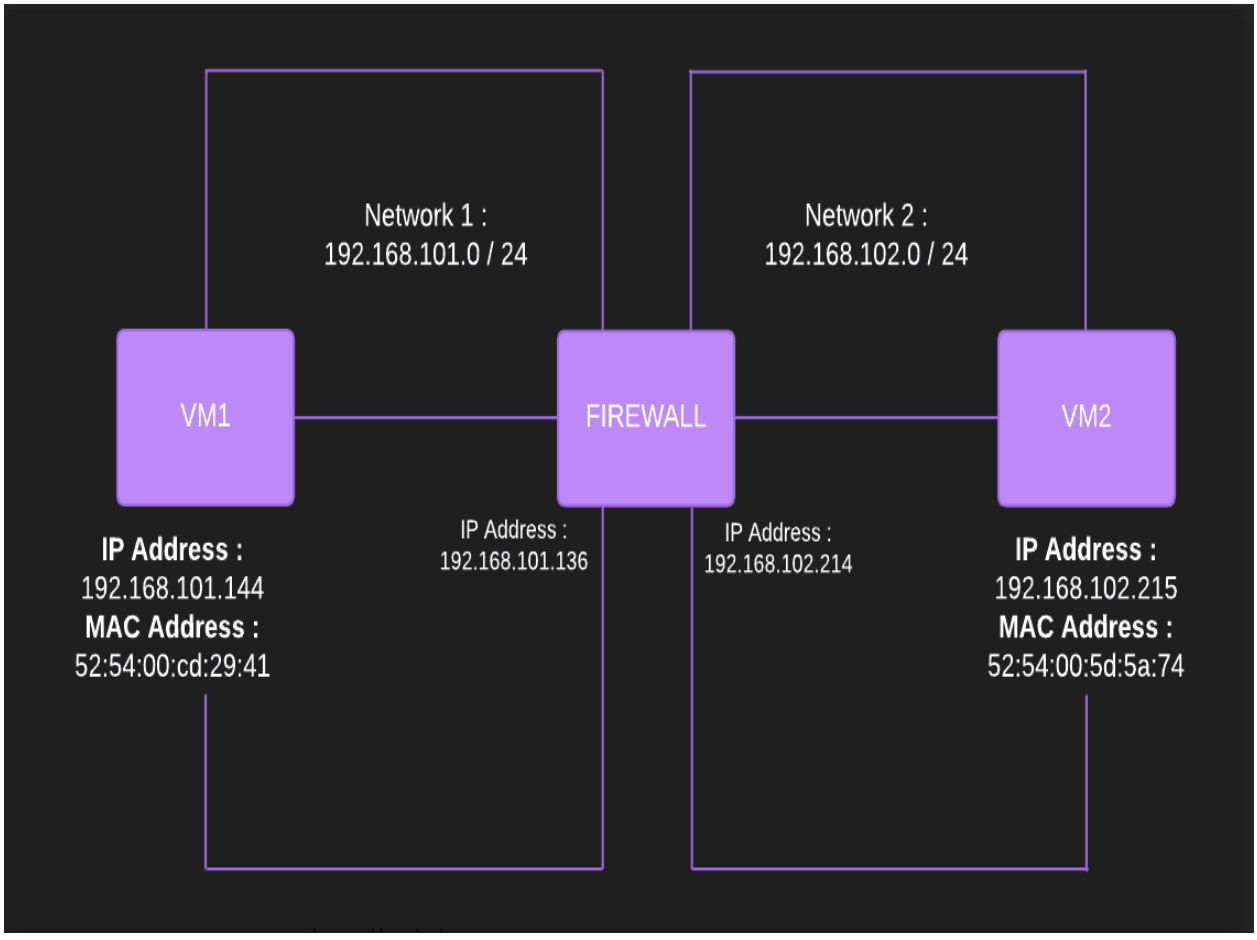
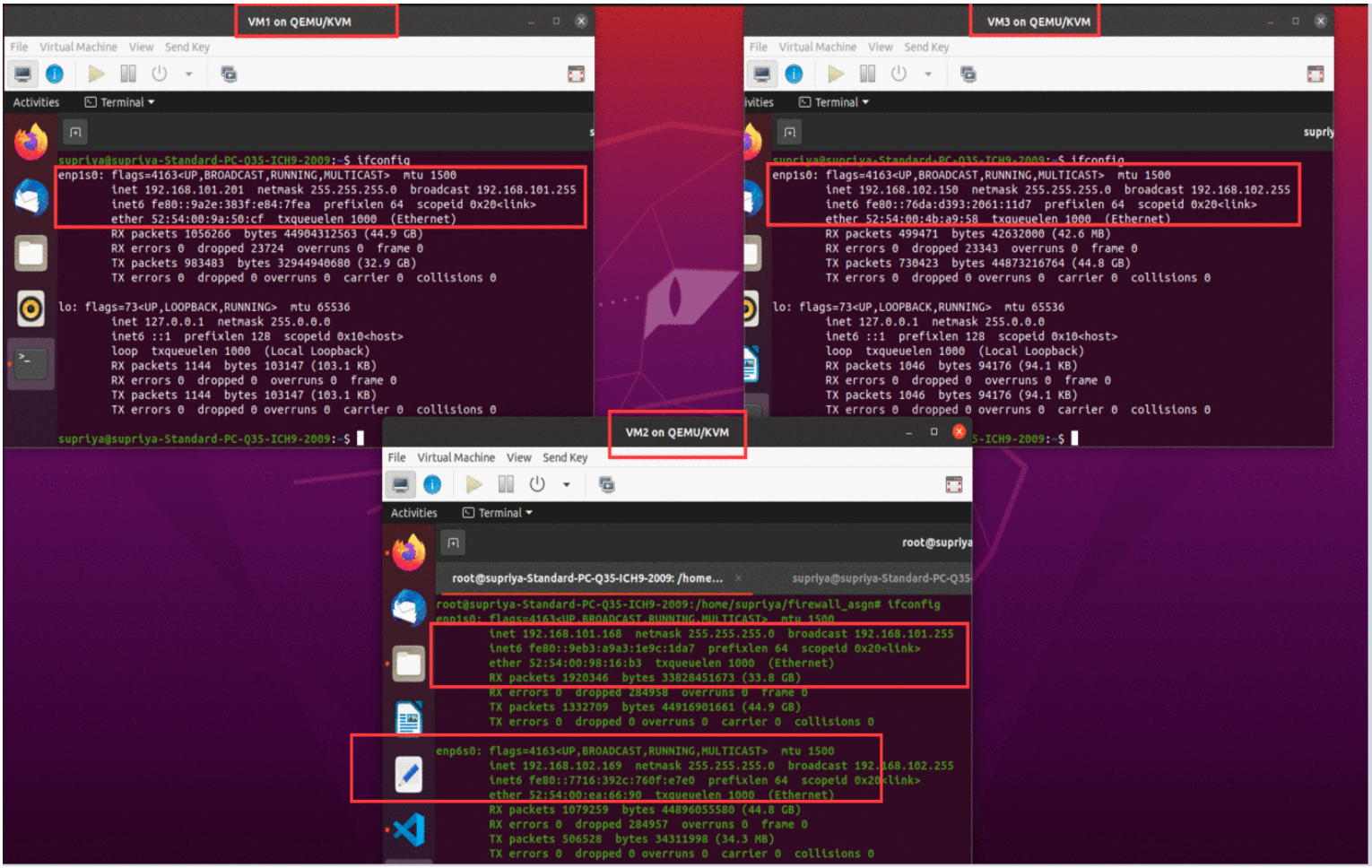
Firewall Assignment

Group Members –
K Saravanan (cs22mtech12007)
Neel Yogendra Kansagra (cs22mtech11002)

Network Configuration and Setup:

We've created 3 Virtual Machines where VM2 acts as Firewall, VM1 acts as an internal network that we want to protect from, VM3 acts as external host. Also we have created networks. VM1 and interface 1 of firewall connected by network 1 and VM2 and interface 2 of firewall connected by network 2.





VM3 is connected to the host via bridge(192.168.102.1 in this case). VM3 is able to ping the internet.

Task 1 :

There are two files simple_firewall.py and adv_firewall.py. First file implements a simple firewall with two network interface cards(enp1s0 and enp6s0) connecting to the external network (internet) and the internal network. Mode of operation is using a router(visible).

Ping external network from internal network. Traffic first goes to the firewall, where the MAC address of the destination is changed from firewall to external host. Firewall sends the packet to the external host.

Hard-coded logic:

```
def vm_1_inf(inf1, inf2):

    while True:
        dropped = False
        raw_data = inf1.recvfrom(65535)
        pkg, new_raw = packet_parse(raw_data[0])
        for i in pkg:
            print(i + ":")
            for j in pkg[i]:
                if str(pkg[i][j]) == '188.166.104.231':
                    print("\033[1;31m" + j + ": " + str(pkg[i][j]), end = '\t')
                    print("\033[1;32m")
                    dropped = True
                else:
                    print("\033[1;32m" + j + ": " + str(pkg[i][j]), end = '\t')
            print('\n')
        print("\033[1;32m ==========================")
        if(not dropped):
            inf2.sendall(new_raw)
```

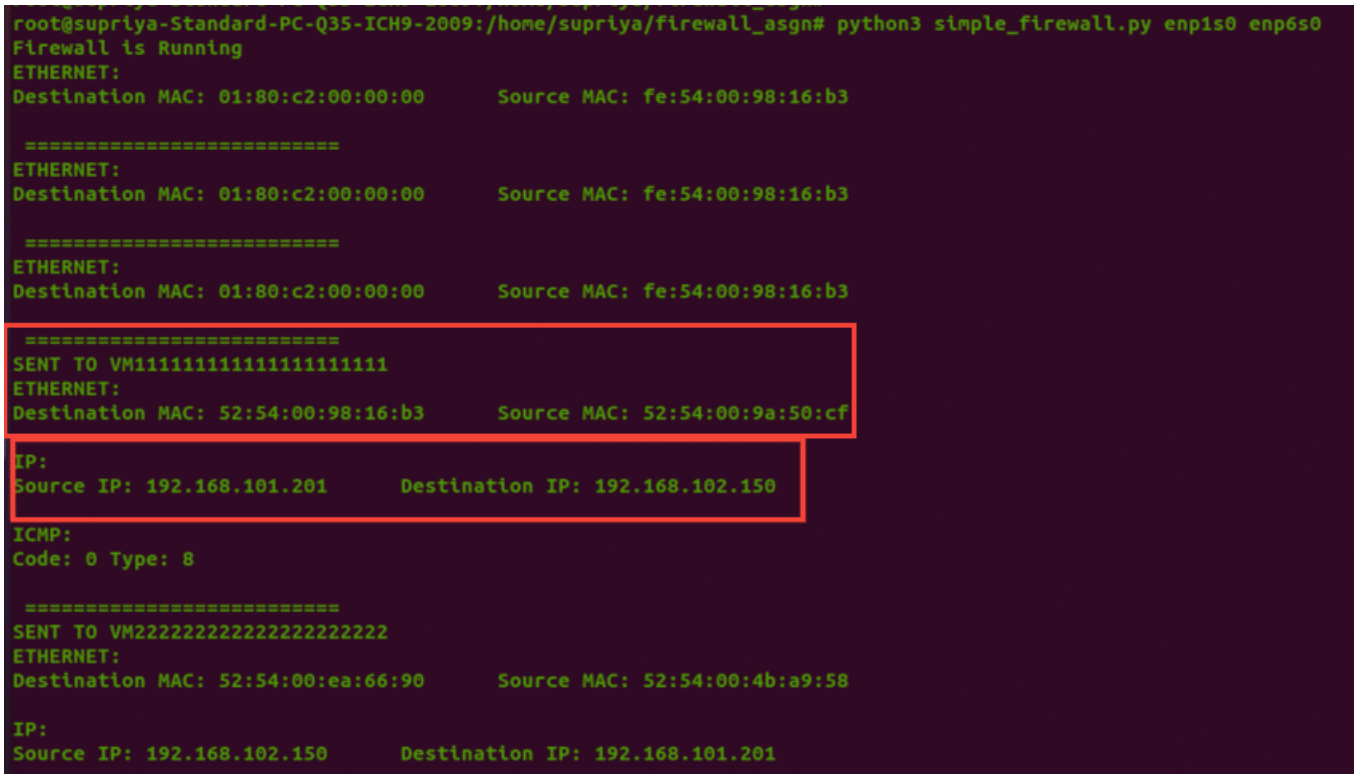
Routes at internal network & pinging external network from internal network:

```
supriya@supriya-Standard-PC-Q35-ICH9-2009:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         _gateway        0.0.0.0         UG    0      0        0 enp1s0
default         supriya-Super-S 0.0.0.0         UG    20100  0        0 enp1s0
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 enp1s0
192.168.101.0   0.0.0.0         255.255.255.0   U     100    0        0 enp1s0
supriya@supriya-Standard-PC-Q35-ICH9-2009:~$ ping 192.168.102.150
PING 192.168.102.150 (192.168.102.150) 56(84) bytes of data.
64 bytes from 192.168.102.150: icmp_seq=1 ttl=64 time=1.06 ms
64 bytes from 192.168.102.150: icmp_seq=2 ttl=64 time=0.925 ms
64 bytes from 192.168.102.150: icmp_seq=3 ttl=64 time=1.63 ms
64 bytes from 192.168.102.150: icmp_seq=4 ttl=64 time=1.12 ms
64 bytes from 192.168.102.150: icmp_seq=5 ttl=64 time=1.16 ms
^C
--- 192.168.102.150 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 0.925/1.178/1.632/0.239 ms
```

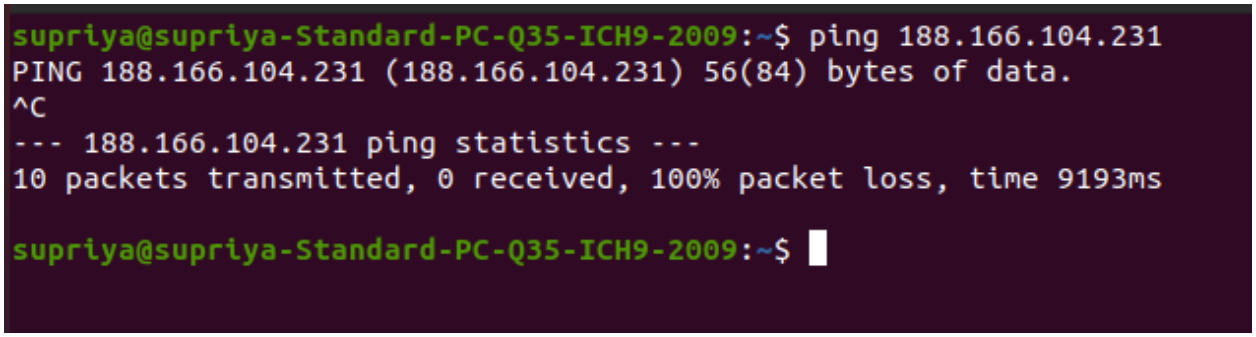
Routes at external network & pinging google.com from external network

```
supriya@supriya-Standard-PC-Q35-ICH9-2009:~$ sudo route add 192.168.101.201 gw 192.168.102.169
supriya@supriya-Standard-PC-Q35-ICH9-2009:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         supriya-Super-S 0.0.0.0         UG    0      0        0 enp1s0
link-local      0.0.0.0         255.255.0.0     U     1000   0        0 enp1s0
192.168.101.201 supriya-Standar 255.255.255.255 UGH   0      0        0 enp1s0
192.168.102.0   0.0.0.0         255.255.255.0   U     100    0        0 enp1s0
supriya@supriya-Standard-PC-Q35-ICH9-2009:~$ ping google.com
PING google.com (142.250.77.174) 56(84) bytes of data.
64 bytes from maa05s17-in-f14.1e100.net (142.250.77.174): icmp_seq=1 ttl=55 time=19.7 ms
64 bytes from maa05s17-in-f14.1e100.net (142.250.77.174): icmp_seq=2 ttl=55 time=19.2 ms
64 bytes from maa05s17-in-f14.1e100.net (142.250.77.174): icmp_seq=3 ttl=55 time=19.5 ms
```

Firewall running:



It is observed the firewall drops the packet that is received from *188.166.104.231* before it passes it along to the VM1 i.e internal network.



Task 2:

In this task, we have to add rules at different layers like Ethernet, IP, TCP, UDP and ICMP. For this we have added the "**add_rule()**" method. First we are taking input from the user in which header field the user wants to add the rule. It's like in the Ethernet layer we can add Source MAC and Destination MAC which we want to restrict. In the IP header we can restrict IPV4 Source IP and IPV4 Destination IP. Similarly, for IPV6 we can restrict IPV6 Source IP and Destination IP. For TCP and UDP we can restrict the start and end port of both Source and Destination port. Lastly, for ICMP v4/6 the type and code can be restricted.

In order to delete rules we have implemented the **"delete_rule()"** method. In this we pass the rule id of the rule that we want to delete.

For updating the rules, **"update_rule"** is used. Here we pass the id and choice. Here choice directs to the field that we want to update. For example, if we want to update source mac we pass choice as 1.

For displaying the stats and the graphs, **"show_stat"** function is executed. It outputs the graph of Number of packets accepted/discarded Vs Time.
Rules are saved in a rule_file.json.

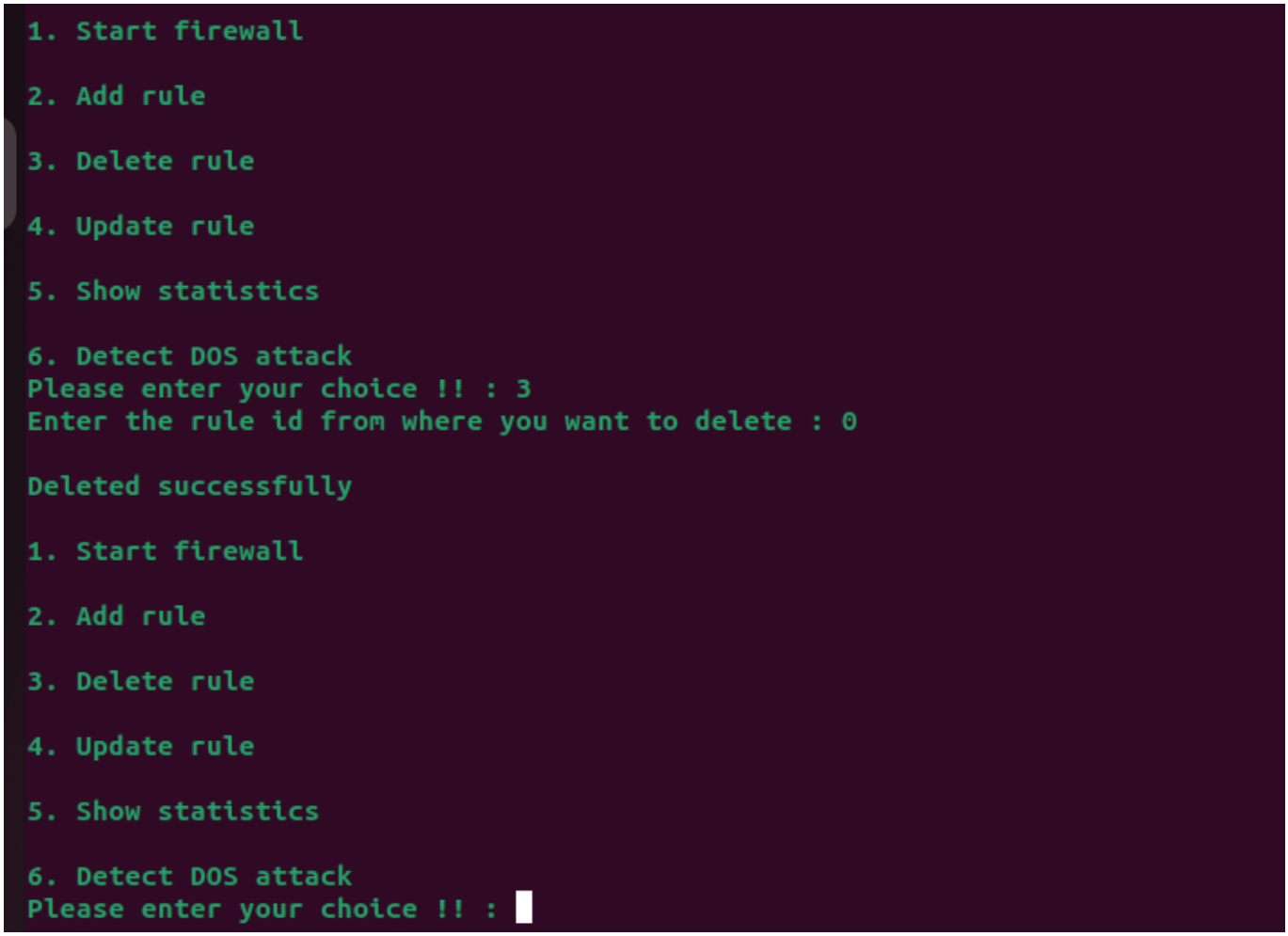
Adding a new rule

```
1. Start firewall

2. Add rule

3. Delete rule

4. Update rule

5. Show statistics

6. Detect DOS attack
Please enter your choice !! : 2

1. Ethernet Header

2. IPV4 Header

3. IPV6 Header

4. TCP Header

5. UDP Header

6. ICMP Header

 Enter where you want to make changes : 2

Enter a uniqe ID : 101

Change Source IP (0/1)1

Enter the source IPV4 address you want to restrict :  192.168.128.117

Change Destination IP (0/1)1

Enter the destination IPV4 address you want to restrict :  192.168.129.156
```

Updating a rule

```
1. Start firewall

2. Add rule

3. Delete rule

4. Update rule

5. Show statistics

6. Detect DOS attack
Please enter your choice !! : 4
Enter the rule id from where you want to Update : 0

1. Update Source MAC

2. Update Destination MAC

3. Update Source IPV4

4. Update Destination IPV4

5. Update Source IPV6

6. Update Destination IPV6

7. Update Source Port

8. Update Destination Port

 Enter where you want to make changes : 7

Enter the range of source port you want to restrict (Enter same port if do not want to provide range):

Enter the starting port : 0

Enter the ending port : 100

1. Start firewall
```

Deleting the rule with rule id:



TCP Traffic monitoring

```
vm1@vm1-Standard-PC-i440FX-PIIX-1996:~$ sudo su
[sudo] password for vm1:
root@vm1-Standard-PC-i440FX-PIIX-1996:/home/vm1# ./netconfig
root@vm1-Standard-PC-i440FX-PIIX-1996:/home/vm1# iperf -c 192.168.102.145
------------------------------------------------------------
Client connecting to 192.168.102.145, TCP port 5001
TCP window size: 85.0 KByte (default)
------------------------------------------------------------
[  3] local 192.168.101.205 port 38514 connected with 192.168.102.145 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec   574 MBytes   481 Mbits/sec
root@vm1-Standard-PC-i440FX-PIIX-1996:/home/vm1#
```

```
vm2@vm2-Standard-PC-i440FX-PIIX-1996:~$ iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  128 KByte (default)
------------------------------------------------------------
[  4] local 192.168.102.145 port 5001 connected with 192.168.101.205 port 38514
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0-10.2 sec   574 MBytes   470 Mbits/sec
```

UDP Traffic Monitoring

```
root@vm1-Standard-PC-i440FX-PIIX-1996:/home/vm1# iperf -c 192.168.102.145 -u
------------------------------------------------------------
Client connecting to 192.168.102.145, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 192.168.101.205 port 45211 connected with 192.168.102.145 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec
[  3] Sent 893 datagrams
[  3] Server Report:
[  3]  0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec   0.133 ms    0/  893 (0%)
root@vm1-Standard-PC-i440FX-PIIX-1996:/home/vm1#
```

ICMP traffic monitoring

```
supriya@supriya-Standard-PC-Q35-ICH9-2009:~$ ping 192.168.102.150
PING 192.168.102.150 (192.168.102.150) 56(84) bytes of data.
64 bytes from 192.168.102.150: icmp_seq=1 ttl=64 time=1.43 ms
64 bytes from 192.168.102.150: icmp_seq=2 ttl=64 time=1.03 ms
64 bytes from 192.168.102.150: icmp_seq=3 ttl=64 time=0.813 ms
^C
--- 192.168.102.150 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.813/1.090/1.429/0.255 ms
```

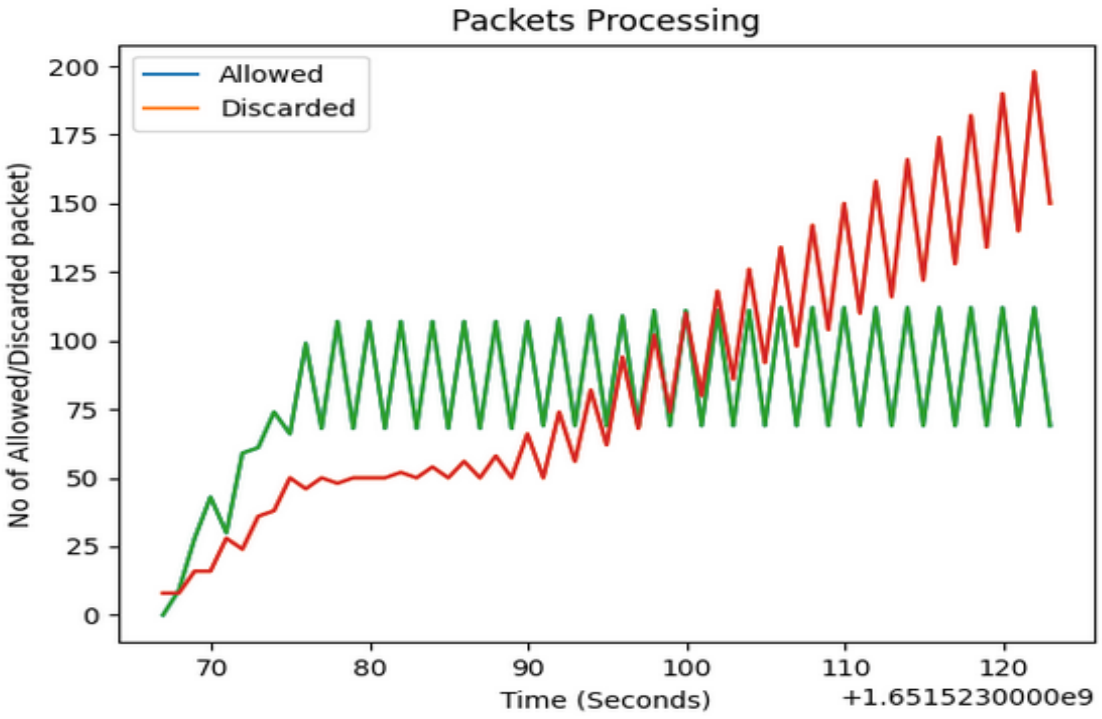
Task 3:

In this task, we analyzed the packet per second handling power of the system and also the performance of the system with respect to matching fields and number of rules.

Below screenshot shows the statistics, that is the number of packets accepted and discarded. According to the number of packets accepted and discarded a graph is generated.

```
1. Add rule

2. Delete rule

3. Update rule

4. Show statistics

5. Detect DOS attack

6. Exit
Please enter your choice !! : 4

Statistics of Number of Acccepted/Discarded Packets Vs Time
No of packets allowed :  77

No of packets dropped :  42
```

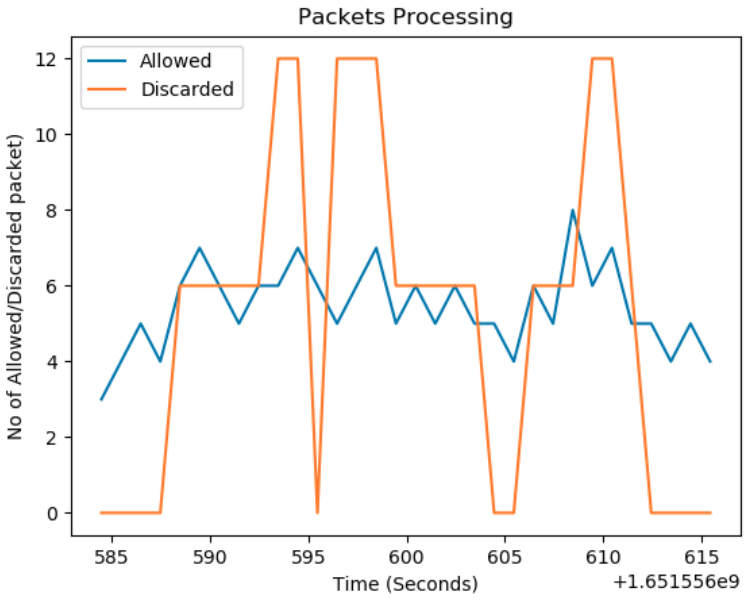
Statistics Display



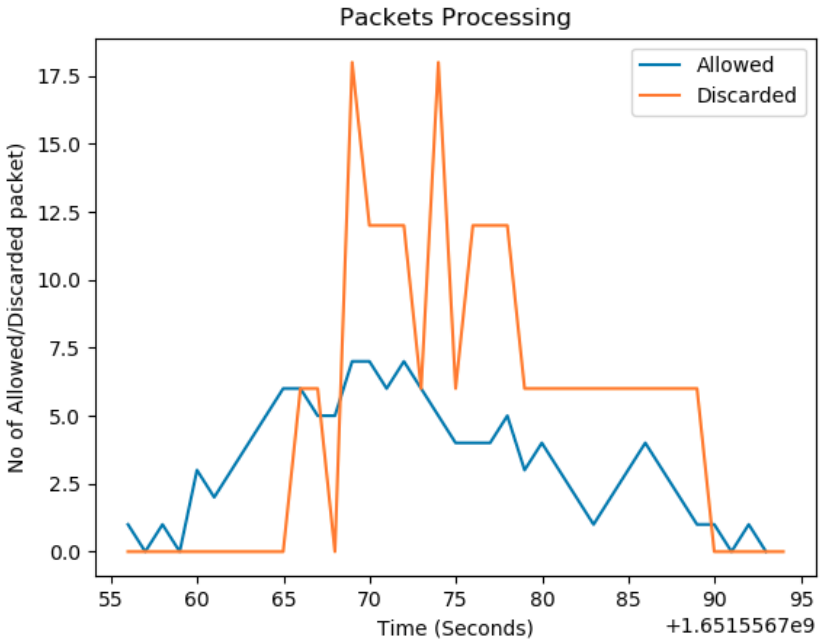
Packet Processing plot with respect to Number of Packets Accepted and Discarded

Performance of Packets with respect to number of rules

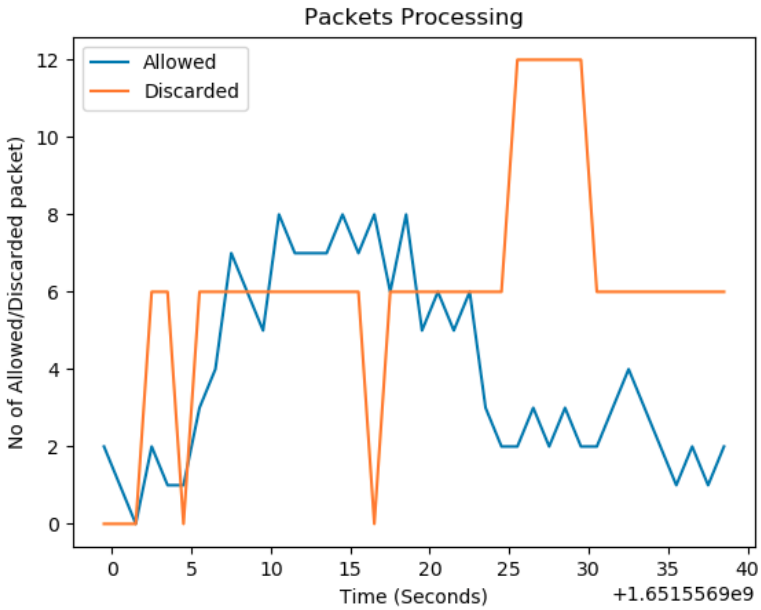
1. **Number of Rules = 10**



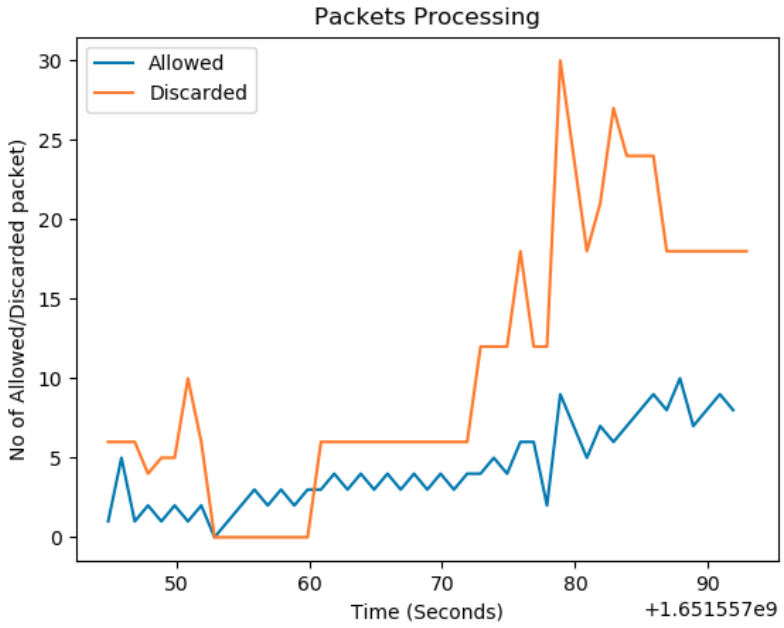
2. **Number of Rules = 20**



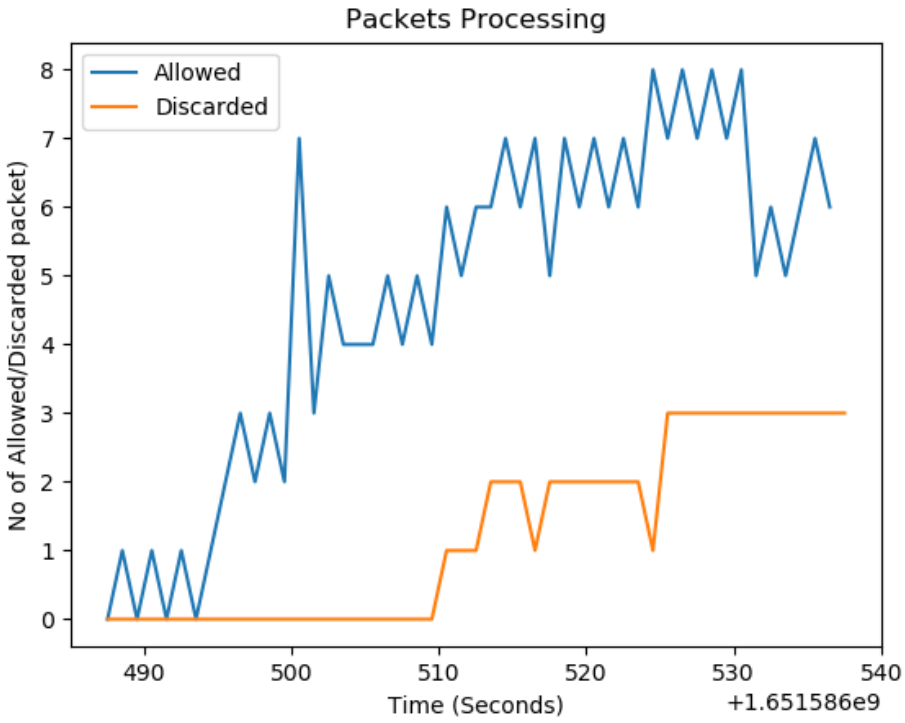
3. **Number of Rules = 50**



4. **Number of Rules = 100**

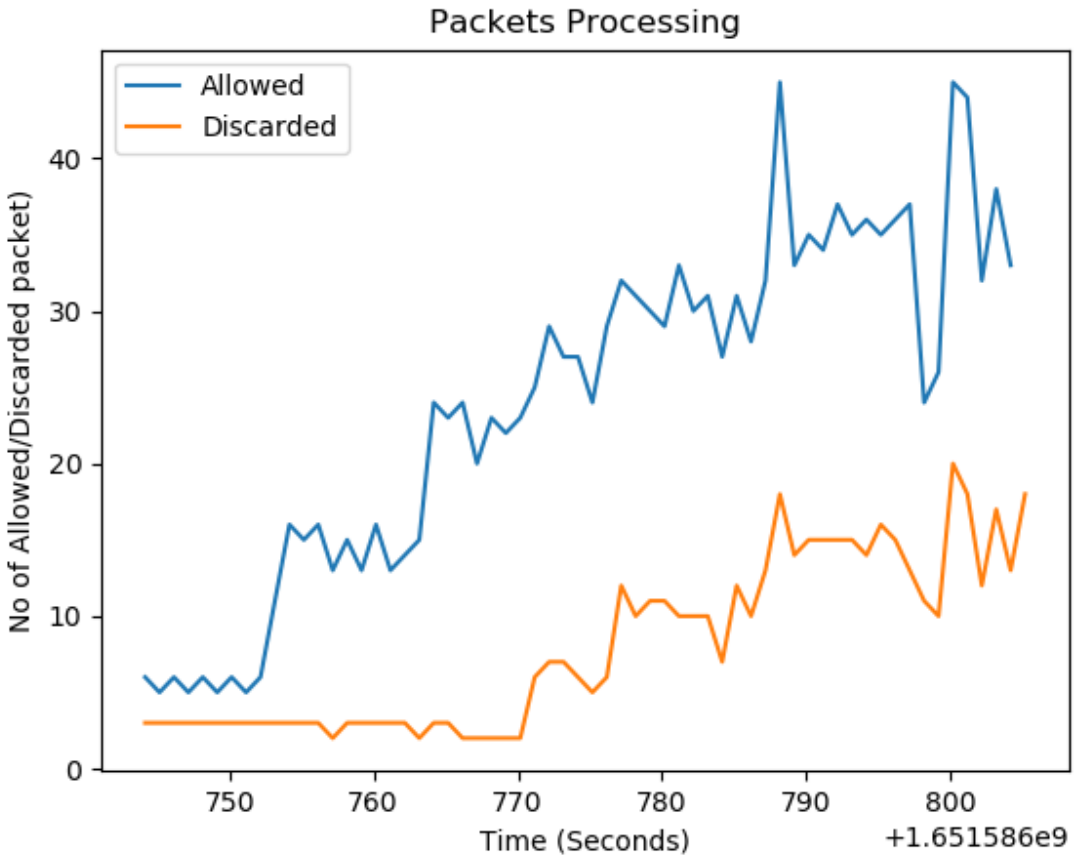


5. **<u>PPS with Maximum matching 2 with 100 rules</u>**



Observation: As the number of rules increases, the PPS (packet per second) decreases. This happens because the number of matching fields increases that means more time is spent in comparing the fields and hence PPS decreases.

https://docs.google.com/document/d/1qi5pHMdTg2fgrl1LE9ZZBPgjYX2Hac3ggbYtjtN6ewo/edit
Benchmarking for 5 users requesting ping at the rate of 1 packet per 0.2 second checked across 100 rules.



As it is observed that with such work load. Firewall is able to process upto **45 packets per second**

The traffic generators used in the given graphs are

1. Ping — ping <IP address>
2. Iperf — iperf -s (for server)
   Iperf -c <IP_of_server>

Task 4:

DoS Attack : Denial of Service attack is launched by overloading a machine preventing legitimate users from accessing it.

We created a **dos_attack** dictionary which consists of the count of each ip address that enters the system. The user gives dos_threshold as input which sets the threshold, if any ip address crosses that threshold limit for particular time then that means it is trying to launch a DOS attack and the program prints **"DOS Detected"**.

Case 1: DoS Detected

```
1. Start firewall

2. Add rule

3. Delete rule

4. Update rule

5. Show statistics

6. Detect DOS attack
Please enter your choice !! : 6
Want to Turn on DoS detection? (y/n)y
Enter new threshold limit :
8

 DoS Detected

 Count of each IP address
{'192.168.101.201': 17, '192.168.102.1': 1}
```

Here, the threshold is set to 8 by the user and one of the ip addresses is crossing that limit. THat's why DOS Detected.

Case 2: DoS not Detected

```
1. Start firewall

2. Add rule

3. Delete rule

4. Update rule

5. Show statistics

6. Detect DOS attack
Please enter your choice !! : 6
Want to Turn on DoS detection? (y/n)y
Enter new threshold limit :
20

 DoS not Detected

 Count of each IP address
{'192.168.101.201': 9, '192.168.101.168': 10, '192.168.102.169': 6, '192.168.101.1': 5, '192.168.102.1': 4, '52.182.141.63': 2}
```

Here, the threshold is set to 20 by the user, since no ip address crosses that limit. That's why DoS is not Detected.

We also have implemented an alternative implementation of DoS program which takes into consideration parameters like

Time and Threshold