CSE 232: Programming Assignment 3
Using Linux iptables
Due date: Oct 27, 2024
Total: 21 points

Q1. VM Setup and Basic Configuration
Objective: Set up four VMs and configure network routing with VM2 as a gateway.

a) **IP Configuration and Routing**: Each VM was assigned IP addresses according to the setup diagram, with VM1 as the client, VM2 as the gateway, and VM3 and VM4 as servers. Routing was configured on VM2 to forward incoming traffic to the servers.

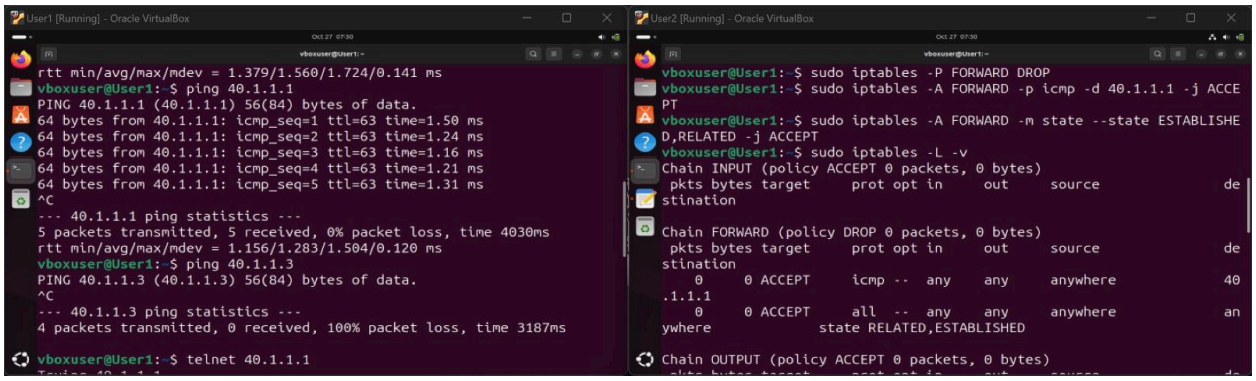
b) **Gateway Forwarding Configuration**: Using `iptables` on VM2, packet forwarding was enabled by setting up IP forwarding:

```
sudo sysctl -w net.ipv4.ip_forward=1
```

Q2. Traffic Filtering at the Gateway
Objective: Implement firewall rules on VM2 to selectively block traffic.

Blocking All Traffic Except Ping to 40.1.1.1: An `iptables` rule was added to block all traffic to server `40.1.1.1/24` except for ICMP (ping) traffic:

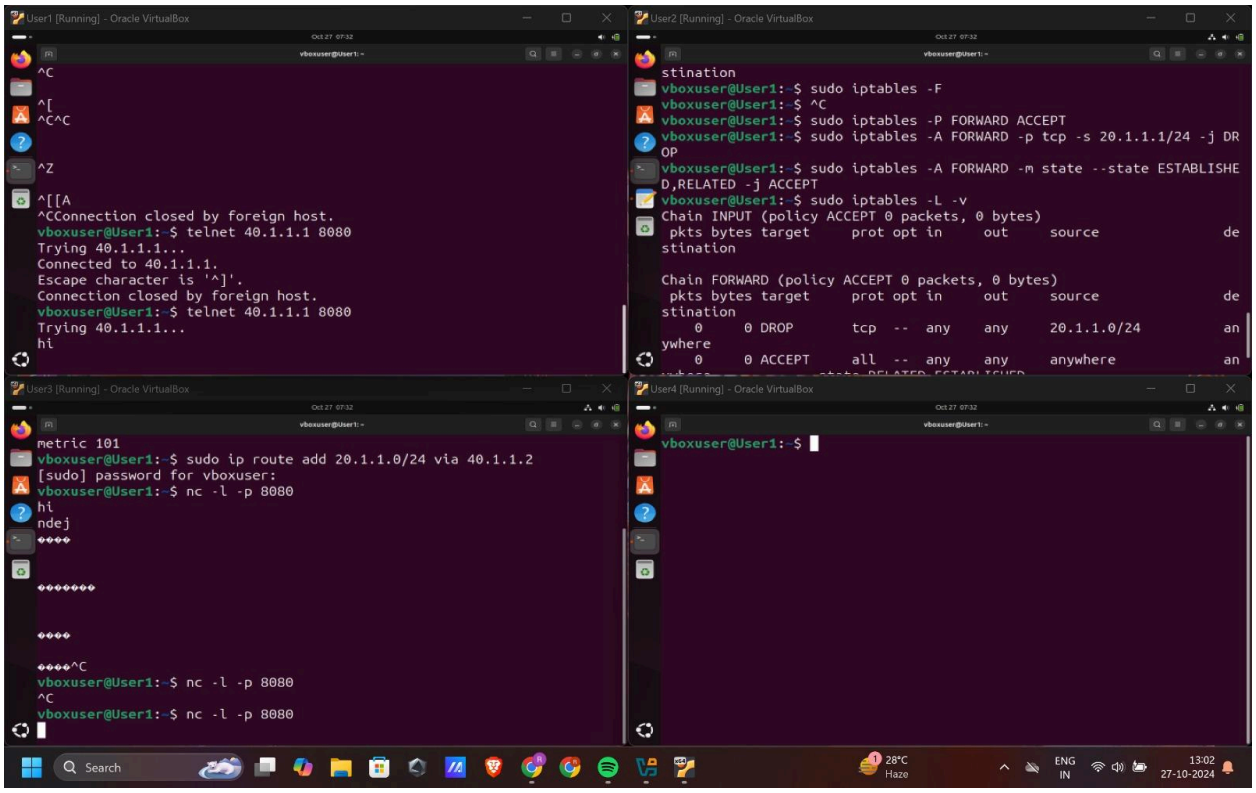
```
sudo iptables -A FORWARD -d 40.1.1.1 -p icmp -j ACCEPT
sudo iptables -A FORWARD -d 40.1.1.1 -j DROP
```



a) Output Reference: Screenshots showed successful ping requests and blocked TCP/UDP requests to 40.1.1.1.

Blocking TCP Traffic Initiated by 20.1.1.1: An `iptables` rule was added to block all TCP traffic from `20.1.1.1/24`:

```
sudo iptables -A FORWARD -s 20.1.1.1 -p tcp -j DROP
```

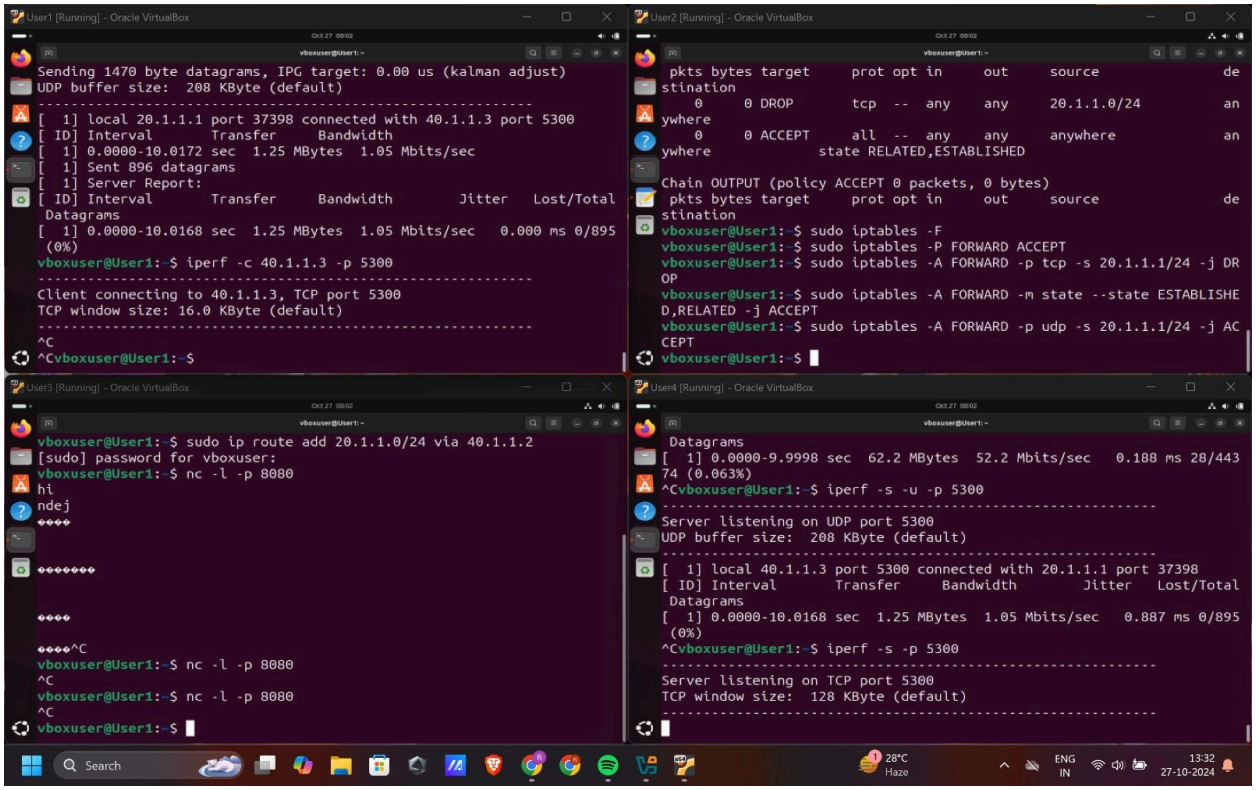


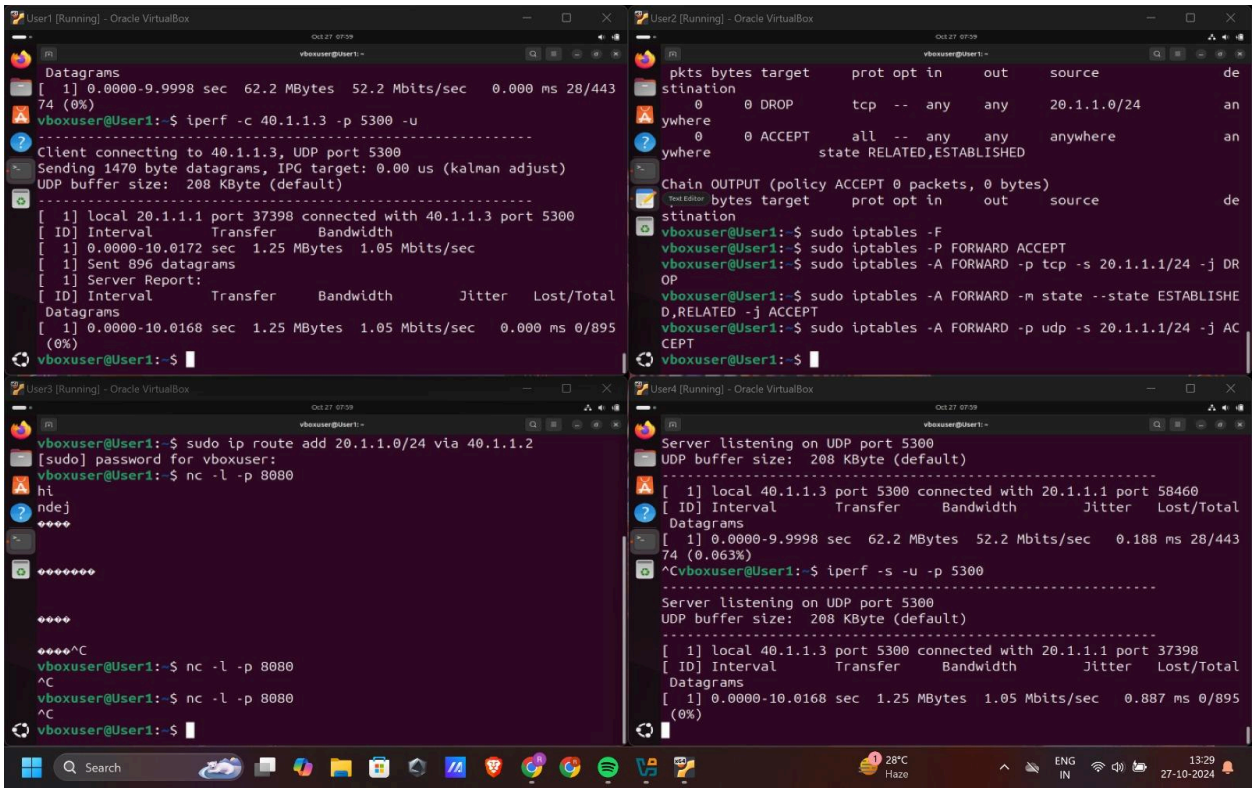
b) Output Reference: Screenshots confirmed that TCP traffic from 20.1.1.1 was blocked, while other protocols were unaffected.

Q3. Testing Bandwidth and RTT with `iperf`
Objective: Measure TCP/UDP bandwidth between VMs and analyze RTT between client and servers.

a) **TCP and UDP Bandwidth Testing**: Using `iperf`, bandwidth was measured between `20.1.1.1/24` and `40.1.1.3/24`.

Command: `iperf -c 40.1.1.3 -p [port] -t [time]`





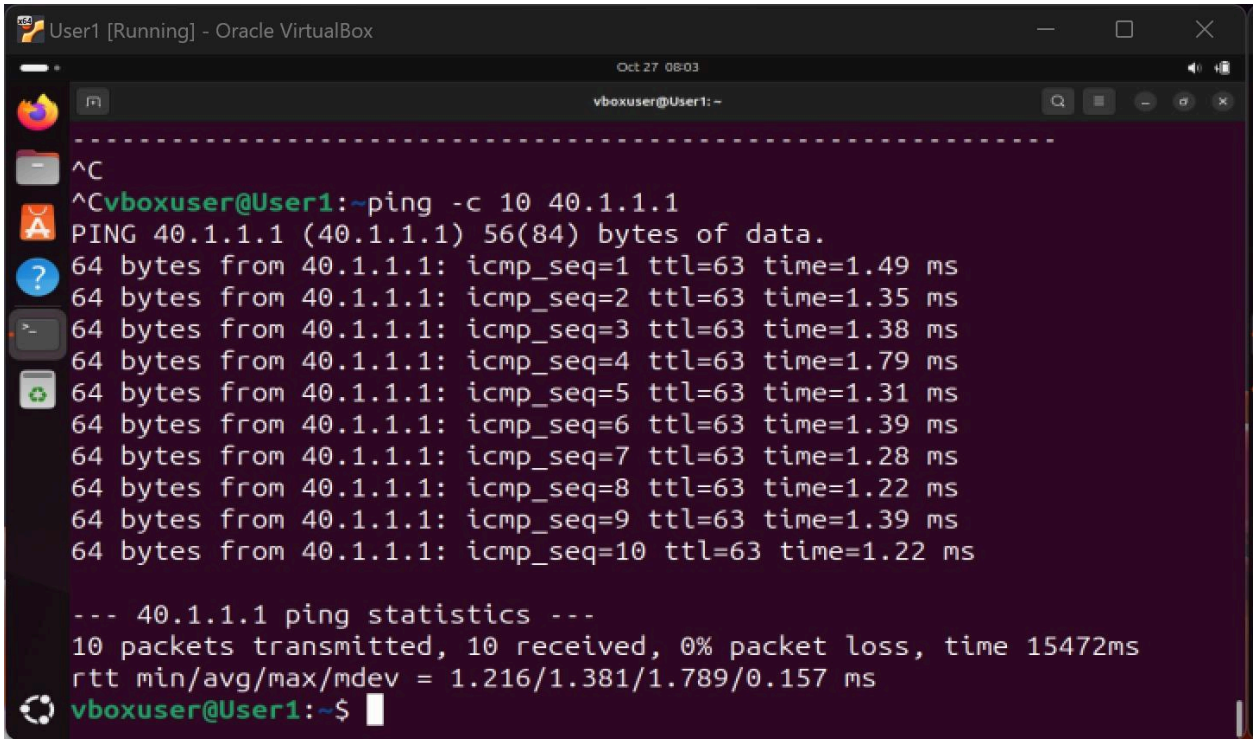
Output Reference: The bandwidth test screenshots indicated TCP and UDP bandwidth results.

b) **RTT Measurement**: RTT was measured from VM1 (20.1.1.1) to both servers.

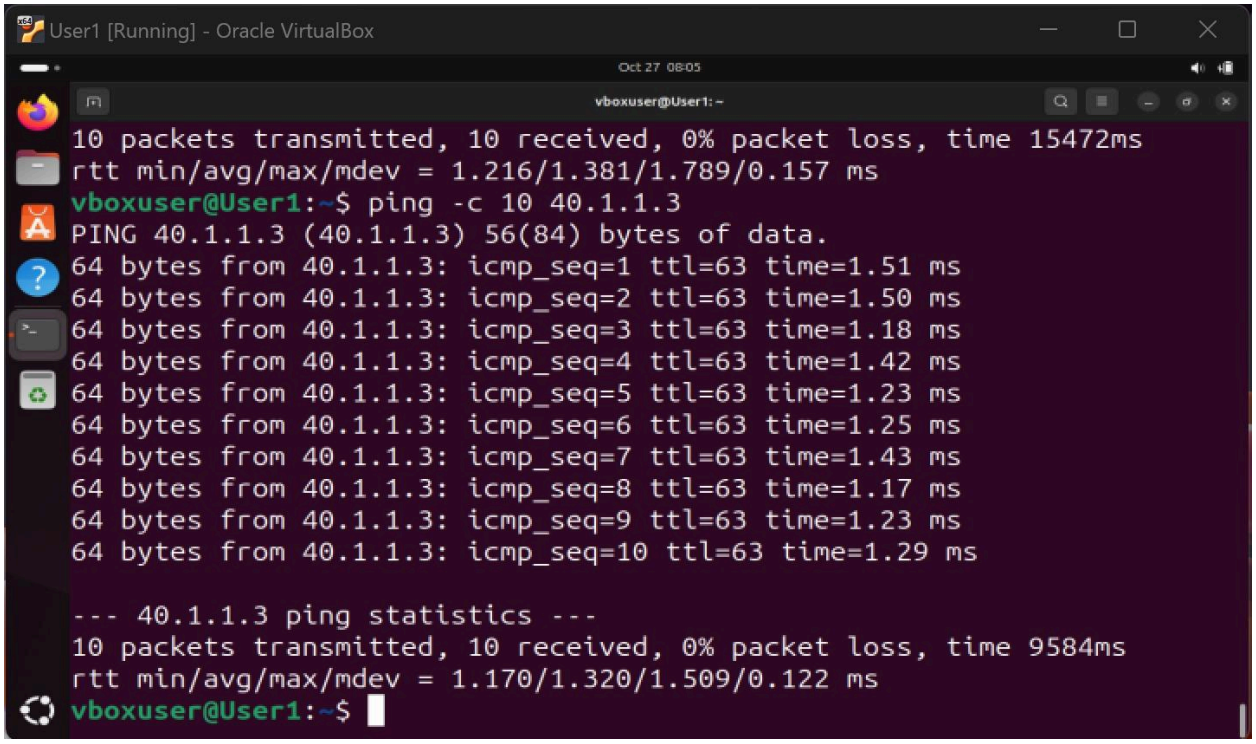
Ping Command: `ping -c 10 [destination IP]`

RTT Results: Results showed a comparison of RTT for both servers, with a notable RTT difference attributed to network load or distance.

i)



ii)



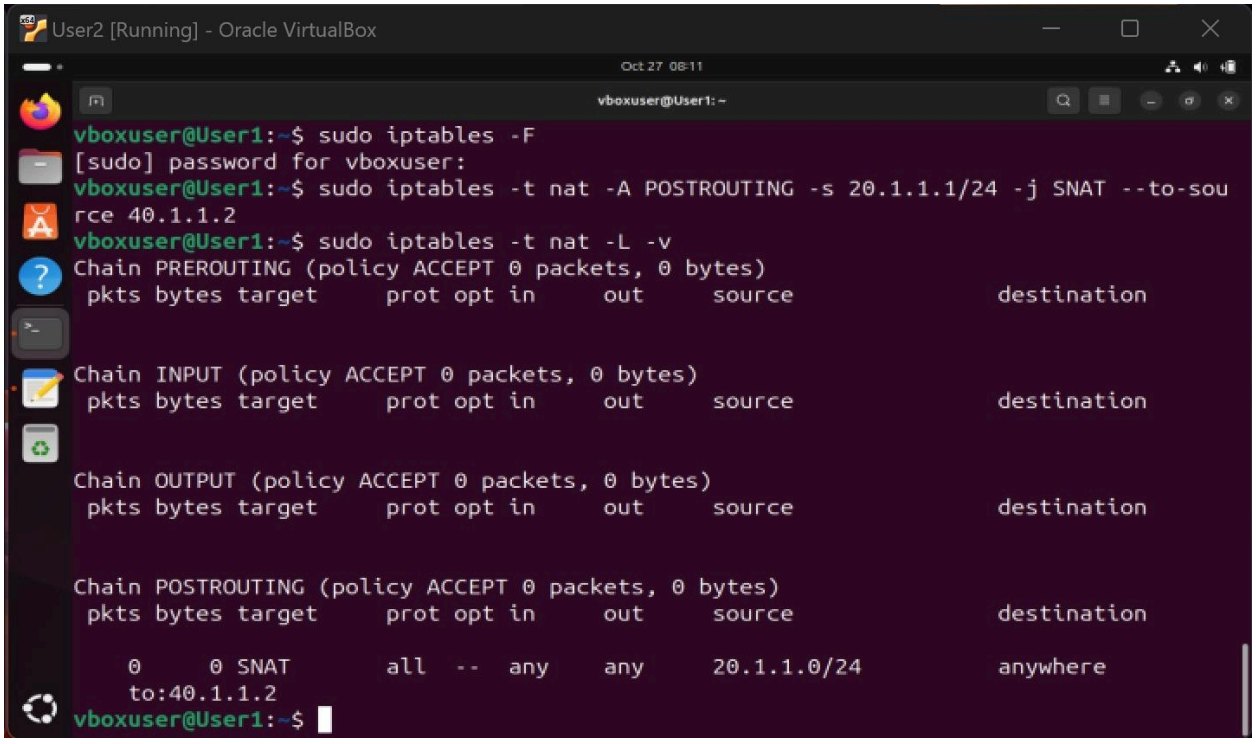
> *Output Reference*: Screenshots captured minimum, average, and maximum RTT for both 40.1.1.1 and 40.1.1.3, indicating the observed differences.

iii) Much difference was not observed between 40.1.1.1 and 40.1.1.3 but 40.1.1.3 was seen to have slightly lower RRT than 40.1.1.1.

Q4. Network Address Translation (NAT) Configuration
Objective: Implement source and destination NAT on VM2 to rewrite packet headers.

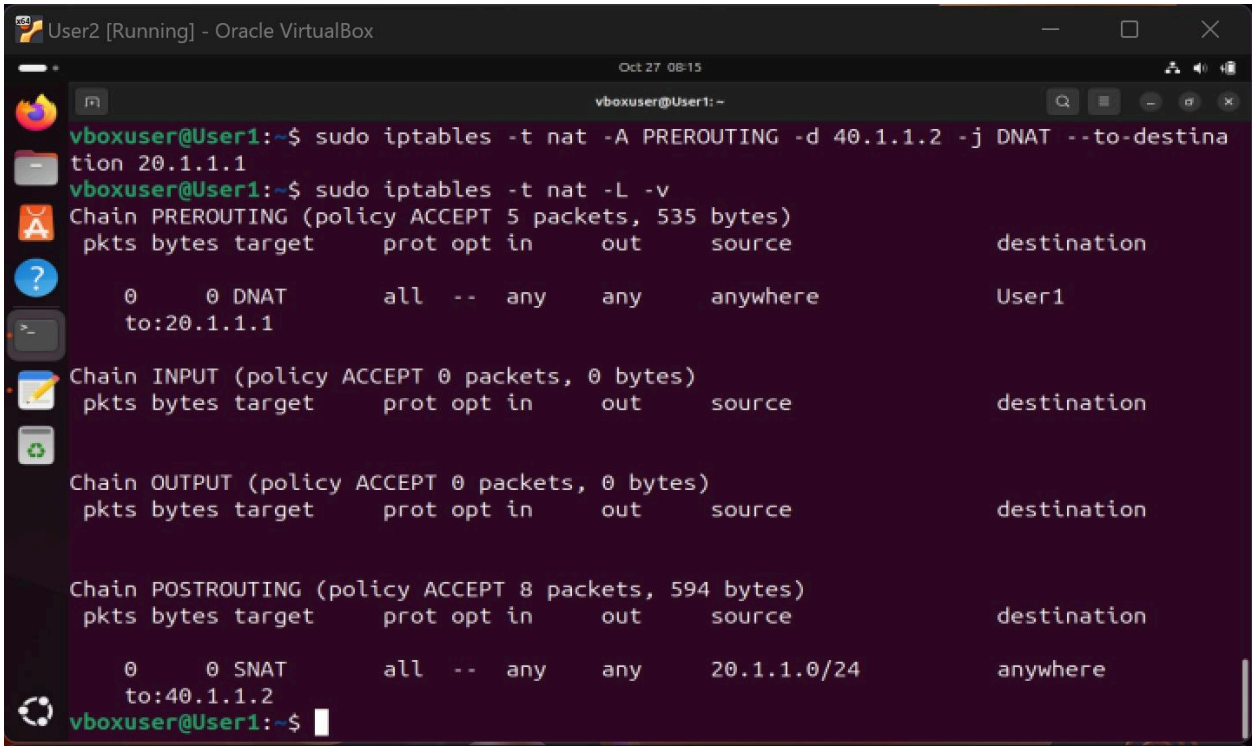
a) **Source NAT (SNAT)**: A rule was configured on VM2 to change the source IP of packets from `20.1.1.1/24` to `40.1.1.2/24`:

```
sudo iptables -t nat -A POSTROUTING -s 20.1.1.1 -j SNAT --to-source 40.1.1.2
```



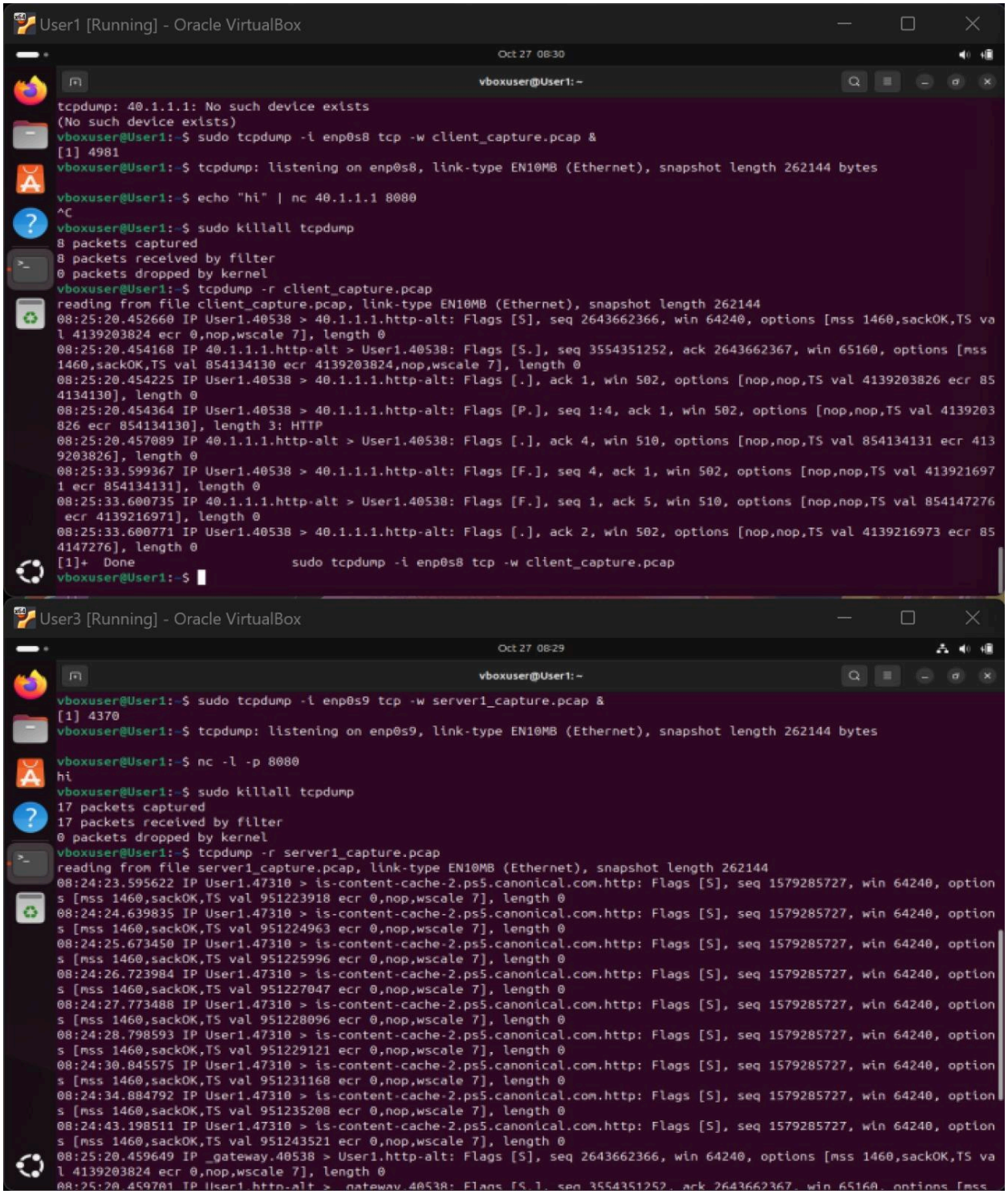
b) **Destination NAT (DNAT)**: Incoming packets addressed to `40.1.1.2` were modified back to the original source IP (`20.1.1.1`):

```
sudo iptables -t nat -A PREROUTING -d 40.1.1.2 -j DNAT
--to-destination 20.1.1.1
```



Validation with `tcpdump`: Traffic was observed at each VM using `tcpdump` to confirm IP changes:

Command: `sudo tcpdump -i [interface] host [destination IP]`



Output Reference: Screenshots from `tcpdump` on VMs confirmed that NAT was functioning correctly, showing translated source and destination addresses.

Q5. Load Balancing at the Gateway

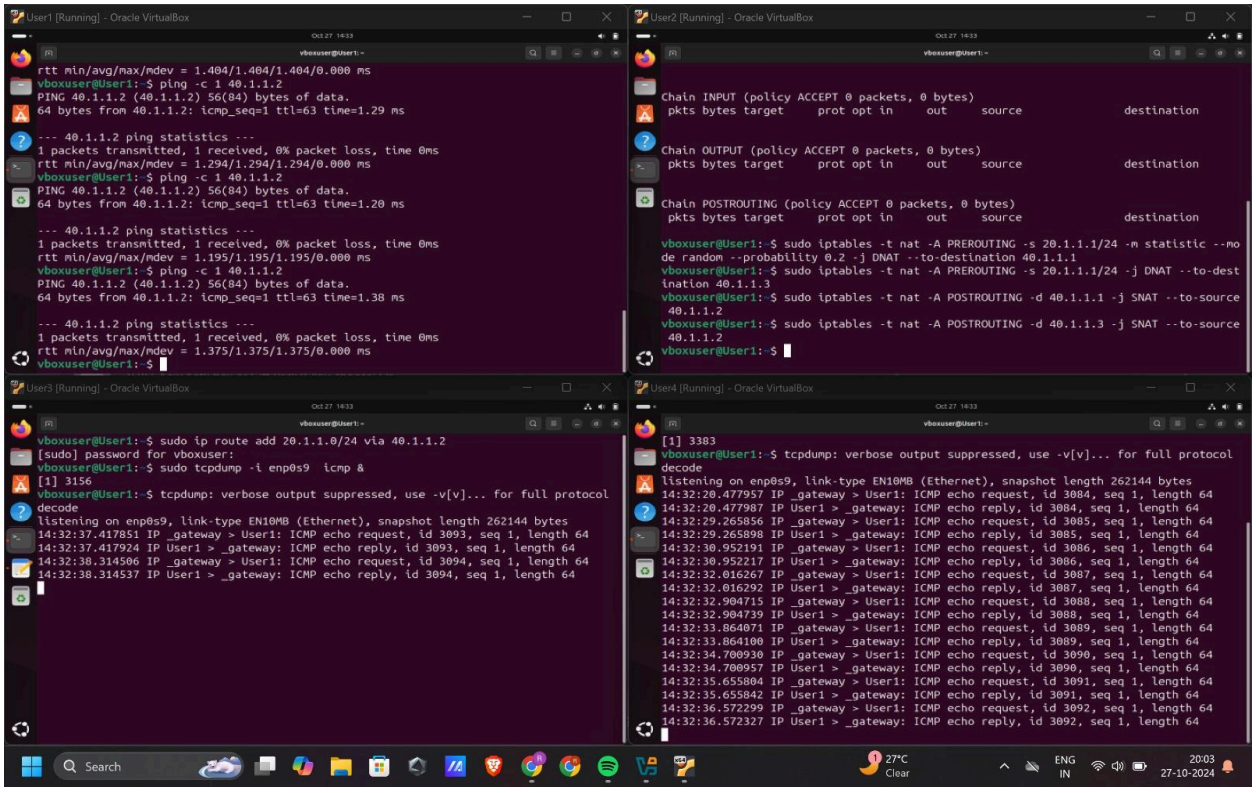
Objective: Distribute traffic from `20.1.1.1/24` to servers `40.1.1.1/24` and `40.1.1.3/24` with a probability-based load balancing.

Setting up Probability-based DNAT: The following `iptables` rules were set to direct 80% of traffic to `40.1.1.3` and 20% to `40.1.1.1`, based on RTT results showing `40.1.1.3` had lower latency:

```
sudo iptables -t nat -A PREROUTING -d 20.1.1.2 -m statistic --mode
random --probability 0.2 -j DNAT --to-destination 40.1.1.1
```

```
sudo iptables -t nat -A PREROUTING -d 20.1.1.2 -j DNAT
--to-destination 40.1.1.3
```

Testing with Ping Packets: From VM1, multiple ping requests were sent to validate traffic distribution. `tcpdump` on the servers confirmed the 80-20 split:



Output Reference: `tcpdump` results indicated that approximately 80% of packets reached `40.1.1.3`, and 20% reached `40.1.1.1`, confirming the load balancing configuration.