Q1.a)

Client:

```
  GNU nano 6.2                          /etc/netplan/02-netcfg.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s8:
      addresses: [20.1.1.1/24]
      dhcp4: no
      gateway4: 20.1.1.2
```

```
abhirup@client:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe83:8794  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:83:87:94  txqueuelen 1000  (Ethernet)
        RX packets 7  bytes 2010 (2.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 16  bytes 1816 (1.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 20.1.1.1  netmask 255.255.255.0  broadcast 20.1.1.255
        inet6 fe80::a00:27ff:fe92:7735  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:92:77:35  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 22  bytes 1506 (1.5 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 92  bytes 7348 (7.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 92  bytes 7348 (7.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
abhirup@client:~$ sudo ip route add 40.1.1.0/24 via 20.1.1.2
abhirup@client:~$ ip route show
default via 20.1.1.2 dev enp0s8 proto static
default via 10.0.2.2 dev enp0s3 proto dhcp src 10.0.2.15 metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
10.0.2.2 dev enp0s3 proto dhcp scope link src 10.0.2.15 metric 100
20.1.1.0/24 dev enp0s8 proto kernel scope link src 20.1.1.1
40.1.1.0/24 via 20.1.1.2 dev enp0s8
192.168.1.1 via 10.0.2.2 dev enp0s3 proto dhcp src 10.0.2.15 metric 100
```

Server1:

```
  GNU nano 6.2                          /etc/netplan/02-netcfg.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s8:
      addresses: [40.1.1.1/24]
      dhcp4: no
      gateway4: 40.1.1.2
```

```
abhirup@server1:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe35:47f8  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:35:47:f8  txqueuelen 1000  (Ethernet)
        RX packets 33  bytes 7014 (7.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 64  bytes 5946 (5.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 40.1.1.1  netmask 255.255.255.0  broadcast 40.1.1.255
        inet6 fe80::a00:27ff:fec7:4ab0  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:c7:4a:b0  txqueuelen 1000  (Ethernet)
        RX packets 98  bytes 9100 (9.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 208  bytes 15878 (15.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 212  bytes 18664 (18.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 212  bytes 18664 (18.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
abhirup@server1:~$ sudo ip route add 20.1.1.0/24 via 40.1.1.2
abhirup@server1:~$ ip route show
default via 40.1.1.2 dev enp0s8 proto static
default via 10.0.2.2 dev enp0s3 proto dhcp src 10.0.2.15 metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
10.0.2.2 dev enp0s3 proto dhcp scope link src 10.0.2.15 metric 100
20.1.1.0/24 via 40.1.1.2 dev enp0s8
40.1.1.0/24 dev enp0s8 proto kernel scope link src 40.1.1.1
192.168.1.1 via 10.0.2.2 dev enp0s3 proto dhcp src 10.0.2.15 metric 100
```

Server 2:

```
  GNU nano 6.2                          /etc/netplan/02-netcfg.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s8:
      addresses: [40.1.1.3/24]
      dhcp4: no
      gateway4: 40.1.1.2
```

```
abhirup@server2:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:feee:4a5  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:ee:04:a5  txqueuelen 1000  (Ethernet)
        RX packets 46  bytes 9072 (9.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 81  bytes 7240 (7.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 40.1.1.3  netmask 255.255.255.0  broadcast 40.1.1.255
        inet6 fe80::a00:27ff:fea9:af82  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:a9:af:82  txqueuelen 1000  (Ethernet)
        RX packets 101  bytes 10748 (10.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 327  bytes 24472 (24.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 242  bytes 22830 (22.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 242  bytes 22830 (22.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
abhirup@server2:~$ sudo ip route add 20.1.1.0/24 via 40.1.1.2
abhirup@server2:~$ ip route show
default via 40.1.1.2 dev enp0s8 proto static
default via 10.0.2.2 dev enp0s3 proto dhcp src 10.0.2.15 metric 100
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.15 metric 100
10.0.2.2 dev enp0s3 proto dhcp scope link src 10.0.2.15 metric 100
20.1.1.0/24 via 40.1.1.2 dev enp0s8
40.1.1.0/24 dev enp0s8 proto kernel scope link src 40.1.1.3
192.168.1.1 via 10.0.2.2 dev enp0s3 proto dhcp src 10.0.2.15 metric 100
```

Gateway: 2 adapters, one for connecting to client and the other for the server.

```
  GNU nano 6.2                          /etc/netplan/02-netcfg.yaml
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s8:
      addresses: [40.1.1.2/24]
      dhcp4: no
    enp0s9:
      addresses: [20.1.1.2/24]
      dhcp4: no
```

```
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:feb4:d3b3  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:b4:d3:b3  txqueuelen 1000  (Ethernet)
        RX packets 11  bytes 2576 (2.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 22  bytes 2246 (2.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 40.1.1.2  netmask 255.255.255.0  broadcast 40.1.1.255
        inet6 fe80::a00:27ff:febc:c221  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:bc:c2:21  txqueuelen 1000  (Ethernet)
        RX packets 33  bytes 2100 (2.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 28  bytes 1886 (1.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s9: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 20.1.1.2  netmask 255.255.255.0  broadcast 20.1.1.255
        inet6 fe80::a00:27ff:fe79:6394  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:79:63:94  txqueuelen 1000  (Ethernet)
        RX packets 15  bytes 900 (900.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 38  bytes 2486 (2.4 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 106  bytes 8852 (8.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 106  bytes 8852 (8.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Q1.b)

```
abhirup@client:~$ sudo sysctl -w net.ipv4.ip_forward=1
[sudo] password for abhirup:
net.ipv4.ip_forward = 1
```

```
abhirup@client:~$ ping 40.1.1.1
PING 40.1.1.1 (40.1.1.1) 56(84) bytes of data.
64 bytes from 40.1.1.1: icmp_seq=1 ttl=63 time=1.42 ms
64 bytes from 40.1.1.1: icmp_seq=2 ttl=63 time=1.88 ms
64 bytes from 40.1.1.1: icmp_seq=3 ttl=63 time=1.34 ms
64 bytes from 40.1.1.1: icmp_seq=4 ttl=63 time=1.25 ms
64 bytes from 40.1.1.1: icmp_seq=5 ttl=63 time=1.65 ms
^C
--- 40.1.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 1.249/1.507/1.883/0.229 ms
```

Above is an example that the forwarding works. I have pinged server1 from the client.

Q2.a)

```
abhirup@gateway:~$ sudo iptables -A FORWARD -d 40.1.1.1 -p icmp --icmp-type echo-request -j ACCEPT
abhirup@gateway:~$ sudo iptables -A OUTPUT -d 40.1.1.1 -p icmp --icmp-type echo-request -j ACCEPT
abhirup@gateway:~$ sudo iptables -A FORWARD -d 40.1.1.1 -j DROP
abhirup@gateway:~$ sudo iptables -A OUTPUT -d 40.1.1.1 -j DROP
```

In the above image, the first 2 lines are to allow ping. The next 2 lines drop all other traffic.

```
abhirup@gateway:~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     icmp --  any    any     anywhere             40.1.1.1             icmp echo-r
equest
    0     0 DROP       all  --  any    any     anywhere             40.1.1.1

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     icmp --  any    any     anywhere             40.1.1.1             icmp echo-r
equest
    0     0 DROP       all  --  any    any     anywhere             40.1.1.1
```

```
abhirup@server1:~$ nc -l -p 1234
^C
```

```
abhirup@server2:~$ nc -l -p 1234
hi
bye
```

```
abhirup@client:~$ nc 40.1.1.1 1234
hu
^C
abhirup@client:~$ nc 40.1.1.3 1234
hi
bye
^C
abhirup@client:~$
```

Here, the connection to 40.1.1.1 failed since ‘hu’ didn’t get printed server side, but the one to 40.1.1.3 succeeded, indicating that the packets to 40.1.1.1 except ping, are being blocked.

```
abhirup@client:~$ ping 40.1.1.1
PING 40.1.1.1 (40.1.1.1) 56(84) bytes of data.
64 bytes from 40.1.1.1: icmp_seq=1 ttl=63 time=1.75 ms
64 bytes from 40.1.1.1: icmp_seq=2 ttl=63 time=1.76 ms
64 bytes from 40.1.1.1: icmp_seq=3 ttl=63 time=1.44 ms
64 bytes from 40.1.1.1: icmp_seq=4 ttl=63 time=1.89 ms
64 bytes from 40.1.1.1: icmp_seq=5 ttl=63 time=1.37 ms
^C
--- 40.1.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4819ms
rtt min/avg/max/mdev = 1.372/1.644/1.890/0.200 ms
```

The above image shows that ping still works, as requested by the question.

Q2.b)

```
abhirup@gateway:~$ sudo iptables -A FORWARD -s 20.1.1.1 -p TCP -j DROP
abhirup@gateway:~$ sudo iptables -A INPUT -s 20.1.1.1 -p TCP -j DROP
abhirup@gateway:~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DROP       tcp  --  any    any     20.1.1.1             anywhere

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     icmp --  any    any     anywhere             40.1.1.1             icmp echo-r
equest
    0     0 DROP       all  --  any    any     anywhere             40.1.1.1
   17  1020 DROP       tcp  --  any    any     20.1.1.1             anywhere

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     icmp --  any    any     anywhere             40.1.1.1             icmp echo-r
equest
    0     0 DROP       all  --  any    any     anywhere             40.1.1.1
```

```
abhirup@server2:~$ nc -l -p 1234
^C
abhirup@server2:~$ nc -lu -p 1234
hi
bye
^C
```

```
abhirup@client:~$ nc 40.1.1.3 1234
hi
^C
abhirup@client:~$ nc -u 40.1.1.3 1234
hi
bye
^C
```

Above, when we don't add the -u flag for udp connection, 'hi' doesn't get printed, indicating that tcp traffic is blocked. But on addition of -u flag, the text gets printed, so UDP does work when initiated from 20.1.1.1/24.

Q3.a)

TCP:

```
abhirup@server2:~$ iperf -s -t 30
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size:  128 KByte (default)
------------------------------------------------------------
abhirup@server2:~$ _
```

```
abhirup@client:~$ iperf -c 40.1.1.3
^C^Cabhirup@client:~$
```

No connection made, as expected since all TCP traffic was blocked by the netfilter in 2.b). So, bandwidth is 0.

UDP:

```
abhirup@server2:~$ iperf -s -u
------------------------------------------------------------
Server listening on UDP port 5001
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  1] local 40.1.1.3 port 5001 connected with 20.1.1.1 port 48670
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  1] 0.0000-10.0166 sec  1.25 MBytes  1.05 Mbits/sec   0.215 ms 0/895 (0%)
```

```
abhirup@client:~$ iperf -c 40.1.1.3 -u
------------------------------------------------------------
Client connecting to 40.1.1.3, UDP port 5001
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  1] local 20.1.1.1 port 48670 connected with 40.1.1.3 port 5001
[ ID] Interval       Transfer     Bandwidth
[  1] 0.0000-10.0165 sec  1.25 MBytes  1.05 Mbits/sec
[  1] Sent 896 datagrams
[  1] Server Report:
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  1] 0.0000-10.0166 sec  1.25 MBytes  1.05 Mbits/sec   0.215 ms 0/895 (0%)
```

The bandwidth here is 1.05 Mbps.

Q3.(b) Calculating RTT after 10 ping requests are sent.

(i)

```
abhirup@client:~$ ping 40.1.1.1 -c 10
PING 40.1.1.1 (40.1.1.1) 56(84) bytes of data.
64 bytes from 40.1.1.1: icmp_seq=1 ttl=63 time=1.45 ms
64 bytes from 40.1.1.1: icmp_seq=2 ttl=63 time=1.58 ms
64 bytes from 40.1.1.1: icmp_seq=3 ttl=63 time=1.53 ms
64 bytes from 40.1.1.1: icmp_seq=4 ttl=63 time=2.06 ms
64 bytes from 40.1.1.1: icmp_seq=5 ttl=63 time=1.81 ms
64 bytes from 40.1.1.1: icmp_seq=6 ttl=63 time=1.64 ms
64 bytes from 40.1.1.1: icmp_seq=7 ttl=63 time=1.69 ms
64 bytes from 40.1.1.1: icmp_seq=8 ttl=63 time=1.62 ms
64 bytes from 40.1.1.1: icmp_seq=9 ttl=63 time=2.05 ms
64 bytes from 40.1.1.1: icmp_seq=10 ttl=63 time=1.81 ms

--- 40.1.1.1 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 10275ms
rtt min/avg/max/mdev = 1.451/1.724/2.060/0.195 ms
```

Min RTT → 1.451 ms
Avg RTT → 1.724 ms
Max RTT → 2.060 ms

(ii)

```
abhirup@client:~$ ping 40.1.1.3 -c 10
PING 40.1.1.3 (40.1.1.3) 56(84) bytes of data.
64 bytes from 40.1.1.3: icmp_seq=1 ttl=63 time=2.12 ms
64 bytes from 40.1.1.3: icmp_seq=2 ttl=63 time=1.99 ms
64 bytes from 40.1.1.3: icmp_seq=3 ttl=63 time=1.78 ms
64 bytes from 40.1.1.3: icmp_seq=4 ttl=63 time=2.24 ms
64 bytes from 40.1.1.3: icmp_seq=5 ttl=63 time=3.03 ms
64 bytes from 40.1.1.3: icmp_seq=6 ttl=63 time=2.21 ms
64 bytes from 40.1.1.3: icmp_seq=7 ttl=63 time=2.39 ms
64 bytes from 40.1.1.3: icmp_seq=8 ttl=63 time=1.31 ms
64 bytes from 40.1.1.3: icmp_seq=9 ttl=63 time=2.12 ms
64 bytes from 40.1.1.3: icmp_seq=10 ttl=63 time=1.98 ms

--- 40.1.1.3 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 15944ms
rtt min/avg/max/mdev = 1.309/2.116/3.025/0.415 ms
```

Min RTT → 1.309 ms
Avg RTT → 2.116 ms
Max RTT → 3.025 ms

(iii) On average, the RTT was quite a bit longer to send a packet to 40.1.1.3 as compared to 40.1.1.1. This could be because all other traffic has been blocked to 40.1.1.1, so the ICMP messages take much less time to go through, as compared to 40.1.1.3.

Q4.a)

```
abhirup@gateway:~$ sudo iptables -t nat -A POSTROUTING -s 20.1.1.1 -j SNAT --to-source 40.1.1.2
abhirup@gateway:~$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
SNAT       all  --  20.1.1.1             anywhere             to:40.1.1.2
```

b)

```
abhirup@gateway:~$ sudo iptables -t nat -A PREROUTING -d 40.1.1.2 -j DNAT --to-destination 20.1.1.1
abhirup@gateway:~$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
DNAT       all  --  anywhere             gateway              to:20.1.1.1

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
SNAT       all  --  20.1.1.1             anywhere             to:40.1.1.2
```

c)

```
abhirup@client:~$ nc -l -p 1234
hey from client
reply from server2
```

```
abhirup@server2:~$ nc 40.1.1.2 1234
hey from client
reply from server2
```

The connection working showcases that traffic to 40.1.1.2 has successfully been routed to 20.1.1.1.

enp0s9 → 20.1.1.2

```
abhirup@gateway:~$ sudo tcpdump -i enp0s9 host 40.1.1.3
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s9, link-type EN10MB (Ethernet), snapshot length 262144 bytes
20:20:13.283784 IP 20.1.1.1.1234 > 40.1.1.3.43876: Flags [P.], seq 1651611881:1651611897, ack 245914
6655, win 510, options [nop,nop,TS val 3361088839 ecr 4156976518], length 16
20:20:13.284679 IP 40.1.1.3.43876 > 20.1.1.1.1234: Flags [.], ack 16, win 502, options [nop,nop,TS v
al 4156996821 ecr 3361088839], length 0
20:20:20.763136 IP 40.1.1.3.43876 > 20.1.1.1.1234: Flags [P.], seq 1:19, ack 16, win 502, options [n
op,nop,TS val 4157004299 ecr 3361088839], length 18
20:20:20.763650 IP 20.1.1.1.1234 > 40.1.1.3.43876: Flags [.], ack 19, win 510, options [nop,nop,TS v
al 3361096319 ecr 4157004299], length 0
```

In the 3rd line (20:20:20.763136), since the packet has gone from 40.1.1.3 to 20.1.1.1, it shows that the prerouting of all traffic with destination 40.1.1.2 to 20.1.1.1 has worked.

enp0s8 → 40.1.1.2

```
abhirup@gateway:~$ sudo tcpdump -i enp0s8 host 40.1.1.3
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s8, link-type EN10MB (Ethernet), snapshot length 262144 bytes
20:09:01.985246 IP 40.1.1.3.46838 > gateway.1234: Flags [S], seq 3442986290, win 64240, options [mss
 1460,sackOK,TS val 4156325509 ecr 0,nop,wscale 7], length 0
20:09:01.985921 IP gateway.1234 > 40.1.1.3.46838: Flags [S.], seq 2941575012, ack 3442986291, win 65
160, options [mss 1460,sackOK,TS val 3360417552 ecr 4156325509,nop,wscale 7], length 0
20:09:01.986781 IP 40.1.1.3.46838 > gateway.1234: Flags [.], ack 1, win 502, options [nop,nop,TS val
 4156325510 ecr 3360417552], length 0
20:09:07.089438 ARP, Request who-has 40.1.1.3 tell gateway, length 28
20:09:07.090148 ARP, Reply 40.1.1.3 is-at 08:00:27:a9:af:82 (oui Unknown), length 46
20:09:07.148255 ARP, Request who-has gateway tell 40.1.1.3, length 46
20:09:07.148271 ARP, Reply gateway is-at 08:00:27:bc:c2:21 (oui Unknown), length 28
20:09:07.646589 IP gateway.1234 > 40.1.1.3.46838: Flags [P.], seq 1:17, ack 1, win 510, options [nop
,nop,TS val 3360423212 ecr 4156325510], length 16
20:09:07.647818 IP 40.1.1.3.46838 > gateway.1234: Flags [.], ack 17, win 502, options [nop,nop,TS va
l 4156331171 ecr 3360423212], length 0
20:09:15.406705 IP 40.1.1.3.46838 > gateway.1234: Flags [P.], seq 1:20, ack 17, win 502, options [no
p,nop,TS val 4156338931 ecr 3360423212], length 19
20:09:15.407713 IP gateway.1234 > 40.1.1.3.46838: Flags [.], ack 20, win 510, options [nop,nop,TS va
l 3360430973 ecr 4156338931], length 0
```

In the 2nd line (20:09:01.985921), since the source is gateway(40.1.1.2), the postrouting has worked to change the source IP to 40.1.1.2 from 20.1.1.1 for any traffic coming from 20.1.1.1.

Q5.a) From 3.b), we see that the avg RTT to 40.1.1.1 is 1.724 ms while the avg RTT to 40.1.1.3 is 2.116 ms. So, we allocate a probability of 0.8 to assigning packet to 40.1.1.1, and 0.2 to 40.1.1.3.

```
abhirup@gateway:~$ sudo iptables -t nat -A PREROUTING -d 20.1.1.2 -m statistic --mode random --proba
bility 0.8 -j DNAT --to-destination 40.1.1.1
abhirup@gateway:~$ sudo iptables -t nat -A PREROUTING -d 20.1.1.2 -j DNAT --to-destination 40.1.1.3
abhirup@gateway:~$ sudo iptables -t nat -A POSTROUTING -d 40.1.1.1 -j SNAT --to-source 40.1.1.2
abhirup@gateway:~$ sudo iptables -t nat -A POSTROUTING -d 40.1.1.3 -j SNAT --to-source 40.1.1.2
abhirup@gateway:~$ sudo iptables -t nat -:
iptables v1.8.7 (nf_tables): unknown option "-:"
Try `iptables -h' or 'iptables --help' for more information.
abhirup@gateway:~$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination
DNAT       all  --  anywhere             gateway              statistic mode random probability 0.79
999999981 to:40.1.1.1
DNAT       all  --  anywhere             gateway              to:40.1.1.3

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
SNAT       all  --  anywhere             40.1.1.1             to:40.1.1.2
SNAT       all  --  anywhere             40.1.1.3             to:40.1.1.2
```

Above are the iptable rules for balancing traffic between 40.1.1.1/24 and 40.1.1.3/24.

Q5.b)

```
abhirup@client:~$ ping 20.1.1.2 -c 1
PING 20.1.1.2 (20.1.1.2) 56(84) bytes of data.
64 bytes from 20.1.1.2: icmp_seq=1 ttl=63 time=1.58 ms

--- 20.1.1.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.581/1.581/1.581/0.000 ms
abhirup@client:~$ ping 20.1.1.2 -c 1
PING 20.1.1.2 (20.1.1.2) 56(84) bytes of data.
64 bytes from 20.1.1.2: icmp_seq=1 ttl=63 time=1.84 ms

--- 20.1.1.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.839/1.839/1.839/0.000 ms
abhirup@client:~$ ping 20.1.1.2 -c 1
PING 20.1.1.2 (20.1.1.2) 56(84) bytes of data.
64 bytes from 20.1.1.2: icmp_seq=1 ttl=63 time=1.68 ms

--- 20.1.1.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.684/1.684/1.684/0.000 ms
abhirup@client:~$ ping 20.1.1.2 -c 1
PING 20.1.1.2 (20.1.1.2) 56(84) bytes of data.
64 bytes from 20.1.1.2: icmp_seq=1 ttl=63 time=1.42 ms

--- 20.1.1.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.423/1.423/1.423/0.000 ms
abhirup@client:~$ ping 20.1.1.2 -c 1
PING 20.1.1.2 (20.1.1.2) 56(84) bytes of data.
64 bytes from 20.1.1.2: icmp_seq=1 ttl=63 time=1.31 ms

--- 20.1.1.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 1.312/1.312/1.312/0.000 ms
```

I sent 5 ping requests to the gateway, 20.1.1.2, to see the distribution of where the ICMP packets went.

Output

Server1:

```
abhirup@server1:~$ sudo tshark -i enp0s8 -f 'icmp'
Running as user "root" and group "root". This could be dangerous.
Capturing on 'enp0s8'
 ** (tshark:2454) 21:49:00.308728 [Main MESSAGE] -- Capture started.
 ** (tshark:2454) 21:49:00.308829 [Main MESSAGE] -- File: "/tmp/wireshark_enp0s87822V2.pcapng"
    1 0.000000000     40.1.1.2 → 40.1.1.1     ICMP 98 Echo (ping) request  id=0x0040, seq=1/256, ttl
=63
    2 0.000027552     40.1.1.1 → 40.1.1.2     ICMP 98 Echo (ping) reply    id=0x0040, seq=1/256, ttl
=64 (request in 1)
    3 6.374651539     40.1.1.2 → 40.1.1.1     ICMP 98 Echo (ping) request  id=0x0041, seq=1/256, ttl
=63
    4 6.374677809     40.1.1.1 → 40.1.1.2     ICMP 98 Echo (ping) reply    id=0x0041, seq=1/256, ttl
=64 (request in 3)
    5 11.730989223     40.1.1.2 → 40.1.1.1     ICMP 98 Echo (ping) request  id=0x0042, seq=1/256, tt
l=63
    6 11.731016715     40.1.1.1 → 40.1.1.2     ICMP 98 Echo (ping) reply    id=0x0042, seq=1/256, tt
l=64 (request in 5)
    7 16.326955287     40.1.1.2 → 40.1.1.1     ICMP 98 Echo (ping) request  id=0x0043, seq=1/256, tt
l=63
    8 16.326979493     40.1.1.1 → 40.1.1.2     ICMP 98 Echo (ping) reply    id=0x0043, seq=1/256, tt
l=64 (request in 7)
```

Server2:

```
abhirup@server2:~$ sudo tshark -i enp0s8 -f 'icmp'
Running as user "root" and group "root". This could be dangerous.
Capturing on 'enp0s8'
 ** (tshark:3168) 21:48:56.962780 [Main MESSAGE] -- Capture started.
 ** (tshark:3168) 21:48:56.962881 [Main MESSAGE] -- File: "/tmp/wireshark_enp0s83CSUV2.pcapng"
    1 0.000000000     40.1.1.2 → 40.1.1.3     ICMP 98 Echo (ping) request  id=0x003f, seq=1/256, ttl
=63
    2 0.000032141     40.1.1.3 → 40.1.1.2     ICMP 98 Echo (ping) reply    id=0x003f, seq=1/256, ttl
=64 (request in 1)
```

4/5 packets went to server 1(40.1.1.1) and 1 went to server 2(40.1.1.3). This is exactly 80% of the time to server 1 which is the ratio we wanted previously (those this is an ideal scenario, and not something you will see all the time).