

SUBMITTED BY: Pranav Patel - 2014 UCP1296 <https://in.linkedin.com/in/pranav-patel-956>  
Deepesh Mathuria - 2014UCP1263 <https://in.linkedin.com/in/deepesh-mathuria-352963a3>

## INTRODUCTION -

Packet generator works to simulate a traffic which resembles a real time human interaction driven traffic and generates a plot of the overall packet transfer - packet length vs departure time of packet and generates a trace file for the same.

The packet length generated is an exponential distribution. The packet passes through different protocol layers - first it pings - ICMP, then WINS comes into action to provide server name and service.

After this happens TCP tries to inject packets into our link (traffic comes from a variety of users : in our case we've generated 20 or configurable no. of different users who essentially share the same network ID but have different host IDs with the intention that people of a certain organization like these 20 or so are trying to inject a simulated but real traffic resembling traffic).

For each packet that TCP tries to inject into the link the packet has attributes TTL and hop, to avoid moving infinitely moving we've kept a fixed no. of hops i.e. 5, and if packet takes more than TTL time to move through then it is dumped. And if it gets through it is allowed to perform DNS query to translate the domain name to IP address.

For generating packets we've used **Poisson distribution** function :

$$P(X = x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

where  $P(X=x)$  is the probability mass function,  $\lambda$  = mean no of occurrences,  $x$  = occurrences.

**Scope:** There are further distributions available which can be used for modelling the packets, **heavy-tailed**, **pareto distribution** etc.

**Inspiration** : We tried to reverse engineer the process that **wireshark** and **tcpdump** use for packet sniffing to generate legit packets.

**Configurations:** You can change the no. users by changing the global variable `user_limit`:

## REQUIREMENTS

This module requires the following:

- \* Python 2.7 or higher
- \* matplotlib and numpy

## INSTALLATION

(Follow the following instructions on linux, FOR windows if you don't have python and pip on your path variable list then add them and then follow the below instructions)

- Unzip the archive PACKET GENERATOR v\_0.0.1.
- Check for python on your system.
- Check for matplotlib and numpy using terminal in python terminal by writing

```
import matplotlib
```

```
import numpy
```

If there is no error log then run the package from the PACKET GENERATOR v\_0.0.1 extracted directory by opening the terminal and issuing the command:

```
python sim.py
```

And packet traces are saved in **trace.txt**.

If numpy and matplotlib are not installed then install them using the following commands:

```
pip install matplotlib
```

```
pip install numpy
```

## METHODS DESCRIPTION -

`generate_packet()` : Initiates the process of packet generation

`WINS()`: provides NetBIOS and service name

`TCP()`: forwards the packet for DNS query.

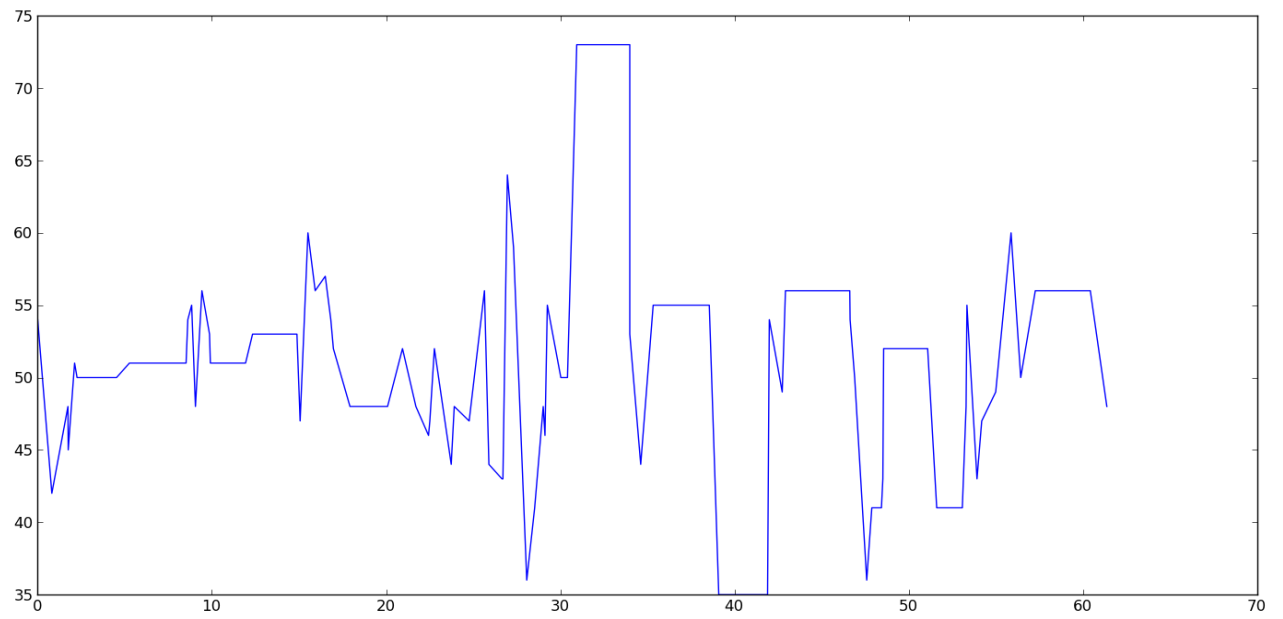
`Int_TCP()`: checks if TCP packet can get through

`ping()`: pings (ICMP) to check if the packet can be forwarded and gets "Router Solicitation"

`poisson_dist()`: Generates packet of different lengths based on the poisson distribution function

## SAMPLE RUNS OF THE MODULE -

For user\_limit = 20: (PACKET LENGTH vs DEPARTURE TIME)



For user\_limit = 100: (PACKET LENGTH vs DEPARTURE TIME)

