

# Having an EAGLE-eye on GENI

Shifali\*, Monika Sachdeva\*\* and Sunny Behal\*\*\*

## ABSTRACT

Experimentation with new network architecture and protocols is one of the idiopathic stimuli for piling future internet test bed such as GENI (Global Environment for Network Innovations) test bed. GENI proffer a virtual laboratory for networking and distributed system scrutiny and breeding. The motive of this test bed is to nurture the experimentation with succeeding protocols, services and network architecture instead to frame succeeding Internet. As a cause of reconnaissance of network, GENI foster innovations in network security, science, applications and services. In this paper, we have an eagle eye on GENI by scrutinizing its Resource Reservation Tool - Jack and Instrumentation and Measurement Tool-GENI Desktop. As part of the work, GENI has been used to synthetically generate different kinds of network traffic datasets like legitimate traffic, high rate DDoS attack and Flash Events(FE) traffic.

**Keywords:** Validation Techniques, GENI, GENI-Desktop, Workflow of GENI.

## 1. INTRODUCTION

In the Recent time, the collegial, public & self-sufficing facility tabbed to be Internet which is handy for hundreds of millions of people worldwide has spread its wings globally. However the intrinsic frangibility of Internet provides opportunities to attackers. Distributed Denial of Service (DDOS) impersonates slang threat to the availability of Internet. DDOS attacks encompass the stiff security problems in today's Internet. With meager or no advance warning, a DDOS attack can easily cripple the resources of its mark within a precise cycle of time. There are various network research validation techniques for analyzing these attacks like Mathematical Models, Simulation, Emulation and Real time experiment. By the whole of these, GENI is tremendously used in network related research based on real time experiments. In GENI, the experiment is not stipulate by system, but is licensed by the experimenter.

- MOTIVATION:

The idea for using GENI test bed is motivated from [1], since number of researcher has used the publicly available datasets for validating their proposed approaches. However most of these available datasets are obsolete. So there is need to generate realistic datasets. Based on this, we have synthetically generated our own datasets in GENI test bed.

- CONTRIBUTIONS:

The major contribution of this paper is:

- (1) Distinct research validation techniques are given along with the brief introduction of GENI test bed.

---

\* Department of Computer and Science Engineering SBS State Technical Campus, Ferozepur, Punjab, India, *Email: shifalichawla91@yahoo.in*

\*\* Department of Computer and Science Engineering SBS State Technical Campus, Ferozepur, Punjab, India, *Email: monika.sal@rediffmail.com*

\*\*\* Department of Computer and Science Engineering SBS State Technical Campus, Ferozepur, Punjab, India, *Email: sunnybehal@rediffmail.com*

- (2) The designing of network based experiments using GENI testbed is explored comprehensively.
- (3) Hands on experience on utilizing benchmark tools on GENI testbed for synthetically generating the varieties of network traffic along with their usage are provided.

The rest of the paper is standardized in such a manner that: Section-2 describes the distinct validation techniques available. In Section-3, GENI is characterized and in Section-4 GENI Experimenter workflow is explained. Section-5 represents the benchmark tools used in our experiment and Section-6 represent the realistic traffic that we generate in GENI and the last section conclude the feature of realistic traffic generation in GENI.

## 2. RESEARCH VALIDATION TECHNIQUES

The recipe recommends by researcher in the area of network security needs to be legitimized via the valid set of validation techniques. Four proposals are used for validation [1].

- **Mathematical models:** As per the creation Mathematical model are hypothetical. In these standards, the given applications, system, conditions and platforms are cast figuratively and then mathematically validated.
- **Simulation:** On a single computer, a controllable and repeatable frame for network based analysis is the byproduct of simulation. Prototype and evaluation based environment makes the Programmers pliable to do experiments in order to discard the bad alternatives timely without seeking for full implementation. Examples include NS2 [2], NS3 [3], OMNET++ [4], Qualnet3 [5], OPNET [6], CORE [7] etc.
- **Emulation:** Unification of simulation and real system is called emulation. Emulation scuttles in real setup; however simulation runs in virtual simulated setup. Scalability factor in Emulation build the topology vast enough.
- **Real Systems:** For real operating systems, realistic network conditions, platforms and applications, there is a need for the real system. If any network that operate with speed 1Kbps and we can cast a system that can give an output by more than 1Kbps, then that system will be a real time system with respect to network. Examples include GENI, Planet Lab etc.

## 3. INTRODUCTION TO GENI

GENI Global Environment for Network Innovations proffers a virtual laboratory for networking and distributed system scrutiny and breeding. The motive of this test bed is to nurture the experimentation with succeeding protocols, services and network architecture instead to frame succeeding Internet. By traversing networks at scale, GENI foster innovations in network security, science, applications and services. GENI concede experimenters to:

- 1) To gobble up the compute resources from the different locations of the United States.
- 2) Hook up the compute resources using Layer 2 networks in topologies on the nose to their trial and error.
- 3) Lodge the compute resources with tailor made software or even tailored operating systems.
- 4) Command the network switches to handle the traffic.
- 5) Lodge the protocol software in their compute resources and dart their own Layer 3 and above protocols on them.

### 3.1. Why GENI?

- **Colossal framework for trial and error:** - GENI can equip vastly greater resources than is customarily orchestrate in any of the laboratory. GENI accord ingress to hundreds of universally distributed resources

including network as well as compute resources , the network resources includes - switches, links and WiMax base stations however the compute resources include - virtual machines as well as bare-machines.

- **Non-IP connectivity across resources:** - GENI authorize the lodging of Layer 2 connections betwixt the compute resources and dart Layer 3 and above protocols connecting these resources.
- **Deep programmability:** - GENI authorize to analyze with novel network layer protocol or novel IP - routing algorithm by granting programming of the switches in the core of the network together with end host of the network.
- **Reproducibility:** - GENI yield the ability to replay your trial by granting exclusive access to certain GENI-resources.
- **Instrumentation and measurement tools:** - GENI proffer 2 instrumentation and measurement systems for instrument the experiments. These systems proffer tools for visualizing as well as analyzing measurement data.

### 3.2. Key GENI Concepts

- *Project:* A project comprises both people and their experiments and systemizes the research in GENI. The responsible individual leads the project: the Project Lead. The project could be composed of as many experimenters as possible to be as members and also an experimenter could be a part of as many projects as possible. Eventually the Project Lead is answerable for every single action of project members in project. To set up a project, GENI experimenters need to have privileges from project lead. GENI permit only the faculty members and high-ranking members of the organization to be a project leads, however students cannot be a project leads.
- *Slice:* Multiple experimenters can run multiple experiments at the same time that make the GENI a shared test bed. This is feasible as of the course of slice. A GENI slice is [8]:
  - 1) A unit of isolation for experiments. A GENI experiment's life is within slice, modification to experiment can only be by the members of slice.
  - 2) A can of resources to be dressed by an experiment. GENI Experimenters can dart experiment by adding resources to slice.
  - 3) A unit of access control. The experimenter can elect who will have access to slice. As a matter of course the Project Lead is part of every slice of the project.
- *GENI Aggregates:* GENI aggregate such as a GENI Rack at a university grant resources to experimenter. Discrete aggregates earmark discrete resources. Certain aggregates grant compute resources, some else issue network resources. The Figure 1 shows a GENI slice with reservation of the resources from multiple aggregates.
- *The GENI AM API and GENI RSpecs:* Appealing of the resources is through a standard API tabbed to be GENI AM API (GENI Aggregate Manager API). The AM API permit the experimenters to:
  - 1) Screenshot the resources available at an aggregate.
  - 2) Appeal for the specific resources to be assigned to their slices from the aggregate.
  - 3) Help to perceive the status of the resources that are allocated to slices from the aggregate.
  - 4) X-out the resources from their slices.

The Figure 2 shows resources added from two distinct aggregates to slice using the Allocate call.

- *Resource Reservation using RSpecs and the AM API*



Figure 1: GENI Slice with resources reserved from multiple aggregates

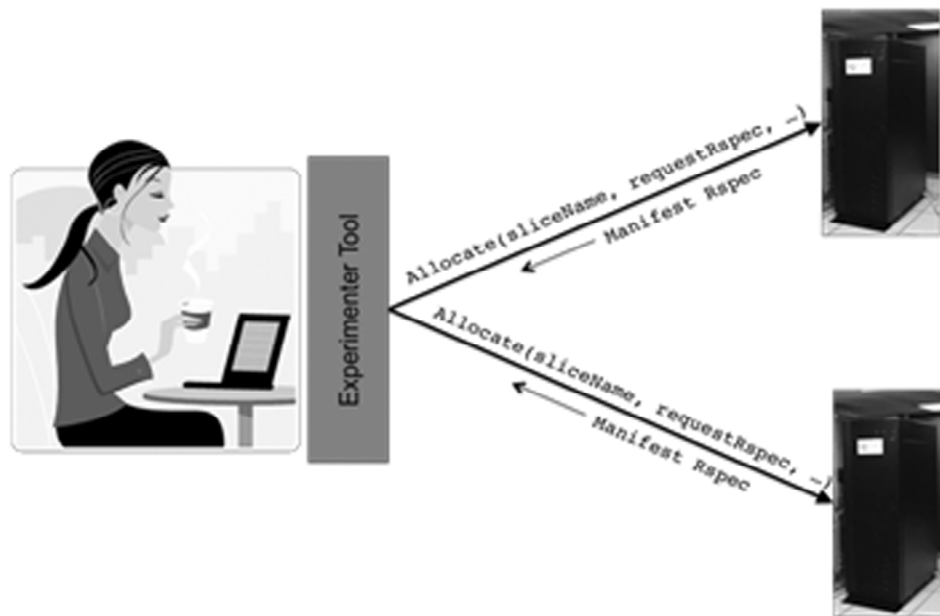


Figure 2: Resource added from two different aggregates by the Experimenter

The AM API utilize resource specification document, tabbed to be GENI RSpecs, to specify resources. RSpecs are just XML documents in a prescribed format. Three types of RSpecs are:

- 1) Request RSpec: Experimenters request to the aggregates via the request RSpec that catalogue the resources as they wish.
- 2) Advertisement RSpec: The aggregates reports the experimenter a manifest RSpec that catalogue the resources they grab.
- 3) Manifest RSpec: The RSpec returned by an aggregate when an experimenter screened the resources accessible at the aggregate.

The Figure 3 shows an experimenter reserving resources with three different types of RSpec.

### 3.3. GENI Experimenter Tools

GENI experimenter tools cover a range of topics:

- 1) **Resource reservation tools:** Creating slices, adding or removing resources from slices, and to frame experimental topologies are piled through Resource Reservation tool. For e.g. Jack, Jfed, Omni [9], GENI-lib [10].

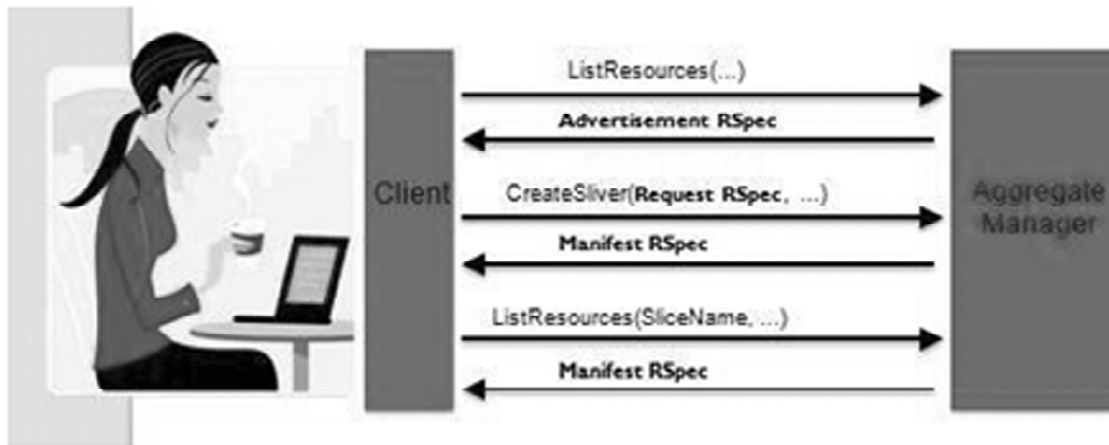


Figure 3: Experimenter reserving resources with three types of rspec

- 2) **Instrumentation and measurement tools:** Measuring the statistics of nodes and to orchestrate the experiment procedure, the Instrumentation and measurement tools are the base. For e.g. GENI Desktop, Lab-Wiki [11].
- 3) **Services running on GENI:** These are services running on GENI but are accessible to GENI experimenters. For e.g. GENI Experimenter Engine (GEE) [12], IDMS: Intelligent Data Movement Service [13], Virtual Topology Service (VTS) [14].
- 4) **GENI-federated and outside tools:** These are resources or test beds available to experimenters with GENI credentials. For e.g. Cloud Lab.

#### 4. THE GENI EXPERIMENTER WORKFLOW

The following is the workflow for a typical GENI experiment.[8]

##### 4.1. Step-by-step Instructions to work with Jack and GENI-Desktop

###### 1) Step 1: Get Ready:

- Step to GENI Experimenter Portal, hold use GENI button & then choose your organization. Choose GENI Project Office if your account is via GENI Identity Provider.

###### 2) Step 2: Launch your experiment:

- Beat create slice button in the portal.
- Named it something like demo.
- After loading, beat add Resources button, Jack tool is integrated in portal which grant to frame topologies.
- In Jack beat the black VM box and yank into the canvas.
- Iterate this step in many instances as you wish.
- Beat along one of the VM-box and yank toward other VM and then discharge, a line connecting two VM will show.
- Iterate the above mentioned steps. Two VM boxes will be appeared on the canvas.
- Press along one of the VM boxes, and then drag near the other VM. Release, once reached the other VM. A line as well as a box will be appeared showing the link betwixt the two VMs.
- Name of VM can be modified using the Name field that appears in the left.

- Choose the aggregate and then beat reserve button.
- Stay during your resources being reserved. The nodes appearing green signify that they are ready.

**Tip:** To use GENI-Desktop, press add Global node before clicking on reserve resources.

### 3) *Step 3: Authorize GENI-Desktop:*

- What is the GENI-Desktop?
  - The GENI-Desktop is a tool that grants numerous means to “visualize” a slice, & exert operations to resources within a slice. GENI-Desktop brace services tabbed as Modules that ease the operations as ssh, file-copy, running a command, rebooting nodes, etc. It also proffers graphs and table data for variegated statistics of the node. However this is through the use of GENI-Desktop monitoring tools that interplay with the experiment’s resources over the control plane via a Global Node. By its nature GENI-Desktop adds software to a slice that will scan & rate network traffic and other statistics. The GENI-Desktop is also integrated with JACKS.
  - Login to GENI-Desktop
- 1) Beat the GENI-Desktop button on GENI portal. This will move to new tab, beat authorize GENI-Desktop. We will authorize through our GENI account.
  - 2) Now after authorizing, you can either create a new slice or can use a slice which was created in GENI portal.
  - 3) Once resources are ready ,the screen then transitions to GENI-Desktop page for that slice with the status window displaying a status of Initializing followed by Instrumentizing [15]
  - 4) Once the slice has been initialized & instrumentized, you should also be able to use different modules of GENI-Desktop.
    - For loading the software onto the nodes, File Upload module will be used. Two shapes of File upload module are available- direct and cloud. In direct mode file will be uploaded directly from system to GENI’s nodes howbeit in cloud mode there is a two-step process, First uploading from user’s local machine to GENI-Desktop cloud server and then to any nodes of slice. Load your software onto nodes and run commands on nodes
    - For running Commands on Nodes, beat the command button, it will pop a window, select your nodes and then type command you wish to run on that node.
    - For Ssh’ing into nodes in GENI-Desktop, beat the Ssh button, select your nodes & then beat open Browser ssh button then accept security certificate, it will ssh into that node.
    - For viewing Traffic, two modules could be used, Passive Graphs Lite proffer basic view of the graphs but no control over pausing the live view or zoom in/zoom out. Howbeit for an enhanced view with control over pausing the live view, zoom in/zoom out, viewing data points and setting alerts Passive graph is best.
    - For downloading the Files from GENI, beat the File Download module it will pop a window, fill the location of file which you want to download. It will download the file from GENI nodes to your own system.
    - For restart your nodes, beat the reboot module.

## 5. BENCHMARK TOOLS USED

As according to [1], a number of researchers have used the publicly available dataset for validating their proposed methods. However most of the available real datasets are obsolete. In this paper we have

synthetically generated our own datasets in GENI test bed [8]. The benchmark attack tools and traffic generators [16], [23] that are used to synthetically generate the 3 scenarios: normal traffic, DDoS traffic and FE traffic are as follows:

- **Httpperf:** httpperf is a tool to measure web server performance. It speaks the HTTP protocol both in its HTTP/1.0 and HTTP/1.1 flavors and offers a variety of workload generators. While running, it keeps track of a number of performance metrics. The most basic operation of httpperf is to generate a fixed number of HTTP GET requests and to measure how many replies (responses) came back from the server and at what rate the responses arrived [17]. An example of using httpperf is shown below:

```
httpperf -server server_name --port 80 --uri /index.html \ --rate 150 --num-conn 27000 --num-call 1 --timeout 5
```

This command causes httpperf to use the web server on the host with IP name `server_name`, running at port 80. The web page being retrieved is “/index.html” and, in this simple test, the same page is retrieved repeatedly. The rate at which requests are issued is 150 per second. The test involves initiating a total of 27,000 TCP connections and on each connection one HTTP call is performed (a call consists of sending a request and receiving a reply). The timeout option selects the number of seconds that the client is willing to wait to hear back from the server. If this timeout expires, the tool considers the corresponding call to have failed. Note that with a total of 27,000 connections and a rate of 150 per second, the total test duration will be on the order of 180 seconds, independent of what load the server can actually sustain [18].

- **D-ITG:** D-ITG (Distributed Internet Traffic Generator) is a platform capable to produce IPv4 and IPv6 traffic by accurately replicating the workload of current Internet applications. At the same time D-ITG is also a network measurement tool able to measure the most common performance metrics (e.g. throughput, delay, jitter, packet loss) at packet level. The core features of D-ITG are provided by ITGSend and ITGRecv. ITGSend is the component responsible for generating traffic toward ITGRecv. Exploiting a multithreaded design, ITGSend can send multiple parallel traffic flows toward multiple ITGRecv instances, and ITGRecv can receive multiple parallel traffic flows from multiple ITGSend instances. A signaling channel is created between each couple of ITGSend and ITGRecv components to control the generation of all the traffic flows between them [19]. An example of using ITG-Send/Recv is shown below:

```
/ITGSend -T UDP -a 127.0.0.1 -c 100 -C 10 -t 15000 \ -l sender.log -x receiver.log
```

This way ITGSend will generate one UDP flow with constant payload size (100 bytes) and constant packet rate (10 pps) for 15 seconds (15000 ms) and two packet-level log files will be generated both on sender (-l option) and receiver (-x option) side.

- **BoNeSi:** BoNeSi, the DDoS Botnet Simulator is a Tool to simulate Botnet Traffic in a test bed environment on the wire. It is designed to study the effect of DDoS attacks. BoNeSi generates ICMP, UDP and TCP (HTTP) flooding attacks from a defined botnet size (different IP addresses). BoNeSi is highly configurable and rates, data volume, source IP addresses, URLs and other parameters can be configured. There are plenty of other tools out there to spoof IP addresses with UDP and ICMP, but for TCP spoofing, there is no solution. BoNeSi is the first tool to simulate HTTP-GET floods from large-scale bot networks. BoNeSi also tries to avoid to generate packets with easy identifiable patterns (which can be filtered out easily) [20]. An example of using BoNeSi is shown below:

```
bonesi -p udp -s 1200 -r 2 -i 50k-bots -u /site.html -d eth1 192.168.2.1:2405
```

This command will request `http://192.168.2.1/site.html` with 2 udp packets per second along with a payload of 1200bytes to host 192.168.2.1 on port 2405 via eth1.

- Golden Eye: Golden Eye is a HTTP/S Layer 7 Denial-of-Service Testing Tool [21]. It uses Keep Alive (and Connection: keep-alive) paired with Cache-Control options to persist socket connection busting through caching (when possible) until it consumes all available sockets on the HTTP/S server. This tool is a dos tool that is meant to put heavy load on HTTP servers in order to bring them to their knees by exhausting the resource pool. This tool is meant for research purposes only and any malicious usage of this tool is prohibited. An example of using Golden Eye is as follow:

```
sudo ./goldeneye.py http://www.goldeneyetestsite.com/
```

The above command uses -w = 10 workers, -s = 10 simultaneous connections, -m = random, a mix of GET and POST, A Perfect DoS.

- Curl-loader: curl-loader [22] (also known as “omes-nik” and “davlka”) is an open-source tool written in C-language, simulating application load and application behavior of thousands and tens of thousands HTTP/HTTPS and FTP/FTPS clients, each with its own source IP-address. In contrast to other tools curl-loader is using real C-written client protocol stacks, namely, HTTP and FTP stacks of libcurl and TLS/SSL of openssl, and simulates user behavior with support for login and authentication flavors. An example of a simple configuration file to be used for traffic generation client side against web server:

---

```
##### GENERAL SECTION #####
```

```
BATCH_NAME= 10K-clients
CLIENTS_NUM_MAX=10000
CLIENTS_NUM_START=100
CLIENTS_RAMPUP_INC=50
INTERFACE =eth0
NETMASK=255.255.0.0
IP_ADDR_MIN= 192.168.1.1
IP_ADDR_MAX= 192.168.53.255
CYCLES_NUM= -1
URLS_NUM= 1
```

```
##### URLs SECTION #####
```

```
URL=http://localhost/index.html
URL_SHORT_NAME="local-index"
REQUEST_TYPE=GET
TIMER_URL_COMPLETION = 0
TIMER_AFTER_URL_SLEEP = 0
```

---

In this file, the name of the batch is “10K-clients” with the maximum number of clients to be 10000; load will be started with initial 100 clients and each seconds will be added 50 more clients. Network interface eth0 will be used for the load with unique IP-address for each client taken from 192.168.1.1 up to 192.168.53.255 with netmask 255.255.0.0. Number of cycles is to be performed by each client in not limited (CYCLES\_NUM= -1) and load will proceed till user presses Cntl-C to stop it. Each client fetches a single url (URLS\_NUM= 1) http://localhost/index.html.



The url was given a short name “local-index”, and this name will appear at the Load Status GUI. The url will be fetched using HTTP GET request method and the download will not be limited by time ( $TIMER\_URL\_COMPLETION = 0$ ). If we would like to limit this operation by time, we could place here some number in milli-seconds, e.g.  $TIMER\_URL\_COMPLETION = 3000$ ; the timer will be monitored, enforced and reported, if missed, at Load Status GUI and other statistics. By default all missed url-completion timers are considered as errors and added to statistics as T-err. Each client after accomplishing a url, including possible protocol redirections, waits/sleeps for 0 ms ( $TIMER\_AFTER\_URL\_SLEEP = 0$ , which means, that actually client does not wait/sleep) and goes to the next URL, if any, to fetch.

## 6. REALISTIC TRAFFIC GENERATION IN GENI

In our experiment, we have used the above mentioned tools to generate the 3 scenarios such as: Normal traffic, Attack traffic, Flash traffic. The Figure 4, 5, 6 shows the probability distributions for normal traffic, DDoS attack traffic and Flash traffic respectively.

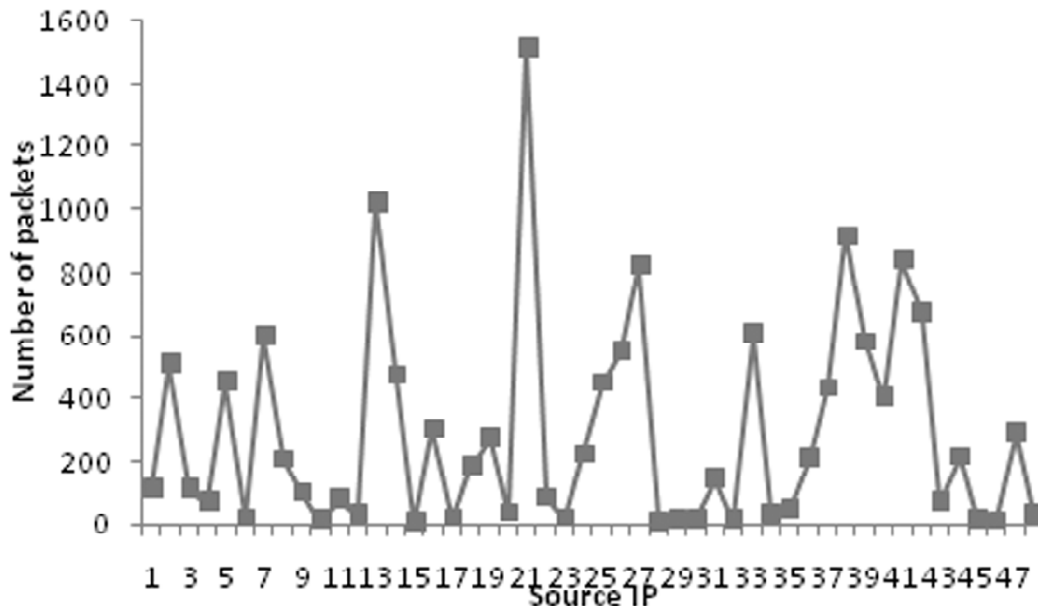


Figure 4: Probability distributions of Normal Traffic

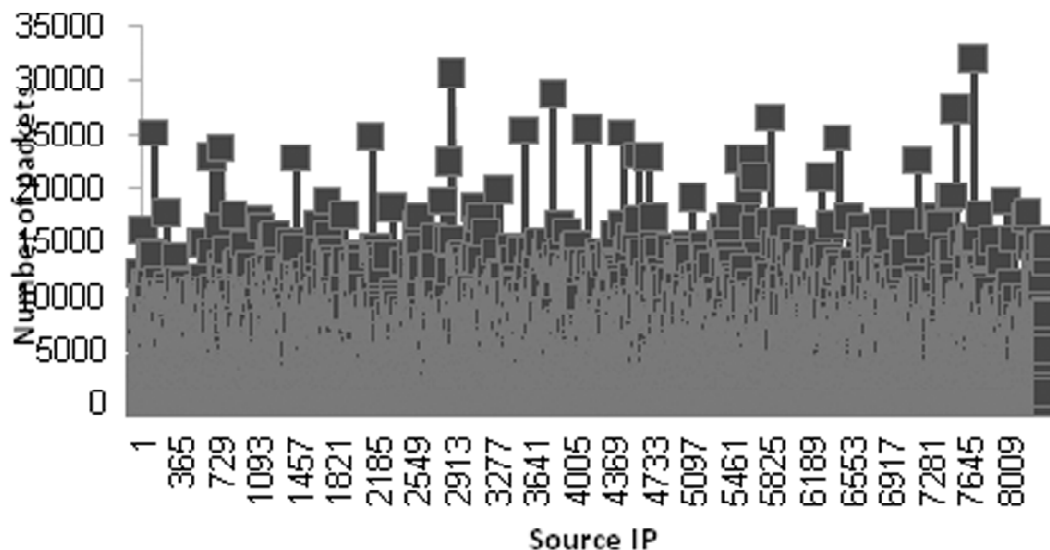


Figure 5: Probability distributions of DDoS traffic

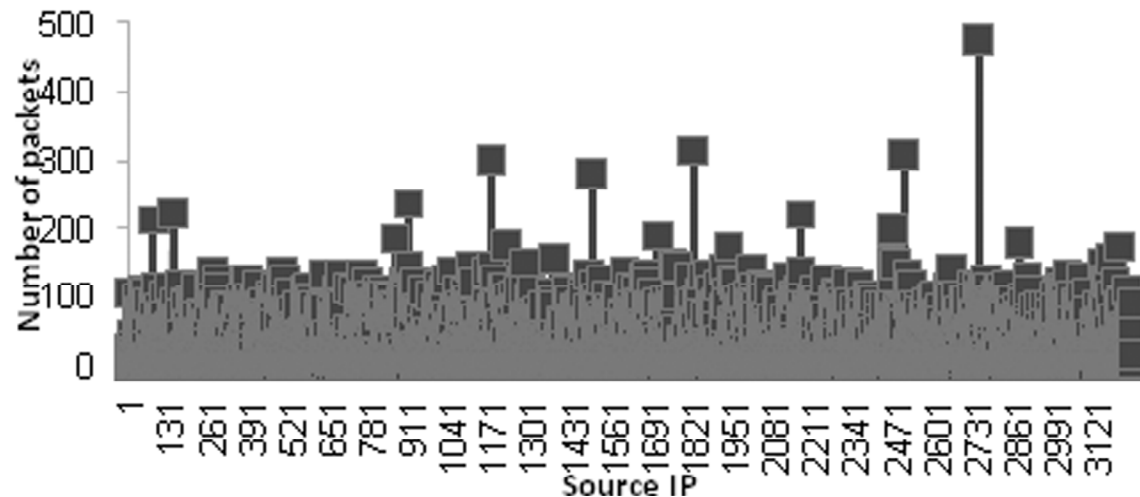


Figure 6: Probability distributions of Flash traffic

## 7. CONCLUSION

Hypothetical nature of mathematical model makes them impractical for the experiments based on networking. The techniques of simulation can approximate certain hardware and software functions but can't execute real applications. The emulation deals with the real applications and the hardware, on the other hand the types of hardware, number of nodes; difficulty in management, configuration, and reproducibility of experiments restricts its performance. So for the validation of network based research, real systems are ideal; they too are restricted in their functionality as per their complex creation. However, GENI, a real system based networking test bed, has emerged in the recent time, is discussed in this paper. GENI assay to integrate all the extant favored test beds such as Planet Lab [16], Emulab [17] etc. as single platform for the execution of real experiments at scale, GENI proffer leading edge for resource to engulf various detection & defense mechanism. So it is concluded that today's research is shifted towards real experiments such as in GENI. It will guide the researcher society to build the detection and discrimination mechanism with more efficacy. As a part of the future work we will use these datasets generated in GENI as well as the real publicly available datasets to detect high rate attack and further will discriminate Flash traffic from DDoS traffic.

## REFERENCES

- [1] S. Behal and K. Kumar, "Trends in validation of DDoS Research," *Procedia Computer Science*, vol. 85, pp. 7–15, 2016.
- [2] "The NS2 (network simulator 2)," [Online]. Available: <http://www.isi.edu/nsnam/ns/>.
- [3] "The NS3 (network simulator 3)," [Online]. Available: <https://www.nsnam.org/>.
- [4] "OMNET++ (The Network simulator)," [Online]. Available: <https://omnetpp.org/>.
- [5] "Qualnet (The Network simulator)," [Online]. Available: <http://web.scalable-networks.com/content/qualnet/>.
- [6] R.Chertov, S.Fahmy, and N. B.Shroff, "Emulation versus simulation: A case study of tcp-targeted denial of service attacks," in *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006. 2nd International Conference on. IEEE, 2006*, pp.10.
- [7] J. Ahrenholz, C. Danilov, T. R. Henderson, and J. H. Kim, "Core: A real-time network emulator," in *Military Communications Conference, 2008. MILCOM 2008. IEEE. IEEE, 2008*, pp. 1–7.
- [8] "GENI (Global Environment for Network Innovations)," [Online]. Available: <http://www.geni.net>.
- [9] "OMNI (The resource reservation tool)," [Online]. Available: <http://groups.geni.net/geni/wiki/HowToUseOmni>
- [10] "Geni-lib(The resource reservation tool)," [Online]. Available: <http://groups.geni.net/geni/wiki/HowToGenerateRSpecUsingGENILIB>.
- [11] "Labwiki(The Instrumentation and measurement tool)," Available: <http://groups.geni.net/geni/wiki/GEC18Agenda/LabWikiAndOEDL/Introduction>.

- 
- [12] “Geni Experimenter Engine(The Services running on Geni),” [Online]. Available: <http://groups.geni.net/geni/wiki/GeniExperimentEngine>.
  - [13] “Intelligent Data Movement Service(The services running on Geni),” [Online]. Available: <http://groups.geni.net/geni/wiki/sol4/IDMS>.
  - [14] “Virtual topology service (The services running on Geni),” [Online]. Available: <http://geni-lib.readthedocs.org/en/latest/tutorials/simplevts.html>.
  - [15] “GENI-Desktop Instrumentizing (Instrumentizing),” [Online]. Available: <http://groups.geni.net/geni/wiki/GENIExperimenter/Tutorials/GENIDesktop/>.
  - [16] H. Kaur, S. Behal, and K. Kumar, “Characterization and comparison of distributed denial of service attack tools,” in Green Computing and Internet of things (ICGCIoT), 2015 International Conference on. IEEE, 2015, pp. 1139–1145.
  - [17] “Httpperf tool to measure web server performance” [online] Available <http://linux.die.net/man/1/httpperf>.
  - [18] D. Mosberger and T. Jin, “httpperf a tool for measuring web server performance” ACM SIGMETRICS Performance Evaluation Review, vol. 26,no.3 , 1998.
  - [19] “Distributed Intrnet Traffic Generator (DITG)” [online] Available <http://traffic.comics.unina.it/software/ITG/manual/>.
  - [20] “BONESI” [online] Available <https://github.com/Markus-Go/bonesi>.
  - [21] “Golden Eye Tool” [Online] Available <http://sattia.blogspot.in/2014/12/dos-website-with-goldeneye-layer-7-dos.html>.
  - [22] “Curl-Loader” [Online] Available <http://curl-loader.sourceforge.net/>.
  - [23] S. Behal and K. Kumar ,”Characterization and Comparison of DDoS attack tools and Traffic Generators- A Review, “*International Journal of Network Security*, vol.19, No.3, pp.383-393, 2016.