

UDP, TCP and ICMP Pingers

Pingers

Part 1: UDP Pinger

Firstly, to start the UDP Pinger server run the command “python3 UDPPingerServer.py”. Then run the UDP Pinger client that can talk to this server. After sending the request client waits for 1 timeout for receiving the response from the server and if the response is not received from the server then the TimeoutError exception is raised. This exception was handled by waiting for one more timeout to receive any ICMP Error message packets from the server. If no ICMP error packet is received from the server then the message request timed out is printed. If any ICMP packet with error is received from the server then a corresponding error message is printed. If the client is interrupted by sending either SIGINT or SIGTERM in between sending the requests then the pings are stopped and the summary is printed based on RTT data collected so far. The simulation of packet dropping is done at the application layer in the server while packets are dropped using “iptables” ; one can use the “UDPPingerModifiedServer.py” program for the server much about how to do it .

In the basic case of UDPPingerClient there would be no Interrupts, terminations and ICMP replies. The server would reply and the response is captured or dropped. Following image shows the client output in this case: This image also shows the case when requests are timed out(requests 1,2)

```
edComputerNetworks/Assignment2/PART1$ sudo python3 UDPPingerClient.py
ICMP socket opened to listen for ICMP messages
Enter number of pings to be sent:
5
send to 127.0.0.1:11000: ping 1 1725933169.813111
Request timed out for the packet 1
send to 127.0.0.1:11000: ping 2 1725933171.8188567
Request timed out for the packet 2
send to 127.0.0.1:11000: ping 3 1725933173.828995
received from 127.0.0.1:11000 PING 3 1725933173.828995 rtt=0.502ms
send to 127.0.0.1:11000: ping 4 1725933173.829579
received from 127.0.0.1:11000 PING 4 1725933173.829579 rtt=0.457ms
send to 127.0.0.1:11000: ping 5 1725933173.830109
received from 127.0.0.1:11000 PING 5 1725933173.830109 rtt=0.337ms
=====
Summary:
=====
Packets Sent: 5
Packets Received: 3
MIN RTT: 0.337ms
MAX RTT: 0.502ms
AVG RTT: 0.432ms
Loss Rate: 40.000%
=====
```

Server Output: Server had dropped the responses for 1 and 2 because of which Request timed out is seen at the client as there is no response from the server.

```
cedComputerNetworks/Assignment2/PART1$ python3 UDPPingerServer.py
Server bound to port 11000
received from 127.0.0.1:31000: ping 1 1725933169.813111
dropped message: PING 1 1725933169.813111
received from 127.0.0.1:31000: ping 2 1725933171.8188567
dropped message: PING 2 1725933171.8188567
received from 127.0.0.1:31000: ping 3 1725933173.828995
sent to 127.0.0.1:31000: PING 3 1725933173.828995
received from 127.0.0.1:31000: ping 4 1725933173.829579
sent to 127.0.0.1:31000: PING 4 1725933173.829579
received from 127.0.0.1:31000: ping 5 1725933173.830109
sent to 127.0.0.1:31000: PING 5 1725933173.830109
□
```

Following is the output when an interrupt(KeyboardInterrupt or SIGINT) event is received by the client while sending pings to the server.

```
edComputerNetworks/Assignment2/PART1$ sudo python3 UDPPingerClient.py
ICMP socket opened to listen for ICMP messages
Enter number of pings to be sent:
10
send to 127.0.0.1:11000: ping 1 1725933300.748046
received from 127.0.0.1:11000 PING 1 1725933300.748046 rtt=0.241ms
send to 127.0.0.1:11000: ping 2 1725933300.748306
received from 127.0.0.1:11000 PING 2 1725933300.748306 rtt=0.093ms
send to 127.0.0.1:11000: ping 3 1725933300.7484105
received from 127.0.0.1:11000 PING 3 1725933300.7484105 rtt=0.089ms
send to 127.0.0.1:11000: ping 4 1725933300.74851
^CReceived interrupt stopping pings
=====
Summary:
=====
Packets Sent: 3
Packets Received: 3
MIN RTT: 0.089ms
MAX RTT: 0.241ms
AVG RTT: 0.141ms
Loss Rate: 0.000%
=====
Closing icmp socket
Closing UDP socket
```

Following is the output when SIGTERM(using kill) is received by the client while sending the pings to the server:

```
edComputerNetworks/Assignment2/PART1$ sudo python3 UDPPingerClient.py
ICMP socket opened to listen for ICMP messages
Enter number of pings to be sent:
10
send to 127.0.0.1:11000: ping 1 1725933699.0687962
received from 127.0.0.1:11000 PING 1 1725933699.0687962 rtt=0.576ms
send to 127.0.0.1:11000: ping 2 1725933699.0694394
received from 127.0.0.1:11000 PING 2 1725933699.0694394 rtt=0.378ms
send to 127.0.0.1:11000: ping 3 1725933699.069879
received from 127.0.0.1:11000 PING 3 1725933699.069879 rtt=0.407ms
send to 127.0.0.1:11000: ping 4 1725933699.0703487
received from 127.0.0.1:11000 PING 4 1725933699.0703487 rtt=0.290ms
send to 127.0.0.1:11000: ping 5 1725933699.0707233
received from 127.0.0.1:11000 PING 5 1725933699.0707233 rtt=0.320ms
send to 127.0.0.1:11000: ping 6 1725933699.071107
received from 127.0.0.1:11000 PING 6 1725933699.071107 rtt=0.245ms
send to 127.0.0.1:11000: ping 7 1725933699.0714076
received from 127.0.0.1:11000 PING 7 1725933699.0714076 rtt=0.184ms
send to 127.0.0.1:11000: ping 8 1725933699.0716417
Received SIGTERM stopping pings
=====
Summary:
=====
Packets Sent: 8
Packets Received: 7
MIN RTT: 0.184ms
MAX RTT: 0.576ms
AVG RTT: 0.343ms
Loss Rate: 12.500%
=====
Closing icmp socket
Closing UDP socket
akhil@pop-os: /media/akhil/Images/backup_laptop/Documents/IITH/Department
```

When ICMP errors are received from the server for requests sent by the client. To

```
T1$ sudo python3 UDPPingerClient.py
ICMP socket opened to listen for ICMP messages
Enter number of pings to be sent:
1
send to 127.0.0.1:11000: ping 1 1725933815.5841548
Port Unreachable for the packet 1
=====
Summary:
=====
No response received from server
Packets Sent: 1
Packets Received: 0
MIN RTT: NA
MAX RTT: NA
AVG RTT: NA
Loss Rate: 100.000%
=====
Closing icmp socket
Closing UDP socket
akhil@pop-os: /media/akhil/Images/backup_laptop/Document
```

```
T1$ sudo python3 UDPPingerClient.py
ICMP socket opened to listen for ICMP messages
Enter number of pings to be sent:
1
send to 127.0.0.1:11000: ping 1 1725933869.457132
Host Unreachable for the packet 1
=====
Summary:
=====
No response received from server
Packets Sent: 1
Packets Received: 0
MIN RTT: NA
MAX RTT: NA
AVG RTT: NA
Loss Rate: 100.000%
=====
Closing icmp socket
Closing UDP socket
akhil@pop-os: /media/akhil/Images/backup_laptop/Document
```

```
T1$ sudo python3 UDPPingerClient.py
ICMP socket opened to listen for ICMP messages
Enter number of pings to be sent:
1
send to 127.0.0.1:11000: ping 1 1725933918.6100605
Net Unreachable for the packet 1
=====
Summary:
=====
No response received from server
Packets Sent: 1
Packets Received: 0
MIN RTT: NA
MAX RTT: NA
AVG RTT: NA
Loss Rate: 100.000%
=====
Closing icmp socket
Closing UDP socket
akhil@pop-os: /media/akhil/Images/backup_laptop/Document
```

```
T1$ sudo python3 UDPPingerClient.py
ICMP socket opened to listen for ICMP messages
Enter number of pings to be sent:
1
send to 127.0.0.1:11000: ping 1 1725933952.5002577
Protocol Unreachable for the packet 1
=====
Summary:
=====
No response received from server
Packets Sent: 1
Packets Received: 0
MIN RTT: NA
MAX RTT: NA
AVG RTT: NA
Loss Rate: 100.000%
=====
Closing icmp socket
Closing UDP socket
akhil@pop-os: /media/akhil/Images/backup_laptop/Document
```

simulate these ICMP error messages I have used the command “sudo iptables -A INPUT -s <source_ip_address> -p udp --ports <ports> -j REJECT --reject-with <error_response>”. Following are the outputs for four cases when “reject-with” option was set to icmp-port-unreachable, icmp-host-unreachable, icmp-net-unreachable, and icmp-proto-unreachable. The error handling is not limited to these cases. I have followed

[RFC 792](#) and also the [IANA specification](#) for ICMP to handle all the error cases present in it. More details which error messages are handled is presented in the PART 3

When multiple clients simultaneously ping the UDP server most of the clients receive the response from the server if the server has enough buffer to store the received packets. This is because the data received by the server socket is queued in its receive buffer and requests are handled one by one. But if there are enough connections to fill up this receive buffer then the new packets that come to the server socket are dropped. As there is no reliability in UDP these requests would be lost forever. As the packet is not received by the server no response will be received by the client and request would be timed out. I have tried with 3 clients and most of the time the requests were handled. The request timed out messages in the output were because of the random dropping of packets. For clarity I have provided client wise output logs in the following images. I have also included messages received on the server. At the server I observed that the messages are being received in an interleaved fashion from all three clients. To improve this when one has longer tasks, larger application payloads, and multiple clients try pinging I have used multithreading reducing the chances of the socket buffer being full.

```
send to 127.0.0.1:11000: ping 43 1725934343.1870005
received from 127.0.0.1:11000 PING 43 1725934343.1870005 rtt=0.413ms
send to 127.0.0.1:11000: ping 44 1725934343.1874828
received from 127.0.0.1:11000 PING 44 1725934343.1874828 rtt=0.387ms
send to 127.0.0.1:11000: ping 45 1725934343.18794
received from 127.0.0.1:11000 PING 45 1725934343.18794 rtt=0.304ms
send to 127.0.0.1:11000: ping 46 1725934343.1883311
Request timed out for the packet 46
send to 127.0.0.1:11000: ping 47 1725934345.1979249
received from 127.0.0.1:11000 PING 47 1725934345.1979249 rtt=0.447ms
send to 127.0.0.1:11000: ping 48 1725934345.1984632
received from 127.0.0.1:11000 PING 48 1725934345.1984632 rtt=0.402ms
send to 127.0.0.1:11000: ping 49 1725934345.1989267
received from 127.0.0.1:11000 PING 49 1725934345.1989267 rtt=0.418ms
send to 127.0.0.1:11000: ping 50 1725934345.1994145
received from 127.0.0.1:11000 PING 50 1725934345.1994145 rtt=0.406ms
send to 127.0.0.1:11000: ping 51 1725934345.1998835
received from 127.0.0.1:11000 PING 51 1725934345.1998835 rtt=0.373ms
send to 127.0.0.1:11000: ping 52 1725934345.2003188
Request timed out for the packet 52
send to 127.0.0.1:11000: ping 53 1725934347.2098572
received from 127.0.0.1:11000 PING 53 1725934347.2098572 rtt=0.224ms
send to 127.0.0.1:11000: ping 54 1725934347.2101061
Request timed out for the packet 54
send to 127.0.0.1:11000: ping 55 1725934349.219887
Request timed out for the packet 55
send to 127.0.0.1:11000: ping 56 1725934351.229878
received from 127.0.0.1:11000 PING 56 1725934351.229878 rtt=0.479ms
send to 127.0.0.1:11000: ping 57 1725934351.2304564
received from 127.0.0.1:11000 PING 57 1725934351.2304564 rtt=0.480ms
send to 127.0.0.1:11000: ping 58 1725934351.2310302
Request timed out for the packet 58
send to 127.0.0.1:11000: ping 59 1725934353.2409072
Request timed out for the packet 59
send to 127.0.0.1:11000: ping 60 1725934355.250862
```

```
send to 127.0.0.1:11000: ping 71 1725934347.831084
received from 127.0.0.1:11000 PING 71 1725934347.831084 rtt=0.483ms
send to 127.0.0.1:11000: ping 72 1725934347.831646
Request timed out for the packet 72
send to 127.0.0.1:11000: ping 73 1725934349.842265
received from 127.0.0.1:11000 PING 73 1725934349.842265 rtt=0.434ms
send to 127.0.0.1:11000: ping 74 1725934349.842766
received from 127.0.0.1:11000 PING 74 1725934349.842766 rtt=0.382ms
send to 127.0.0.1:11000: ping 75 1725934349.8432133
received from 127.0.0.1:11000 PING 75 1725934349.8432133 rtt=0.360ms
send to 127.0.0.1:11000: ping 76 1725934349.8436334
received from 127.0.0.1:11000 PING 76 1725934349.8436334 rtt=0.383ms
send to 127.0.0.1:11000: ping 77 1725934349.8440766
received from 127.0.0.1:11000 PING 77 1725934349.8440766 rtt=0.367ms
send to 127.0.0.1:11000: ping 78 1725934349.8445022
received from 127.0.0.1:11000 PING 78 1725934349.8445022 rtt=0.409ms
send to 127.0.0.1:11000: ping 79 1725934349.844976
received from 127.0.0.1:11000 PING 79 1725934349.844976 rtt=0.353ms
send to 127.0.0.1:11000: ping 80 1725934349.8453727
Request timed out for the packet 80
send to 127.0.0.1:11000: ping 81 1725934351.8558497
received from 127.0.0.1:11000 PING 81 1725934351.8558497 rtt=0.286ms
send to 127.0.0.1:11000: ping 82 1725934351.856177
Request timed out for the packet 82
send to 127.0.0.1:11000: ping 83 1725934353.8643801
received from 127.0.0.1:11000 PING 83 1725934353.8643801 rtt=0.341ms
send to 127.0.0.1:11000: ping 84 1725934353.8647747
received from 127.0.0.1:11000 PING 84 1725934353.8647747 rtt=0.286ms
send to 127.0.0.1:11000: ping 85 1725934353.8651128
received from 127.0.0.1:11000 PING 85 1725934353.8651128 rtt=0.340ms
send to 127.0.0.1:11000: ping 86 1725934353.865494
received from 127.0.0.1:11000 PING 86 1725934353.865494 rtt=0.225ms
send to 127.0.0.1:11000: ping 87 1725934353.8657522
received from 127.0.0.1:11000 PING 87 1725934353.8657522 rtt=0.236ms
send to 127.0.0.1:11000: ping 88 1725934353.8660212
```

```

send to 127.0.0.1:11000: ping 43 1725934346.0090373
received from 127.0.0.1:11000 PING 43 1725934346.0090373 rtt=0.098ms
send to 127.0.0.1:11000: ping 44 1725934346.0091476
received from 127.0.0.1:11000 PING 44 1725934346.0091476 rtt=0.046ms
send to 127.0.0.1:11000: ping 45 1725934346.009207
Request timed out for the packet 45
send to 127.0.0.1:11000: ping 46 1725934348.018928
received from 127.0.0.1:11000 PING 46 1725934348.018928 rtt=0.198ms
send to 127.0.0.1:11000: ping 47 1725934348.0191488
received from 127.0.0.1:11000 PING 47 1725934348.0191488 rtt=0.165ms
send to 127.0.0.1:11000: ping 48 1725934348.0193324
received from 127.0.0.1:11000 PING 48 1725934348.0193324 rtt=0.125ms
send to 127.0.0.1:11000: ping 49 1725934348.019475
received from 127.0.0.1:11000 PING 49 1725934348.019475 rtt=0.119ms
send to 127.0.0.1:11000: ping 50 1725934348.0196123
Request timed out for the packet 50
send to 127.0.0.1:11000: ping 51 1725934350.0300188
received from 127.0.0.1:11000 PING 51 1725934350.0300188 rtt=0.436ms
send to 127.0.0.1:11000: ping 52 1725934350.0305233
received from 127.0.0.1:11000 PING 52 1725934350.0305233 rtt=0.418ms
send to 127.0.0.1:11000: ping 53 1725934350.0310094
received from 127.0.0.1:11000 PING 53 1725934350.0310094 rtt=0.390ms
send to 127.0.0.1:11000: ping 54 1725934350.0314612
Request timed out for the packet 54
send to 127.0.0.1:11000: ping 55 1725934352.0419388
received from 127.0.0.1:11000 PING 55 1725934352.0419388 rtt=0.445ms
send to 127.0.0.1:11000: ping 56 1725934352.0424547
received from 127.0.0.1:11000 PING 56 1725934352.0424547 rtt=0.423ms
send to 127.0.0.1:11000: ping 57 1725934352.0429459
Request timed out for the packet 57
send to 127.0.0.1:11000: ping 58 1725934354.0528708
received from 127.0.0.1:11000 PING 58 1725934354.0528708 rtt=0.436ms
send to 127.0.0.1:11000: ping 59 1725934354.0533743
received from 127.0.0.1:11000 PING 59 1725934354.0533743 rtt=0.348ms
send to 127.0.0.1:11000: ping 60 1725934354.0537727

```

```

received from 127.0.0.1:31002: ping 87 1725934363.1072223
sent to 127.0.0.1:31002: PING 87 1725934363.1072223
received from 127.0.0.1:31002: ping 88 1725934363.1076462
dropped message: PING 88 1725934363.1076462
received from 127.0.0.1:31001: ping 97 1725934364.4355183
dropped message: PING 97 1725934364.4355183
received from 127.0.0.1:31000: ping 117 1725934364.9208486
sent to 127.0.0.1:31000: PING 117 1725934364.9208486
received from 127.0.0.1:31000: ping 118 1725934364.9211442
sent to 127.0.0.1:31000: PING 118 1725934364.9211442
received from 127.0.0.1:31000: ping 119 1725934364.9213886
sent to 127.0.0.1:31000: PING 119 1725934364.9213886
received from 127.0.0.1:31000: ping 120 1725934364.9216042
sent to 127.0.0.1:31000: PING 120 1725934364.9216042
received from 127.0.0.1:31000: ping 121 1725934364.9218311
sent to 127.0.0.1:31000: PING 121 1725934364.9218311
received from 127.0.0.1:31000: ping 122 1725934364.9220562
sent to 127.0.0.1:31000: PING 122 1725934364.9220562
received from 127.0.0.1:31000: ping 123 1725934364.9222312
sent to 127.0.0.1:31000: PING 123 1725934364.9222312
received from 127.0.0.1:31000: ping 124 1725934364.9223256
sent to 127.0.0.1:31000: PING 124 1725934364.9223256
received from 127.0.0.1:31000: ping 125 1725934364.9224112
sent to 127.0.0.1:31000: PING 125 1725934364.9224112
received from 127.0.0.1:31000: ping 126 1725934364.922496
dropped message: PING 126 1725934364.922496
received from 127.0.0.1:31002: ping 89 1725934365.1178453
sent to 127.0.0.1:31002: PING 89 1725934365.1178453
received from 127.0.0.1:31002: ping 90 1725934365.1182353
sent to 127.0.0.1:31002: PING 90 1725934365.1182353
received from 127.0.0.1:31002: ping 91 1725934365.1185226

```

To run this server on two different machines firstly the firewall should be disabled. To disable the firewall one can use the command “sudo ufw disable” this command works for ubuntu based operating systems and one can also allow the requests to the server port if you don’t want to disable firewall. And then one has to manually set the destination IP address in the client as the address of the UDP server. I have also modified the server code by removing the packet drop simulation that used randint() and replacing it with tc(traffic controller) as asked in the question. The command “sudo tc qdisc add dev <interface> root netem loss <percentage>” to simulate packet dropping using tc at NIC itself. I have modified the code accordingly based on this and updated the code in the UDPPingerModifiedServer.py file. With multi threading as the payload size and processing delays were less in this case it wouldn’t have a larger impact. As said earlier multithreading would be helpful if one has longer tasks, multiple clients, and larger payloads in case of UDP.

Following is the output on the client with the first case where I have simulated the packet drop with randint(): (Here client and server are on different machines)

```
root@faa520359b4a:/workdir/PART1# python3 UDPPingerClient.py
ICMP socket opened to listen for ICMP messages
Enter number of pings to be sent:
5
send to 172.17.0.1:11000: ping 1 1725935406.6360633
received from 172.17.0.1:11000 PING 1 1725935406.6360633 rtt=0.180ms
send to 172.17.0.1:11000: ping 2 1725935406.6362703
Request timed out for the packet 2
send to 172.17.0.1:11000: ping 3 1725935408.64591
Request timed out for the packet 3
send to 172.17.0.1:11000: ping 4 1725935410.6561027
received from 172.17.0.1:11000 PING 4 1725935410.6561027 rtt=0.675ms
send to 172.17.0.1:11000: ping 5 1725935410.656907
received from 172.17.0.1:11000 PING 5 1725935410.656907 rtt=0.538ms
=====
Summary:
=====
Packets Sent: 5
Packets Received: 3
MIN RTT: 0.180ms
MAX RTT: 0.675ms
AVG RTT: 0.464ms
Loss Rate: 40.000%
=====
Closing icmp socket
Closing UDP socket
root@faa520359b4a:/workdir/PART1#
```

Following is the output on the server with the same:

```
rNetworks/Assignment2/PART1$ python3 UDPPingerServer.py
Server bound to port 11000
received from 172.17.0.4:31000: ping 1 1725935406.6360633
sent to 172.17.0.4:31000: PING 1 1725935406.6360633
received from 172.17.0.4:31000: ping 2 1725935406.6362703
dropped message: PING 2 1725935406.6362703
received from 172.17.0.4:31000: ping 3 1725935408.64591
dropped message: PING 3 1725935408.64591
received from 172.17.0.4:31000: ping 4 1725935410.6561027
sent to 172.17.0.4:31000: PING 4 1725935410.6561027
received from 172.17.0.4:31000: ping 5 1725935410.656907
sent to 172.17.0.4:31000: PING 5 1725935410.656907
```


Following is output on the client after modifications:(Here client and server are on different machines)

```
root@faa520359b4a:/workdir/PART1# python3 UDPPingerClient.py
ICMP socket opened to listen for ICMP messages
Enter number of pings to be sent:
5
send to 172.17.0.1:11000: ping 1 1725935610.1637425
received from 172.17.0.1:11000 PING 1 1725935610.1637425 rtt=0.645ms
send to 172.17.0.1:11000: ping 2 1725935610.1645327
received from 172.17.0.1:11000 PING 2 1725935610.1645327 rtt=0.534ms
send to 172.17.0.1:11000: ping 3 1725935610.1651456
Request timed out for the packet 3
send to 172.17.0.1:11000: ping 4 1725935612.1748595
Request timed out for the packet 4
send to 172.17.0.1:11000: ping 5 1725935614.1848583
received from 172.17.0.1:11000 PING 5 1725935614.1848583 rtt=0.218ms
=====
Summary:
=====
Packets Sent: 5
Packets Received: 3
MIN RTT: 0.218ms
MAX RTT: 0.645ms
AVG RTT: 0.466ms
Loss Rate: 40.000%
=====
Closing icmp socket
Closing UDP socket
```

Following is output on the server here even though the server sent the packet one can see that the client has not received it and the wait for receive timed out this packet might have been dropped by tc:

```
rNetworks/Assignment2/PART1$ python3 UDPPingerModifiedServer.py
Server bound to port 11000
received from 172.17.0.4:31000: ping 1 1725935610.1637425
sent to 172.17.0.4:31000: PING 1 1725935610.1637425
received from 172.17.0.4:31000: ping 2 1725935610.1645327
sent to 172.17.0.4:31000: PING 2 1725935610.1645327
received from 172.17.0.4:31000: ping 3 1725935610.1651456
sent to 172.17.0.4:31000: PING 3 1725935610.1651456
received from 172.17.0.4:31000: ping 4 1725935612.1748595
sent to 172.17.0.4:31000: PING 4 1725935612.1748595
received from 172.17.0.4:31000: ping 5 1725935614.1848583
sent to 172.17.0.4:31000: PING 5 1725935614.1848583
```

Part 2: TCP Pinger

A TCP server that can only handle a single connection in the TCPingerServer.py file. In this case if the client sends an empty message to the server then the server side connection should be closed. Also ICMP raw socket is used to receive any ICMP error messages that are sent from the server. Following is the output on the client without packet loss being enabled:

```
root@faa520359b4a:/workdir/PART2# python3 TCPingerClient.py
ICMP socket opened to listen for ICMP messages
Enter number of pings to be sent:
5
Connected to 172.17.0.1:11001
send to 172.17.0.1:11001: ping 1 1725936278.48448
received from 172.17.0.1:11001 PING 1 1725936278.48448 rtt=0.408ms
send to 172.17.0.1:11001: ping 2 1725936278.4849772
received from 172.17.0.1:11001 PING 2 1725936278.4849772 rtt=0.326ms
send to 172.17.0.1:11001: ping 3 1725936278.485409
received from 172.17.0.1:11001 PING 3 1725936278.485409 rtt=0.293ms
send to 172.17.0.1:11001: ping 4 1725936278.4857693
received from 172.17.0.1:11001 PING 4 1725936278.4857693 rtt=0.247ms
send to 172.17.0.1:11001: ping 5 1725936278.4860916
received from 172.17.0.1:11001 PING 5 1725936278.4860916 rtt=0.261ms
=====
Summary:
=====
Packets Sent: 5
Packets Received: 5
MIN RTT: 0.247ms
MAX RTT: 0.408ms
AVG RTT: 0.307ms
Loss Rate: 0.000%
=====
Closing icmp socket
Closing tcp socket
```

Following is the output on the server for the same case:

```
rNetworks/Assignment2/PART2$ sudo python3 TCPingerServer.py
Listening on port 11001
Connected to 172.17.0.4:32000
received from 172.17.0.4:32000: ping 1 1725936278.48448
sent to 172.17.0.4:32000 PING 1 1725936278.48448
received from 172.17.0.4:32000: ping 2 1725936278.4849772
sent to 172.17.0.4:32000 PING 2 1725936278.4849772
received from 172.17.0.4:32000: ping 3 1725936278.485409
sent to 172.17.0.4:32000 PING 3 1725936278.485409
received from 172.17.0.4:32000: ping 4 1725936278.4857693
sent to 172.17.0.4:32000 PING 4 1725936278.4857693
received from 172.17.0.4:32000: ping 5 1725936278.4860916
sent to 172.17.0.4:32000 PING 5 1725936278.4860916
received from 172.17.0.4:32000:
```

After enabling packet loss the RTT for a few requests will increase because of the retransmissions happening in TCP. Following is output on the client:

```
root@faa520359b4a:/workdir/PART2# python3 TCPPingerClient.py
ICMP socket opened to listen for ICMP messages
Enter number of pings to be sent:
5
Connected to 172.17.0.1:11001
send to 172.17.0.1:11001: ping 1 1725936532.4052527
received from 172.17.0.1:11001 PING 1 1725936532.4052527 rtt=207.701ms
send to 172.17.0.1:11001: ping 2 1725936532.6130576
received from 172.17.0.1:11001 PING 2 1725936532.6130576 rtt=0.387ms
send to 172.17.0.1:11001: ping 3 1725936532.6135418
received from 172.17.0.1:11001 PING 3 1725936532.6135418 rtt=0.357ms
send to 172.17.0.1:11001: ping 4 1725936532.6139631
received from 172.17.0.1:11001 PING 4 1725936532.6139631 rtt=0.268ms
send to 172.17.0.1:11001: ping 5 1725936532.6143363
received from 172.17.0.1:11001 PING 5 1725936532.6143363 rtt=206.669ms
=====
Summary:
=====
Packets Sent: 5
Packets Received: 5
MIN RTT: 0.268ms
MAX RTT: 207.701ms
AVG RTT: 83.076ms
Loss Rate: 0.000%
=====
Closing icmp socket
Closing tcp socket
root@faa520359b4a:/workdir/PART2#
```

Following is the output on the server:

```
Connected to 172.17.0.4:32000
received from 172.17.0.4:32000: ping 1 1725936532.4052527
sent to 172.17.0.4:32000 PING 1 1725936532.4052527
received from 172.17.0.4:32000: ping 2 1725936532.6130576
sent to 172.17.0.4:32000 PING 2 1725936532.6130576
received from 172.17.0.4:32000: ping 3 1725936532.6135418
sent to 172.17.0.4:32000 PING 3 1725936532.6135418
received from 172.17.0.4:32000: ping 4 1725936532.6139631
sent to 172.17.0.4:32000 PING 4 1725936532.6139631
received from 172.17.0.4:32000: ping 5 1725936532.6143363
sent to 172.17.0.4:32000 PING 5 1725936532.6143363
received from 172.17.0.4:32000:
□
```

When multiple clients try connecting to this server without multithreading they aren't able to receive a response from the server. The first client connection was established and it was able to use the connection for sending and receiving information but the second

connection always printed a Request timed out message. In this case the connection was established and the client could send the data to the server but the server could only process one client and other clients requests were stored in the receive buffer of the server socket. Because of which the client always printed a request timed out message on the client console. Following is the second client output:

```
root@534b2526fe5e:/workdir/PART2# python3 TCPPingerClient.py
ICMP socket opened to listen for ICMP messages
Enter number of pings to be sent:
20
Connected to 172.17.0.1:11001
send to 172.17.0.1:11001: ping 1 1725936749.6442893
Request timed out for the packet 1
send to 172.17.0.1:11001: ping 2 1725936751.6541061
Request timed out for the packet 2
send to 172.17.0.1:11001: ping 3 1725936753.663983
Request timed out for the packet 3
send to 172.17.0.1:11001: ping 4 1725936755.6730528
Request timed out for the packet 4
send to 172.17.0.1:11001: ping 5 1725936757.6836596
Request timed out for the packet 5
send to 172.17.0.1:11001: ping 6 1725936759.6938019
Request timed out for the packet 6
send to 172.17.0.1:11001: ping 7 1725936761.7036333
Request timed out for the packet 7
send to 172.17.0.1:11001: ping 8 1725936763.7139037
Request timed out for the packet 8
send to 172.17.0.1:11001: ping 9 1725936765.7238743
Request timed out for the packet 9
send to 172.17.0.1:11001: ping 10 1725936767.7338722
Request timed out for the packet 10
```

After sometime when the first client is stopped all the messages that were buffered by the second client on the server were received in a single message.

```
sent to 172.17.0.4:32000 PING 869 1725936775.0540745
received from 172.17.0.4:32000: ping 870 1725936775.054555
sent to 172.17.0.4:32000 PING 870 1725936775.054555
received from 172.17.0.4:32000: ping 871 1725936775.05502
sent to 172.17.0.4:32000 PING 871 1725936775.05502
received from 172.17.0.4:32000: ping 872 1725936775.0554159
sent to 172.17.0.4:32000 PING 872 1725936775.0554159
received from 172.17.0.4:32000: ping 873 1725936775.470026
sent to 172.17.0.4:32000 PING 873 1725936775.470026
received from 172.17.0.4:32000:
Connected to 172.17.0.3:32000
received from 172.17.0.3:32000: ping 1 1725936749.6442893ping 2 1725936751.6541
061ping 3 1725936753.663983ping 4 1725936755.6730528ping 5 1725936757.6836596pi
ng 6 1725936759.6938019ping 7 1725936761.7036333ping 8 1725936763.7139037ping 9
1725936765.7238743ping 10 1725936767.7338722ping 11 1725936769.7444527
sent to 172.17.0.3:32000 PING 1 1725936749.6442893PING 2 1725936751.6541061PING
3 1725936753.663983PING 4 1725936755.6730528PING 5 1725936757.6836596PING 6 17
25936759.6938019PING 7 1725936761.7036333PING 8 1725936763.7139037PING 9 172593
6765.7238743PING 10 1725936767.7338722PING 11 1725936769.7444527
received from 172.17.0.3:32000:
```

To solve the above problem as given in the question, multithreading should be used. The TCP server code to work with multiple clients in the file named TCPpingModifiedServer.py. Now by running multiple clients one can see both the clients able to talk to the server simultaneously.

I tried connecting from 2 clients and the server logged the following output. The server is able to send and receive the data in interleaved fashion from both the clients. Note that the clients have different IP addresses but the same port number.

```
sent to 172.17.0.4:32000 PING 103 1725939109.479941
received from 172.17.0.3:32000: ping 103 1725939109.5169537
sent to 172.17.0.3:32000 PING 103 1725939109.5169537
received from 172.17.0.3:32000: ping 104 1725939109.5174363
sent to 172.17.0.3:32000 PING 104 1725939109.5174363
received from 172.17.0.3:32000: ping 105 1725939109.517957
sent to 172.17.0.3:32000 PING 105 1725939109.517957
received from 172.17.0.4:32000: ping 104 1725939109.6849735
sent to 172.17.0.4:32000 PING 104 1725939109.6849735
received from 172.17.0.4:32000: ping 105 1725939109.894991
sent to 172.17.0.4:32000 PING 105 1725939109.894991
received from 172.17.0.4:32000: ping 106 1725939109.8954716
sent to 172.17.0.4:32000 PING 106 1725939109.8954716
received from 172.17.0.4:32000: ping 107 1725939109.8960035
sent to 172.17.0.4:32000 PING 107 1725939109.8960035
received from 172.17.0.3:32000: ping 106 1725939109.9339595
sent to 172.17.0.3:32000 PING 106 1725939109.9339595
received from 172.17.0.3:32000: ping 107 1725939109.9343395
sent to 172.17.0.3:32000 PING 107 1725939109.9343395
received from 172.17.0.3:32000: ping 108 1725939109.934505
sent to 172.17.0.3:32000 PING 108 1725939109.934505
received from 172.17.0.3:32000: ping 109 1725939109.934725
sent to 172.17.0.3:32000 PING 109 1725939109.934725
```

Following is a test sample output from a client after modifications:

```
root@534b2526fe5e:/workdir/PART2# python3 TCPPingerClient.py
ICMP socket opened to listen for ICMP messages
Enter number of pings to be sent:
5
Connected to 172.17.0.1:11001
send to 172.17.0.1:11001: ping 1 1725939275.21882
received from 172.17.0.1:11001 PING 1 1725939275.21882 rtt=210.143ms
send to 172.17.0.1:11001: ping 2 1725939275.4291072
received from 172.17.0.1:11001 PING 2 1725939275.4291072 rtt=0.415ms
send to 172.17.0.1:11001: ping 3 1725939275.4296343
received from 172.17.0.1:11001 PING 3 1725939275.4296343 rtt=0.404ms
send to 172.17.0.1:11001: ping 4 1725939275.43013
received from 172.17.0.1:11001 PING 4 1725939275.43013 rtt=0.400ms
send to 172.17.0.1:11001: ping 5 1725939275.4306035
received from 172.17.0.1:11001 PING 5 1725939275.4306035 rtt=0.402ms
=====
Summary:
=====
Packets Sent: 5
Packets Received: 5
MIN RTT: 0.400ms
MAX RTT: 210.143ms
AVG RTT: 42.353ms
Loss Rate: 0.000%
=====
Closing icmp socket
Closing tcp socket
root@534b2526fe5e:/workdir/PART2#
```

Following images show ICMP Error handling output when the client is initially connected to the server but during communication server becomes unreachable(For more errors look into PART 3)

```
Port Unreachable for the packet 64
send to 172.17.0.1:11001: ping 65 1725939764.4531121
Port Unreachable for the packet 65
send to 172.17.0.1:11001: ping 66 1725939765.45837
Port Unreachable for the packet 66
send to 172.17.0.1:11001: ping 67 1725939766.4636536
Port Unreachable for the packet 67
send to 172.17.0.1:11001: ping 68 1725939767.4689102
Request timed out for the packet 68
send to 172.17.0.1:11001: ping 69 1725939769.479342
Port Unreachable for the packet 69
send to 172.17.0.1:11001: ping 70 1725939770.4847739
Request timed out for the packet 70
```

```
Net Unreachable for the packet 131
send to 172.17.0.1:11001: ping 132 1725940062.0480592
Net Unreachable for the packet 132
send to 172.17.0.1:11001: ping 133 1725940063.0534954
Net Unreachable for the packet 133
send to 172.17.0.1:11001: ping 134 1725940064.058929
Net Unreachable for the packet 134
send to 172.17.0.1:11001: ping 135 1725940065.0643663
Net Unreachable for the packet 135
send to 172.17.0.1:11001: ping 136 1725940066.0689073
Net Unreachable for the packet 136
send to 172.17.0.1:11001: ping 137 1725940067.0743005
Request timed out for the packet 137
```

```

Protocol Unreachable for the packet 62
send to 172.17.0.1:11001: ping 63 1725940027.1626024
Protocol Unreachable for the packet 63
send to 172.17.0.1:11001: ping 64 1725940028.1670003
Protocol Unreachable for the packet 64
send to 172.17.0.1:11001: ping 65 1725940029.1724067
Protocol Unreachable for the packet 65
send to 172.17.0.1:11001: ping 66 1725940030.177028
Protocol Unreachable for the packet 66
send to 172.17.0.1:11001: ping 67 1725940031.1820402
Protocol Unreachable for the packet 67
send to 172.17.0.1:11001: ping 68 1725940032.1870344
Request timed out for the packet 68

```

```

Host Unreachable for the packet 61
send to 172.17.0.1:11001: ping 62 1725940121.386361
Host Unreachable for the packet 62
send to 172.17.0.1:11001: ping 63 1725940122.3916125
Host Unreachable for the packet 63
send to 172.17.0.1:11001: ping 64 1725940123.3970232
Host Unreachable for the packet 64
send to 172.17.0.1:11001: ping 65 1725940124.4024417
Host Unreachable for the packet 65
send to 172.17.0.1:11001: ping 66 1725940125.4069698
Host Unreachable for the packet 66
send to 172.17.0.1:11001: ping 67 1725940126.4119365
Host Unreachable for the packet 67

```

```

send to 172.17.0.1:11001: ping 83 1725940250.2180323
Communication with Destination Network is Administratively Prohibited for the p
acket 83
send to 172.17.0.1:11001: ping 84 1725940259.2187278
Communication with Destination Network is Administratively Prohibited for the p
acket 84
send to 172.17.0.1:11001: ping 85 1725940260.2241933
Communication with Destination Network is Administratively Prohibited for the p
acket 85
send to 172.17.0.1:11001: ping 86 1725940261.2297032
Communication with Destination Network is Administratively Prohibited for the p
acket 86
send to 172.17.0.1:11001: ping 87 1725940262.2351322
Communication with Destination Network is Administratively Prohibited for the p
acket 87
send to 172.17.0.1:11001: ping 88 1725940263.240568
Communication with Destination Network is Administratively Prohibited for the p
acket 88
send to 172.17.0.1:11001: ping 89 1725940264.3661647
Request timed out for the packet 89
send to 172.17.0.1:11001: ping 90 1725940266.376759
Communication with Destination Network is Administratively Prohibited for the p
acket 90
send to 172.17.0.1:11001: ping 91 1725940268.3601367

```

```

et 25
send to 172.17.0.1:11001: ping 26 1725940349.8069384
Communication with Destination Host is Administratively Prohibited for the pack
et 26
send to 172.17.0.1:11001: ping 27 1725940350.8123684
Communication with Destination Host is Administratively Prohibited for the pack
et 27
send to 172.17.0.1:11001: ping 28 1725940351.8178084
Communication with Destination Host is Administratively Prohibited for the pack
et 28
send to 172.17.0.1:11001: ping 29 1725940352.8232553
Communication with Destination Host is Administratively Prohibited for the pack
et 29
send to 172.17.0.1:11001: ping 30 1725940353.828741
Communication with Destination Host is Administratively Prohibited for the pack
et 30
send to 172.17.0.1:11001: ping 31 1725940354.8341448
Communication with Destination Host is Administratively Prohibited for the pack
et 31
send to 172.17.0.1:11001: ping 32 1725940355.8395066
Communication with Destination Host is Administratively Prohibited for the pack
et 32
send to 172.17.0.1:11001: ping 33 1725940356.8449283
Request timed out for the packet 33

```

```

Communication Administratively Prohibited for the packet 47
send to 172.17.0.1:11001: ping 48 1725940420.3220675
Communication Administratively Prohibited for the packet 48
send to 172.17.0.1:11001: ping 49 1725940421.3274977
Communication Administratively Prohibited for the packet 49
send to 172.17.0.1:11001: ping 50 1725940422.3329465
Communication Administratively Prohibited for the packet 50
send to 172.17.0.1:11001: ping 51 1725940423.3383718
Communication Administratively Prohibited for the packet 51
send to 172.17.0.1:11001: ping 52 1725940424.3429732
Communication Administratively Prohibited for the packet 52
send to 172.17.0.1:11001: ping 53 1725940425.3483908
Communication Administratively Prohibited for the packet 53
send to 172.17.0.1:11001: ping 54 1725940426.3538768
Communication Administratively Prohibited for the packet 54
send to 172.17.0.1:11001: ping 55 1725940427.3592085

```

Part 3: ICMP Pinger

ICMP pinger implements two functions, the ping and the receiveOnePing functions that handle the ping functionality. In the receiveOnePing function it is checked if the IP address in the received ICMP message is the same as the IP address of the destination. Then the response is unpacked to get ICMP type, code, checksum, id, sequence number and timestamp. The timestamp in the request and the time at which the select unblocks is used to calculate the Round Trip Time. And it is checked if the received checksum matches the calculated checksum, and if received ID matches the current process ID. And also written a function to identify the ICMP errors based on the type and code in the ICMP header. Identifying the ping metrics NUM packets sent, NUM packets received, MIN RTT, MAX RTT, and AVG RTT the ping function was written.

Following is the output on pinging to “google.com”:

```
root@534b2526fe5e:/workdir/PART3# python3 ICMPPingerClient.py
Pinging 142.250.199.142 using Python:

Enter number of pings to be sent:
5
Request timed out for packet 1 Request timed out
36 bytes from 142.250.199.142: ttl=111 rtt=72.502ms
36 bytes from 142.250.199.142: ttl=111 rtt=49.423ms
36 bytes from 142.250.199.142: ttl=111 rtt=72.032ms
36 bytes from 142.250.199.142: ttl=111 rtt=48.158ms
=====
Summary:
=====
Packets Sent: 5
Packets Received: 4
MIN RTT: 48.158ms
MAX RTT: 72.502ms
AVG RTT: 60.529ms
Loss Rate: 20.000%
=====
root@534b2526fe5e:/workdir/PART3#
```

The Request timed out messages are because of packet loss that is simulated using TC.


Following is the output when there is no packet loss:

```
root@534b2526fe5e:/workdir/PART3# python3 ICMPPingClient.py
Pinging 142.250.199.142 using Python:

Enter number of pings to be sent:
5
36 bytes from 142.250.199.142: ttl=111 rtt=290.449ms
36 bytes from 142.250.199.142: ttl=111 rtt=62.688ms
36 bytes from 142.250.199.142: ttl=111 rtt=41.665ms
36 bytes from 142.250.199.142: ttl=111 rtt=70.074ms
36 bytes from 142.250.199.142: ttl=111 rtt=51.474ms
=====
Summary:
=====
Packets Sent: 5
Packets Received: 5
MIN RTT: 41.665ms
MAX RTT: 290.449ms
AVG RTT: 103.270ms
Loss Rate: 0.000%
=====
root@534b2526fe5e:/workdir/PART3#
```

Following are the ICMP error messages that are identified from [RFC 792](#) and [Internet Control Message Protocol \(ICMP\) Parameters](#) and implemented in all pingers:

- Type 3 - Code 0 - Net Unreachable
- Type 3 - Code 1 - Host Unreachable
- Type 3 - Code 2 - Protocol Unreachable
- Type 3 - Code 3 - Port Unreachable
- Type 3 - Code 4 - Fragmentation Needed and Don't Fragment was Set
- Type 3 - Code 5 - Source Route Failed
- Type 3 - Code 6 - Destination Network Unknown
- Type 3 - Code 7 - Destination Host Unknown
- Type 3 - Code 8 - Source Host Isolated
- Type 3 - Code 9 - Communication with Destination Network is Administratively Prohibited
- Type 3 - Code 10 - Communication with Destination Host is Administratively Prohibited
- Type 3 - Code 11 - Destination Network Unreachable for Type of Service
- Type 3 - Code 12 - Destination Host Unreachable for Type of Service
- Type 3 - Code 13 - Communication Administratively Prohibited
- Type 3 - Code 14 - Host Precedence Violation
- Type 3 - Code 15 - Precedence cutoff in effect

- 
- Type 4 - Code 0 - Source Quench
 - Type 11 - Code 0 - Time to Live exceeded in Transit
 - Type 11 - Code 1 - Fragment Reassembly Time Exceeded
 - Type 12 - Code 0 - Pointer indicates the error
 - Type 12 - Code 1 - Missing a Required Option
 - Type 12 - Code 2 - Bad Length
 - Type 40 - Code 0 - Bad SPI
 - Type 40 - Code 1 - Authentication Failed
 - Type 40 - Code 2 - Decompression Failed
 - Type 40 - Code 3 - Decryption Failed
 - Type 40 - Code 4 - Need Authentication
 - Type 40 - Code 5 - Need Authorization
 - Type 43 - Code 1 - Malformed Query
 - Type 43 - Code 2 - No Such Interface
 - Type 43 - Code 3 - No Such Table Entry
 - Type 43 - Code 4 - Multiple Interfaces Satisfy Query

Following images show few ICMP error handling cases simulated by using iptables:

```
root@534b2526fe5e:/workdir/PART3# python3 ICMPpingClient.py
Pinging 172.17.0.1 using Python:

Enter number of pings to be sent:
5
Port Unreachable for packet 1 Port Unreachable
Port Unreachable for packet 2 Port Unreachable
Port Unreachable for packet 3 Port Unreachable
Request timed out for packet 4 Request timed out
Port Unreachable for packet 5 Port Unreachable
=====
Summary:
=====
No response received from server
Packets Sent: 5
Packets Received: 0
MIN RTT: NA
MAX RTT: NA
AVG RTT: NA
Loss Rate: 100.000%
=====
root@534b2526fe5e:/workdir/PART3#
```

```
root@534b2526fe5e:/workdir/PART3# python3 ICMPpingClient.py
Pinging 172.17.0.1 using Python:

Enter number of pings to be sent:
5
Host Unreachable for packet 1 Host Unreachable
Request timed out for packet 2 Request timed out
Host Unreachable for packet 3 Host Unreachable
Request timed out for packet 4 Request timed out
Host Unreachable for packet 5 Host Unreachable
=====
Summary:
=====
No response received from server
Packets Sent: 5
Packets Received: 0
MIN RTT: NA
MAX RTT: NA
AVG RTT: NA
Loss Rate: 100.000%
=====
root@534b2526fe5e:/workdir/PART3#
```

```
root@534b2526fe5e:/workdir/PART3# python3 ICMPpingClient.py
Pinging 172.17.0.1 using Python:

Enter number of pings to be sent:
5
Net Unreachable for packet 1 Net Unreachable
Request timed out for packet 2 Request timed out
Net Unreachable for packet 3 Net Unreachable
Net Unreachable for packet 4 Net Unreachable
Request timed out for packet 5 Request timed out
=====
Summary:
=====
No response received from server
Packets Sent: 5
Packets Received: 0
MIN RTT: NA
MAX RTT: NA
AVG RTT: NA
Loss Rate: 100.000%
=====
root@534b2526fe5e:/workdir/PART3#
```

```
root@534b2526fe5e:/workdir/PART3# python3 ICMPpingClient.py
Pinging 172.17.0.1 using Python:

Enter number of pings to be sent:
5
Protocol Unreachable for packet 1 Protocol Unreachable
Protocol Unreachable for packet 2 Protocol Unreachable
Protocol Unreachable for packet 3 Protocol Unreachable
Protocol Unreachable for packet 4 Protocol Unreachable
Request timed out for packet 5 Request timed out
=====
Summary:
=====
No response received from server
Packets Sent: 5
Packets Received: 0
MIN RTT: NA
MAX RTT: NA
AVG RTT: NA
Loss Rate: 100.000%
=====
root@534b2526fe5e:/workdir/PART3#
```

```

root@534b2526fe5e:/workdir/PART3# python3 ICMPPingerClient.py
Pinging 172.17.0.1 using Python:

Enter number of pings to be sent:
5
Communication with Destination Network is Administratively Prohibited for packet 1 C
ommunication with Destination Network is Administratively Prohibited
Communication with Destination Network is Administratively Prohibited for packet 2 C
ommunication with Destination Network is Administratively Prohibited
Communication with Destination Network is Administratively Prohibited for packet 3 C
ommunication with Destination Network is Administratively Prohibited
Communication with Destination Network is Administratively Prohibited for packet 4 C
ommunication with Destination Network is Administratively Prohibited
Request timed out for packet 5 Request timed out
=====
Summary:
=====
No response received from server
Packets Sent: 5
Packets Received: 0
MIN RTT: NA
MAX RTT: NA
AVG RTT: NA
Loss Rate: 100.000%
=====
root@534b2526fe5e:/workdir/PART3#

```

```

root@534b2526fe5e:/workdir/PART3# python3 ICMPPingerClient.py
Pinging 172.17.0.1 using Python:

Enter number of pings to be sent:
5
Request timed out for packet 1 Request timed out
Communication with Destination Host is Administratively Prohibited for packet 2 Comm
unication with Destination Host is Administratively Prohibited
Communication with Destination Host is Administratively Prohibited for packet 3 Comm
unication with Destination Host is Administratively Prohibited
Communication with Destination Host is Administratively Prohibited for packet 4 Comm
unication with Destination Host is Administratively Prohibited
Communication with Destination Host is Administratively Prohibited for packet 5 Comm
unication with Destination Host is Administratively Prohibited
=====
Summary:
=====
No response received from server
Packets Sent: 5
Packets Received: 0
MIN RTT: NA
MAX RTT: NA
AVG RTT: NA
Loss Rate: 100.000%
=====
root@534b2526fe5e:/workdir/PART3#

```

```
root@534b2526fe5e:/workdir/PART3# python3 ICMPPingClient.py
Pinging 172.17.0.1 using Python:

Enter number of pings to be sent:
5
Communication Administratively Prohibited for packet 1 Communication Administratively Prohibited
Request timed out for packet 2 Request timed out
Communication Administratively Prohibited for packet 3 Communication Administratively Prohibited
Communication Administratively Prohibited for packet 4 Communication Administratively Prohibited
Communication Administratively Prohibited for packet 5 Communication Administratively Prohibited
=====
Summary:
=====
No response received from server
Packets Sent: 5
Packets Received: 0
MIN RTT: NA
MAX RTT: NA
AVG RTT: NA
Loss Rate: 100.000%
=====
root@534b2526fe5e:/workdir/PART3#
```



**THE
END**