attacks by post processing the output information produced by different measurement tools in Matlab. The attacks taken into account in this study include; UDP flooding (large packets and small packets), slow post, flash-crowd and GET flooding. The difficulty in achieving a UDP flooding with large packets and GET wo database were reported due to the used environment since the network performance stabilizes after reaching the bottleneck limit.

Performance metrics of VoIP systems and VoWLAN [61] was obtained by post processing the simulated trace file of ns2 using awk scripts. They concluded that the throughput of both systems are the same as indicated in the gnuplot graph. The value read direct from the graph may lead to different interpretations.

Performance of TCP under varied flow control window size in [62] was evaluated on the ns2 platform which generated the trace file. Computations of quantified values of instantaneous throughput and delay were obtained by further processing the trace file on awk scripts. In this approach, the values obtained are really quantitative and showed the relationship between the number of packets sent and received to compute the throughput. The difficulty in this approach it requires frequently changing the parameters which prolong the evaluation process, also instantaneous throughput and delay, needs separate scripts, and different simulation scenarios.

II. QUANTIFYING NETWORK PERFORMANCE METRICS PROPOSED APROACH

The current study which is aimed at quantifying the severity of flooding denial of Service of TCP (SYN), and UDP are performed using the network simulator. Network Simulator version 2 has been selected to facilitate the simulation in quantifying network performance metrics of particular interest in this project is the throughput and delay. Network Simulator version2 is the most popular in education purposes as 70% activities in the network-related research and projects use it [42]. The software is an open source, discrete event simulator for computer networks. It uses OTCL code, which enables a user to define network topology, protocols, applications that users tend to simulate. In this paper, the network simulator software is installed to the personal computer which contain Ubuntu operating system compiled and executed according to the written terminal command language script to link nodes in dumbbell topology.

A. Network performance Metrics quantification Proposed approach

The existing approach in quantifying the network performance metrics provides useful information, which helps to come out with an alternative approach that addresses issues remains uncovered. Network Simulator version 2 is used because it is an open source and consists of prominent features suitable for modeling network performance [38]. Its availability reduces the costs, so that attracts researchers in the network performance arena. The approach involves developing OTCL scripts and execution of the scripts on ns2. The trace file obtained from the simulator is further post transformed by using an Automated Post Processing (APP) tool, where graphical visualization and quantified values of instantaneous throughput and instantaneous delay are obtained.

Network Performance Metrics Preprocessing

The main goal of this paper was to quantify the severity of flooding denial of service on the network performance metrics. This involves selection of metrics of interest as shown in fig. 5, thereafter object terminal Command Language script is developed, when code compiled and run in the ns2 software, the output is the trace file which records the traffics' events took place during simulation. The result of the simulations is an output trace file that can be used to do data processing (calculate delay, throughput, packet loss and jitter) [53], [45].

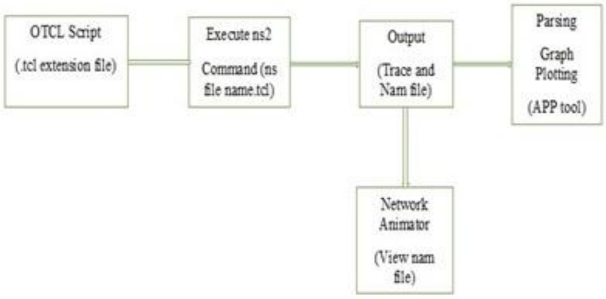


Fig 5: Proposed approach conceptual framework

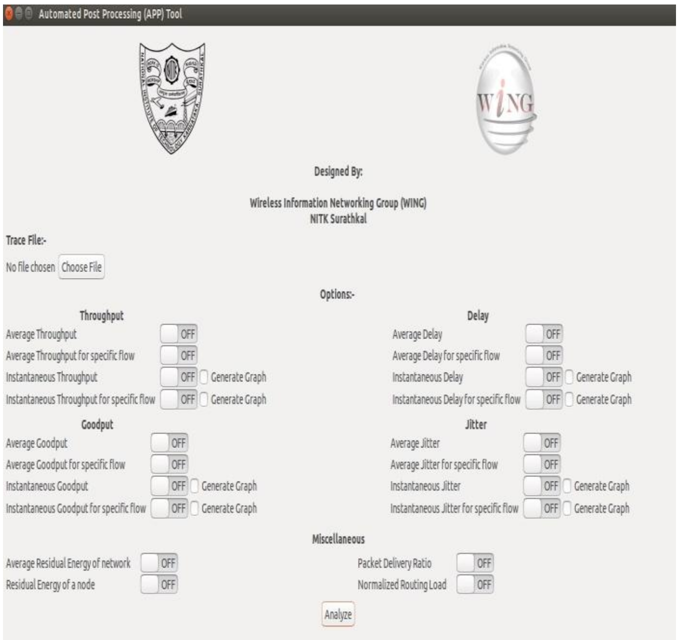
Network Performance Metrics Post Processing

Many network performances Metrics's evaluation simulation based platforms, such as ns2, OPNeT and GloMoSim displays statistics. The statistics show activities regarding the packets flows from source to destination. These statistically displayed values have to be quantified to obtain performance metrics like throughput, end to end delay, and delay variations.

In this paper, Automated Post Processing (APP) tools have been used in post processing the out. tr trace file produced during simulation of OTCL scripts. The metrics of interest is analyzed and visualized both graphically and quantitatively to quantify the throughput and packet delay of the network under free-attacks and during attack scenarios. The tool had an added advantage that it is possible to analyze up to twenty network performance metrics as shown in fig.6, below. Apart from this reason, no one to the best of our knowledge has undertaken network performance related study using this tool.

B. Simulation Setup

This section outlines the network topology which carries the picture of the network infrastructure along with the descriptions of simulation scenarios. In each scenario a detail of OTCL script is done to guide preprocessing and post processing of network performance metrics for severity determinations.


Automated Post Processing (APP) Tool

Designed By:

Wireless Information Networking Group (WING)
NITK Surathkal

Trace File:-

No file chosen Choose File

Options:-

Throughput
Average Throughput OFF
Average Throughput for specific flow OFF
Instantaneous Throughput OFF □ Generate Graph
Instantaneous Throughput for specific flow OFF □ Generate Graph

Goodput
Average Goodput OFF
Average Goodput for specific flow OFF
Instantaneous Goodput OFF □ Generate Graph
Instantaneous Goodput for specific flow OFF □ Generate Graph

Delay
Average Delay OFF
Average Delay for specific flow OFF
Instantaneous Delay OFF □ Generate Graph
Instantaneous Delay for specific flow OFF □ Generate Graph

Jitter
Average Jitter OFF
Average Jitter for specific flow OFF
Instantaneous Jitter OFF □ Generate Graph
Instantaneous Jitter for specific flow OFF □ Generate Graph

Miscellaneous
Average Residual Energy of network OFF
Residual Energy of a node OFF
Packet Delivery Ratio OFF
Normalized Routing Load OFF

Fig 6: Automated Post Processing tool graphical user interface

Experimental Topology

In this paper the Dumbbell Topology has been selected and arranged in such a way that it contains the primary components of flooding denial of service attack scenario that jointly determine all the elements of an attack scenario. These elements are two legitimate traffic client node, one attack traffic node, one router and the server node (target).Dumbbell topology with single bottle neck is used by most of the network researchers [60] for the evaluation of the network characteristics. The topology shown in figure 3.2 above can be easily scaled to unlike sizes. Distinctive resources have their unique communication addresses, so here assumed that all clients had attached processor core as resources, therefore, and treated similarly except that a traffic generator can be attached to resources. Clients, router/target and link are three basic elements in the topology, assume that the resources had infinite buffer size but finite in clients. It means that the packet being dropped or lost cannot occur in resources but only take place in clients. In figure 3.2 shown above: 0, 2 represent the client's node of TCP and UDP protocol correspondingly where the legitimate traffics are generated. On the other hand, 1 is the attacker node for launching attack traffic. While 3 and 4 are router and target relatively, the setup is virtual machine with Ubuntu operating system Throughout in this paper node denoted by 0,1,2,3 and 4 are represented by n_0, n_1, n_2,n_3 and n_4 respectively.

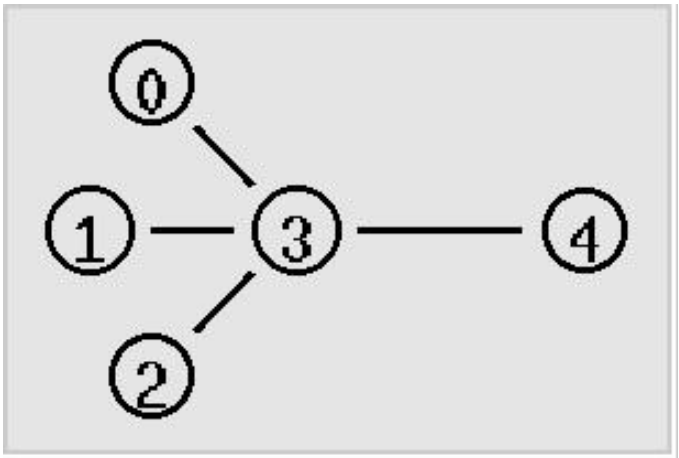


Fig 7: Experimental topology

Simulation Parameters

This simulation in the ns2 platform, parameters shown in table 1 and 2 are assigned to the topology components, also they are used in OTCL scripts. These parameters are used in developed scripts defines the intended behavior of the packets in the topology. In the topology 0,1,2,3 and 4 represent the nodes as the source of legitimate and attack traffic, the bottleneck router and the destination node respectively. For clarity in OTCL scripts these are designated as n_0, n_1, n_2, n_3 and n_4.

Table 1: Links, bandwidth, delay and queue

Link	Bandwidth	Delay	Queue	Queue limit
n_0 to n_3	2Mb	10ms	Drop Tail	
n_1 to n_3	2Mb	10ms	Drop Tail	
n_2 to n_3	2Mb	10ms	Drop Tail	
$n_3$to n_4	1.7Mb	20ms	Drop Tail/RED	10

Table 2: Applications and agent

Node	Agent	Application
n_0	TCP	FTP
n_1	TCP	FTP
n_2	UDP	CBR
n_3	N/A(Sink)	N/A
n_4	N/A(Destination)	N/A

Flooding denial of service OTCL scripts

Fig.7, shows the experimental topology; therefore, we developed OTCL scripts based on ns2 under attack free and attack scenario of both TCP and UDP. These scripts are omitted in this paper due to space limit.

III. SIMULATION RESULTS AND DISCUSSION

This section presents the simulation results such as traffic flows in the topology during attack free and attack scenario. Quantified Instantaneous throughput, delay and their associated graphs of both TCP and UDP with/without attack scenarios were also shown along with the interpretation.

A. *TCP attack-free scenario*

In this scenario traffic, flow is visualized, the quantified values of throughput and delay are analyzed when only legitimate traffic flows from n0 via n3 to n4 as shown in fig. 8, below.

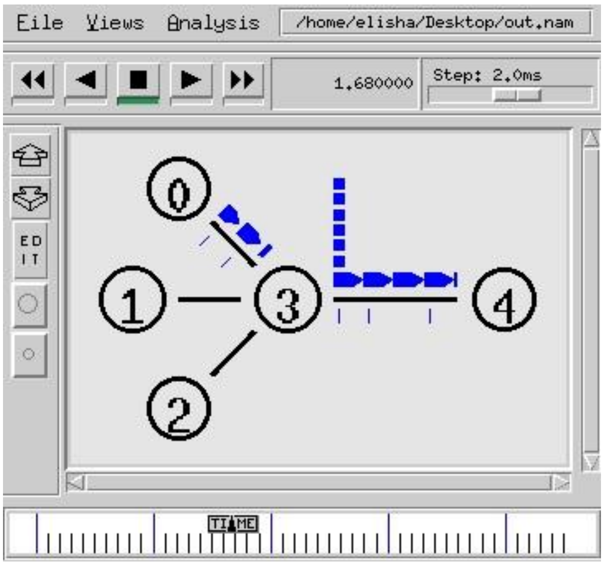


Fig 8: TCP Legitimate traffic flows

The object terminal command language scripts (OTCL) developed in the previous section (III) when executed, the traffic flows within the topology demonstrated in fig.6, it indicates that link n_3-n_4 have the capacity to accommodate the traffic flowing from the source n_0 to the destination n_4. Therefore no packets drops due to link capacity.

- *TCP Instantaneous throughput*

The instantaneous throughput is obtained in the Automated Post Processing tool by setting the tick interval of 0.1. This illustrates the behavior of traffic as they propagate from n0 via n3 to the destination n4. During simulation, the legitimate traffic starts to flow at the interval 0.2 as shown in fig.9, below, however, the allocated value in the scripts is 0.1 this difference between the allotted value and starting time accommodates the time for initialization.

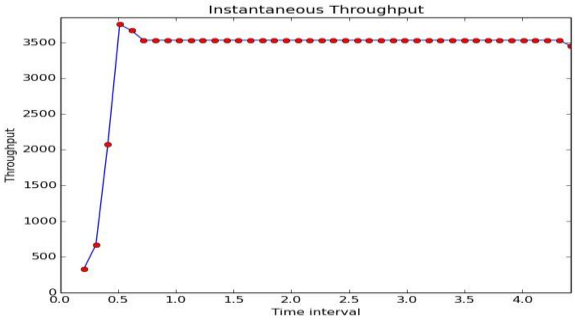


Fig 9: Instantaneous throughput

In fig.9, it can be observed that there are variations of the instantaneous throughput from 330.259 at 0.2 to 3665.48 at 0.6. The rest of an interval remains almost steadfast at approximately 3531. The firm value established in the graph justifies the absence of an attack traffic.

TCP Instantaneous delay

Fig.10, shown below depicts the distributions of the traffic queued up or requested at the particular interval of time. It can be observed that traffic lining up and dequeuing starts at the interval of 0.1 up to 4.4, which are the last interval of simulation instructed in the TCP legitimate traffic OTCL scripts.

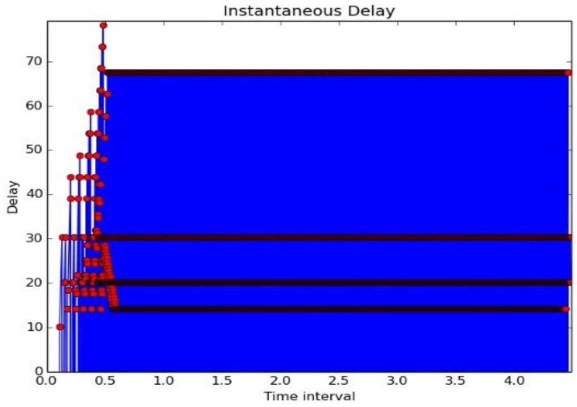


Fig 10: TCP Instantaneous delay

The quantified values of the delay were observed to alternate, that is traffic queuing or dequeuing. However, the average delay was 48.6134; this is the only largest positive value observed to occur mostly once at different interval location. It is also a value of interest, which needs attention so that the delay can be dealt accordingly by reducing or eliminating the agents who may give rise to delay through the network under design or roots delay through the network being optimized.

B. TCP attack scenario

The simulation done at this scenario includes both legitimate and attack traffics allowed to propagate through the network topology but they had different start and stop time.

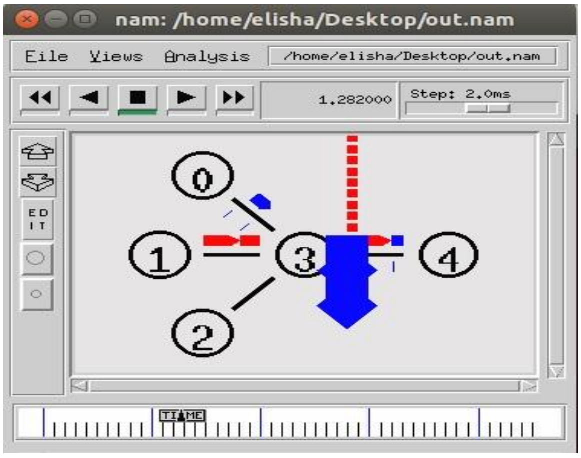

nam: /home/elisha/Desktop/out.nam
File
Views
Analysis
/home/elisha/Desktop/out.nam
1.282000
Step: 2.0ms
EDIT
0
1
2
3
4
TIME

Fig 11: TCP Legitimate and attack traffic flow

The OTCL scripts developed in section (III), which are not included due to space limit; legitimate and attack traffic starts and stop time as (0.1, 4.4 and 1.0, 3.5) respectively. During this scenario, it is observed from fig. 11, above that the attack traffic from node (n_1) and the legitimate traffic from node (n_0) compete on the limited resource link connecting node (n_3) to node (n_4) to send the packets. It results into dropping some of the packets based on the random early detection (RED) queuing algorithm.

TCP Instantaneous throughput

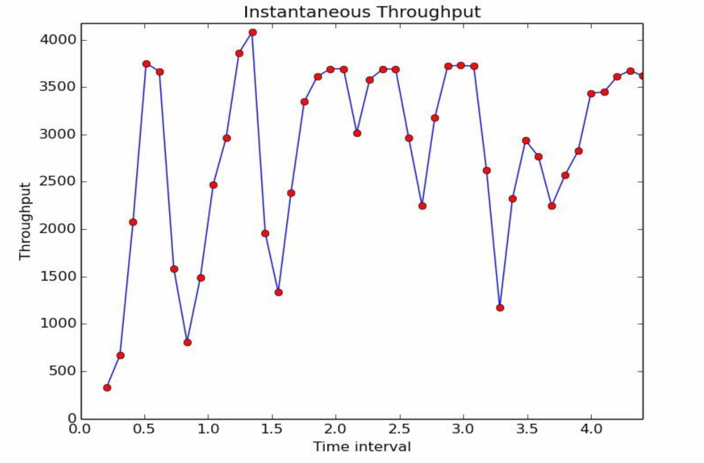


Fig 12: TCP Instantaneous throughput

The variations of the throughput during attack scenario seen in fig.12, above starts at an interval of 1.0 towards the higher interval 4.4.There are some points where the legitimate traffic throughput is higher or small compared to that of the attack traffic, however, average wise, the instantaneous throughput of legitimate traffic is higher than that obtained during attack. scenario which are 3318.28 and 2786.85 respectively.

TCP Instantaneous delay

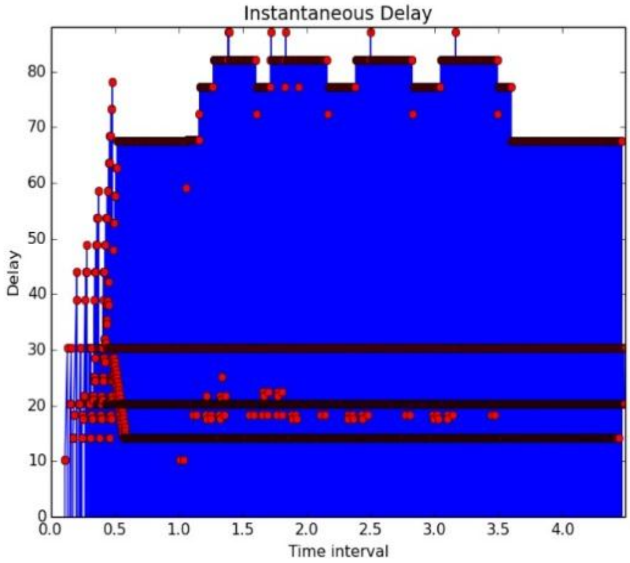


Fig 13: TCP traffic Instantaneous delay

The TCP instantaneous delay is shown graphically in fig.13, above. In this scenario the quantified instantaneous delay obtained over the small interval and validated over the large interval is found to be 52.3308. The value is obtained by averaging the delays occurs only once on the consecutive or the same interval of time. Compared to the delay obtained in fig.10, that is 48.6134, the delay experienced in fig.13, is higher, which means that during an attack scenario, traffics are more delayed.

C. *UDP free attack scenario*

In fig.14, shown below nodes denoted by 0,1,2,3 and 4 are represented as n_0, n_1, n_2,n_3 and n_4 in the OTCL scripts developed in section (III).)

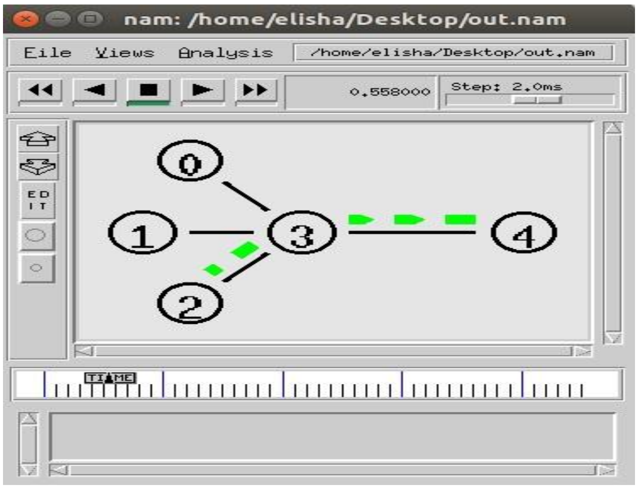

nam: /home/elisha/Desktop/out.nam
File
Views
Analysis
/home/elisha/Desktop/out.nam
0.558000
Step: 2.0ms
EDIT
0
1
3
4
2
TIME

Fig 14: UDP Legitimate traffic flow

The CBR legitimate traffic is generated at node (n_2) flows along n_2-n_3 towards n_4 as illustrated in fig.14, above. The link n_3-n_4 capacity accommodates the traffic generated by node n_2 whose traffic distributions in the network is shown in fig.15, below. This is essentially the instantaneous throughput recorded interpreted by the automated post-processing tool from the trace file during attack.

Instantaneous throughput

Fig.15, is the instantaneous throughput distribution, this indicates that at 0.2 the instantaneous throughput is 1568.63 while the rest of intervals is uniformly distributed at 2000, this implies an absence of attack traffic.

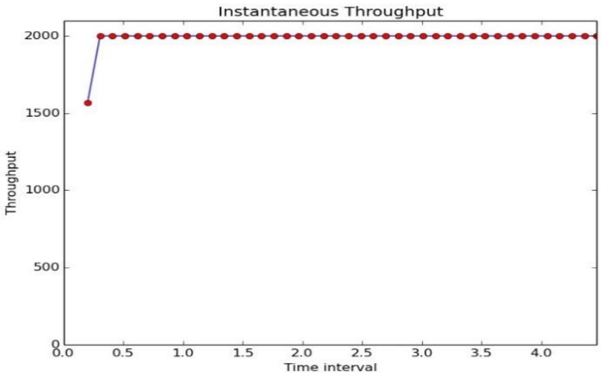


Fig 15: UDP Instantaneous throughput

Instantaneous delay

The instantaneous delay of the CBR legitimate traffic is obtained by quantifying the trace file in an automated post processing tools set at a tick interval of 0.1. This tic interval is as well the point at which traffic generated at node (n_2) as shown in fig.16. The quantified instantaneous delays at an average of 38.706 are obtained from the output of an automated post processing tools whose delay graph is shown below in fig.15. The value could be as well decided based on the value observed to occur most at a different intervals on the output graphical user interface of automated post processing (APP) tool.

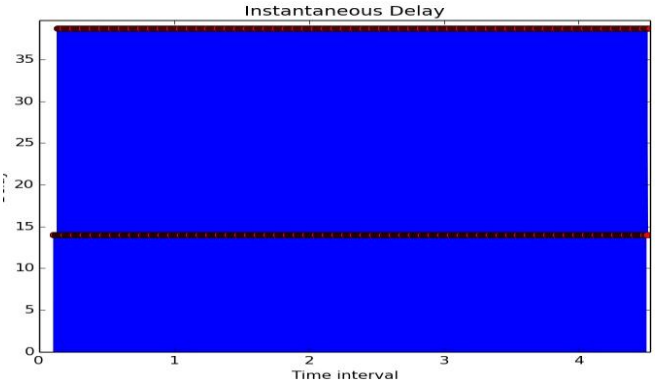


Fig 16 Instantaneous delay

D. *UDP Attack scenario*

Fig.17, shows both UDP attack traffic generated at node n_1 and UDP legitimate traffic generated at node n_2, both sends packets through link n_3-n_4 towards the destination node n_4, due to traffic congestion some packets drops. The consequence of packet drops is the reduction of the instantaneous throughput, this is illustrated in fig.18.

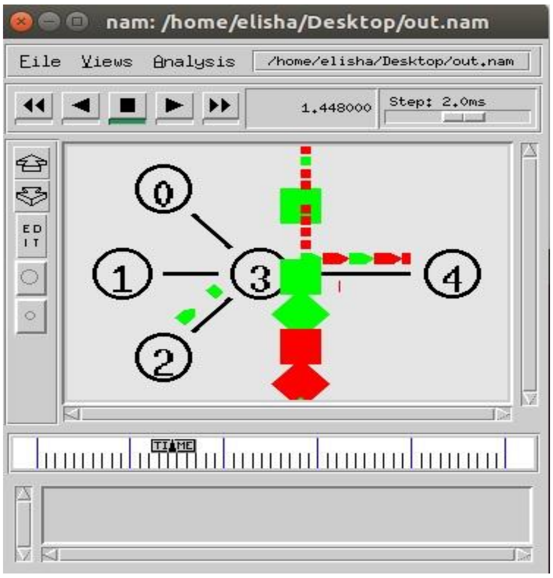

nam: /home/elisha/Desktop/out.nam
File
Views
Analysis
/home/elisha/Desktop/out.nam
1.448000
Step: 2.0ms
EDIT
0
1
3
4
2
TIME

Fig 17: UDP Legitimate and attack traffic flow

Instantaneous throughput

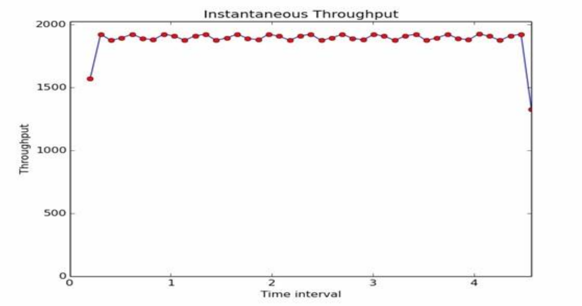


Fig 18: UDP Instantaneous throughput

From the fig.18, shown above it can be seen that the instantaneous throughput is 1568.63 at the 0.2 interval which is the same as the value of an attack scenario shown in fig. 14, at the same interval. The User Datagram Protocol legitimate and attack traffic starts at 0.1 and 1.0 respectively, at the point where the attack traffic starts, the quantified instantaneous throughput of true traffic is shown to be degraded from 2000 to 1910.46. This decrease in instantaneous throughput was due to packet drops and collisions. It is further observed that the instantaneous throughput during the attack scenario is less compared to the attack-free scenario from the starts of an attack to the end of the simulation time.

Instantaneous delay

The legitimate and attack traffic flows in fig.17, are queued when the congestion in the network is detected whose traffic distributions are indicated in fig.19. The quantified value of the instantaneous delay obtained from automated post processing tool is recorded as 94.5223 this value is also validated by computing the average delay within the processing tool. The instantaneous delay obtained during an attack scenario is bigger than that obtained on the attack-free scenario.

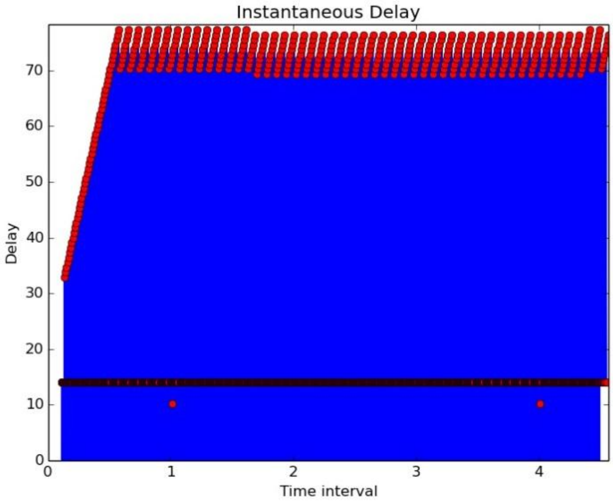


Fig 19: UDP Instantaneous delay

Severity of denial of service attack

Table 3: TCP severity

Metric	Free-attack	Attack	Difference	percentage
Instantaneous throughput	3318.28	2786.85	531.43	16
Instantaneous delay	48.6134	52.3308	3.7174	7

Table 3, above describes the severity intensity caused by TCP flooding denial of service attacks, it is found that the instantaneous throughput intensity simulated by an attack starting at 1.0 and ends at 3.5 is 16%.On the other hand using the same parameters that are starting and stops attack time, the severity is found to be 7%.In theory, throughput is related to the net packets received at the destination node. Therefore, in the scenario shown above if the attack commencement time could be launched early say at 0.5 more packets could drop whose consequence a decrease is in the throughput during that scenario.

The decrease in throughput would result from an increase in severity percentage accordingly a denial of service on a specific application could be experienced. This phenomenon of varying the attack time would result also cause increase in delay during an attack scenario, which would eventually result from a denial of service on that particular application.

Table 4: UDP severity

Metric	Free-attack	Attack	Difference	percentage
Throughput	2000	1910	90	4.5
Delay	38.706	94.5223	55.8163	59

Table 4, above shows the severity of UDP flooding denial of service. The phenomenon described in table 4, applies also in UDP that is varying the attack time would result in a higher severity experienced by the particular application.

IV. SUMMARY, AND CONCLUSION

In this concluding section, a summary of the findings is described, of interest how the intended objectives have been achieved along with the challenges encountered. The direction of the future work is also suggested.

A. Summary

The main goal of this paper was to quantify the severity of flooding based denial of service attack on internet enabled network performance metrics. To accomplish this main goal it became necessary to reach some specific objectives which were: to examine transport control protocol, user datagram protocol, to quantify the network performance metric and to determine the severity caused by the flooding denial of service attack on the network throughput and delay.

Literature survey on transport layer protocol, particularly transport control protocol and user datagram protocol have been done to gain understanding on the embedded mechanisms. This is vital because it is a loophole where malicious take advantage to compromise the network by flooding it with fake traffic that renders denial of service to the legitimate users.

Techniques of analyzing network performance metrics have been conducted to reveal the current practices on the platforms frequently used. It serves also as the basis of identifying the gap which was seen to rely in a particular post processing tool that is AWK, command line and PERL. Analysis of previous work related to the present paper gave foundations of what has not been done, but needs attention. The new approach in evaluation of the network performance metrics was developed, and its performance was demonstrated on network simulator version2 on attack free and attack scenarios.

Lastly the network performance throughput and delay quantified values have been obtained by post processing the trace file of ns2 on an automated post processing tools.

B. Conclusion

Findings of this paper indicate that the quantified instantaneous throughput obtained during an attack-free scenario is larger relative to that obtained during an attack scenario. This implies that the packet's drop observed during simulation causes a reduction of the throughput due to signal-to-noise ratio and bandwidth limitations. The consequence of throughput reduction is the reduction of network performance as well as service degradation. On the other hand, however, the Instantaneous delay observed during an attack scenario is larger compared to the attack-free scenario instantaneous delay. The difference in delay is a result of traffics being queued for a long-period due to attack traffic. If an alternative mechanism to decrease the queuing time is used, the delay can be decreased significantly.

The computed intensity of severity caused by the attack on throughput and delay concluded that, the severity is highly depended on throughput and delay. Therefore, denial of service is likely to occur at some points during attack on file transfer protocol and constant bit rate traffic.

The most important limitations encountered in this paper lies in inability to handle a vast quantified values of delay. Equally important much data resulted from choice of tick interval being small, affects the fidelity of the graph and vice versa. Impressive results can be drawn from this by addressing the limitations encountered. In particular, one should investigate by employing other queuing algorithms other than drop tail and random early detection and manipulation of tick interval.

Our paper underlined the importance of quantifying and visualizing Instantaneous throughput and delay. It is easier to spot the moment at which the performance is expected to be low or high. Therefore, an appropriate measure can be anticipated. Thereby, we believe that our approach is an innovative one in the process of network performance evaluation.

To further our research we plan to investigate the proper way of handling vast quantified values of instantaneous delay.

V. REFERRENCES

[1] Abliz, M. (n.d.). Internet Denial of Service Attacks and Defense Mechanisms.

[2] Aijaz, M., & Parveen, S. (2016). Analysis of Dos and DDos Attacks. *International Journal of Emerging Research in Management &Technology*, *5*(5).

[3] Alosaimi, W., & Al-Begain, K. (2013). An Enhanced Economical Denial of Sustainability Mitigation System for the Cloud. *2013 Seventh International Conference on Next Generation Mobile Apps, Services and Technologies*.

[4] Anchit, B., & Harvinder, S. (2016). Investigation of UDP Bot Flooding Attack. *Indian Journal of Science and Technology*, *9*(21).

[5] Arora, D., Singh, P., & Singh, V. (2014). Impact Analysis of Denial of Service (DoS) due to Packet Flooding. *Int. Journal of Engineering Research and Applications*, *4*(3).

[6] Balchunas, A. (n.d.). *TCP and UDP vl.21*.

[7] Bannwart, C. (2012). *Predicting the Impact of Denial of Service Attacks* (Master's thesis, Swiss Federal Institute of Technology, Zurich, German).

[8] Benzel, T., Braden, R., Kim, D., & Neuman, C. (n.d.). Design, Deployment, and Use of the DETER Testbed.

[9] Bogdanoski, M., Shuminoski, T., & Risteski, A. (2013). Analysis of the SYN Flood DoS Attack. *International Journal of Computer Network and Information Security*,*5*(8).

[10] Bouras, C., Charalambides, S., Drakoulelis, M., Kioumourtzis, G., & Stamos, K. (2012). *A tool for automating network simulation and processing tracing data files. ELSEVIER*. Retrieved from journal homepage: www.elsevier.com/ locate/simpat

[11] Braun, T. (n.d.). *Network Emulation* [PDF Document].

[12] Carson, M., & Santay, D. (n.d.). NIST Net – *A Linux-based Network Emulation Tool.*

[13] Clark, A. (2011). *Internet Engineering Task Force (IETF) Request for Comments:*
6390 (ISSN: 2070-1721). Cisco Systems, Inc.

[14] Douligeris, C., & Mitrokotsa, A. (2004). *DDoS attacks and defense mechanisms: classification and state-of-the-art. Computer Networks, 44(5).*

[15] EGLI, P. R. (2015). *Introduction to TCP;The Internets standard Transport Protocol*. In *TCP: Transmission Control Protocol,* Retrieved from indigo.com

[16] Elleithy, K. M., Blagovic, D., Cheng, W. K., & Sideleau, P. (2005). Denial of Service Attack Techniques: Analysis, Implementation and Comparison. *Computer Science and Information Technology Faculty Publication* [Sacred Heart University].

[17] Fall, K. W., & Stevens, W. R. (2011). *TCP/IP illustrated, Volume 1, The protocols*. Boston, MA: Addison-Wesley.

[18] Gall, A. (2006). *GENI –A new breed of testbed for network Innovation*. The SWISS Education and Research work.

[19] Garg, N. (2015). Network Simulators: A Case Study. *International Journal of Advanced Research in Computer Science and Software Engineering*, *5*(1). Retrieved from www.ijarcsse.com

[20] Ghazali. (2011). Flooding Distributed Denial of Service Attacks-A Review. *Journal of Computer Science*, *7*(8).

[21] Gorla, M. C., Kamaraju, V. M., & Centikaya, E. K. (2015). Network attack experimentation on open flow-enabled GENI testbed. *Electrical and computer engineering department, Mounsuri university of science and technology*.

[22] Gupta, S. K., & Saket, R. K. (2011). Performance Metric Comparison of AODV and DSDV Routing Protocol in MANETs using NS-2. *IJRRAS*, *7*(3)

[23] Gupta, S., & Nikita. (2016). Analyzing the Congestionand Flow Control in UDP Protocol Using NS2. *International Journal of Computer Science and Mobile Computing*, *5*(5).

[24] Houle, K. J., Weaver, G. M., Thomas, R., & Long, N. (2001). *Trends in Denial of Service Attack Technology* (V1.0). CERT® Coordination Center

[25] Hussain, A., Heidemann, J., & Papadopoulos, C. (2003). A framework for classifying denial of service attacks. *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '03*.

[26] *IIC Quarterly Report*. (2015). Industrial Internet CONSORTIUM.

[27] International Business Machines Corporation. (1992). *IBM international technical support centers: TCP/IP tutorial and technical overview*. Research Triangle Park, NC: IBM Corp.

[28] Kapri, H. (2011). *NETWORK Traffic Data Analysis* (Master's thesis, Louisiana State University and Agricultural and Mechanical College).

[29] Kaur, D., Sachdeva, M., & Kumar, K. (2012). Study of DDoS attacks using DETER Testbed. *International Journal of Computing and Business Research (IJCBR)*, *3*(2).

[30] Kaushik, S., Poonam, & Tomar, A. (2014). A Comparative Analysis of Transport Layer Protocols. *International Journal of Information & Computation Technology*, *4*(14).

[31] Li, M., Li, J., & Zhao, W. (2008). Simulation Study of Flood Attacking of DDOS. *2008 International Conference on Internet Computing in Science and Engineering*.

[32] Marsic, I. (2010). Book: Computer Networking - textbook by Ivan Marsic. Retrieved from http://www.ece.rutgers.edu/~marsic/books/CN/

[33] Mirkovic, J., Hussain, A., Wilson, B., Fahmy, S., Reiher, P., Thomas, R. … Schwab, S. (2007). Towards user-centric metrics for denial-of-service measurement. Proceedings of the 2007 workshop on Experimental computer science

[34] Mirkovic, J., Hussain, A., Fahmy, S., Reiher, P., & Thomas, R. (2009). Accurately Measuring Denial of Service in Simulation and Testbed Experiments. *IEEE Transactions on Dependable and Secure Computing*, *6*(2).

[35] Nychis, G. (n.d.). *Network Simulators Emulators and Testbeds* [Power Point slides].

[36] Ochang, P. A., & Irving, P. (2016). Performance Analysis of Wireless Network Throughput and Security Protocol Integration. *International Journal of Future Generation Communication and Networking*, *9*(1).

[37] Owezarski, P. (n.d.). On the impact of DoS attacks on internet traffic characteristics and QoS. Proceedings. 14th International Conference on Computer Communications and Networks, 2005.

[38] Pan, J. (n.d.). A Survey of Network Simulation Tools: Current Status and Future Developments.

[39] Papaleo, D., & Salsano, S. (1999). *The Linux Traffic Generator* (003-004-1999). University of Rome "La Sapienza".

[40] Ponemon Institute LLC. (2016). *Cost of Denial of Services Attacks: Data Center Performance Benchmark Series*.

[41] Q., & Liu, P. (n.d.). Denial of Service Attacks. *Department of Computer Science* [San Marcos].

[42] Raj M.R, C., Chacko, N. M., Major, J., & Shibin, D. (2013). A Comprehensive Overview on Different Network Simulators. *International Journal of Engineering and Technology (IJET)*, *5*(1).

[43] Raychaudhuri, D., Seskar, I., Ott, M., Ganu, S., Ramachandran, K., Kremo, H., … Siracusa, R. (n.d.). Overview of the ORBIT Radio Grid Testbed for Evaluation of Next-Generation Wireless Network Protocols.

[44] Rizzo, L. (n.d.). Dummynet : a simple approach to the evaluation of network protocols.

[45] Salleh, A. U., Ishak, Z., Din, N. M., & Jamaludin, M. Z. (2006). Trace Analyzer for NS-2. *2006 4th Student Conference on Research and Development*.

[46] Schuba, C., Krsul, I., Kuhn, M., Spafford, E., Sundaram, A., & Zamboni, D. (n.d.). Analysis of a denial of service attack on TCP. *Proceedings. 1997 IEEE Symposium on Security and Privacy*.

[47] Shaneel, N. (2014). *Improving Network Performance: An Evaluation of TCP/UDP on Networks* (Doctoral dissertation, UNITEC Institute of Technology, Auckland, New Zealand).

[48] Shea, R., & Liu, J. (2013). Performance of Virtual Machines Under Networked Denial of Service Attacks: Experiments and Analysis. *IEEE Systems Journal*, *7*(2).

[49] Siraj, S., Gupta, A. K., & Badgujar, R. (2012). Network Simulation Tools Survey. *International Journal of Advanced Research in Computer and Communication Engineering*, *1*(4).

[50] Symantec. (2016). *Internet Security Threat Report*.

[51] Stevens, W. R. (n.d.). *TCP/IP Illustrated,The Protocols* [Volume 1].

[52] The TCP/IP Guide - The TCP/IP Guide. (n.d.). Retrieved fromhttp://www.tcpipguide.com/free/index
.htm

[53] Tomar, K., & Tyagi, S. S. (2014). Quantifying the Impact of Flood Attack on Transport Layer Protocol. *International Journal on Computational Sciences & Applications (IJCSA)*, *4*(6).

[54] Torrents, D. T. (2010). *Open Source Traffic Analyzer* (Master's thesis, KTH Information and Technology).

[55] User Datagram Protocol - Wikipedia. (n.d.).
http://en.wikipedia.org/wiki/User_Datagram_Protocol

[56] Won, S. J. (n.d.). TCP SYN Flood - Denial of Service.

[57] Xiaoming, L., Sejdini, V., & Chowdhury, H. (n.d.). Denial of Service (DoS) attack with UDP Flood.

[58] Zada, S. A. (2010). Ad Hoc Networks: Performance Evaluation of Proactive, Reactive and Hybrid Routing Protocols in NS2 (Master's thesis, University West).

[59] https://en.wikipedia.org/wiki/Internet_Flooding_Attack accessed on 7th September 2016]

[60] Man, L., Lu, M., & Zhong, D. (2013). *Evaluation and Comparison of Wired VoIP Systems to VoWLAN*
[PDF]. Retrieved from http://www.sfu.ca

[61] Bhargava, N., Bhargava, R., Kumawat, A., & Kumar,
(2012). Performance of TCP-Throughput on NS2 by Using Different Simulation parameters.
International Journal of Advanced Computer Research, *2*(6), 323-327.