Assignment : ICMP Pinger
Name : Surbhi Thole
NetId: sst390

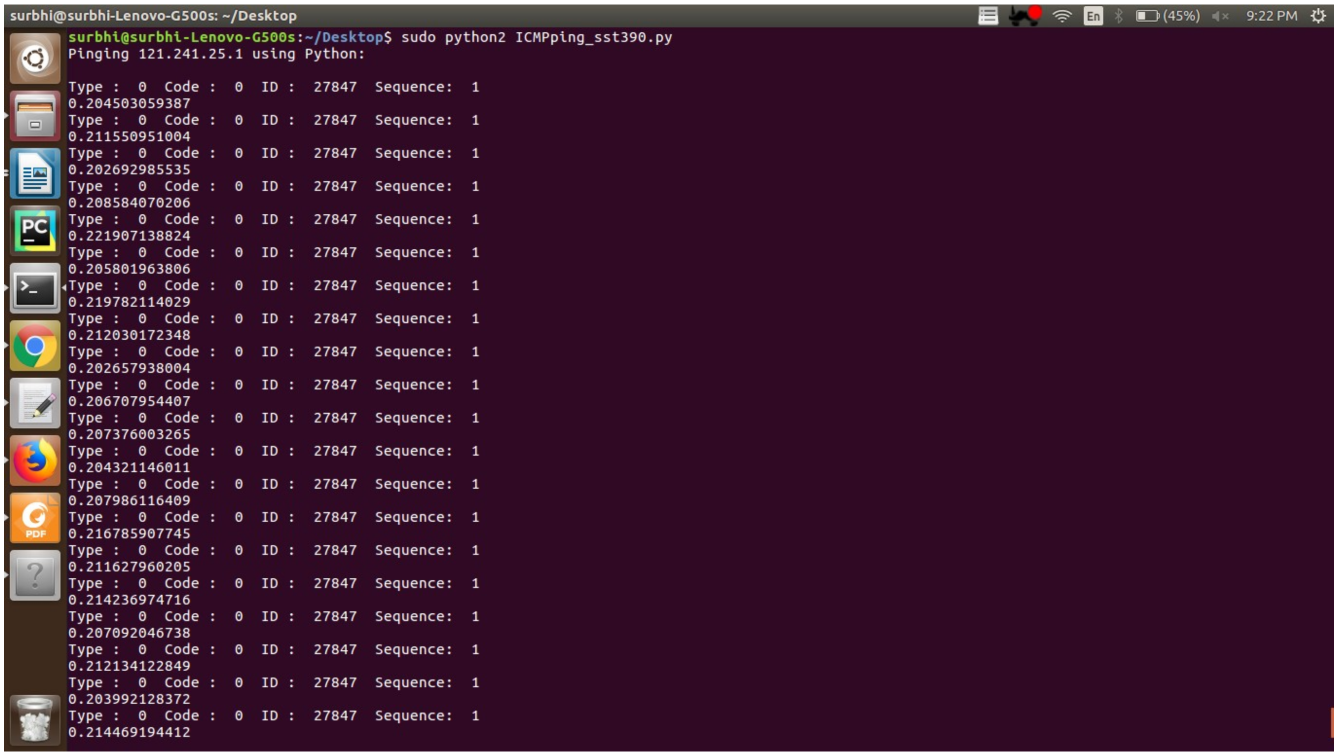
Code:

```
from socket import *
import os
import sys
import struct
import time
import select
import binascii
import socket

ICMP_ECHO_REQUEST = 8
RTT = []
sent = 0;
Recieve = 0;


def checksum(str):
    csum = 0
    countTo = (len(str) / 2) * 2
    count = 0
    while count < countTo:
        thisVal = ord(str[count + 1]) * 256 + ord(str[count])
        csum = csum + thisVal
        csum = csum & 0xffffffffL
        count = count + 2
    if countTo < len(str):
        csum = csum + ord(str[len(str) - 1])
        csum = csum & 0xffffffffL
    csum = (csum >> 16) + (csum & 0xffff)
    csum = csum + (csum >> 16)
    answer = ~csum
    answer = answer & 0xffff
    answer = answer >> 8 | (answer << 8 & 0xff00)
    return answer


def receiveOnePing(mySocket, ID, timeout, destAddr):
    global Recieve, RTT
    timeLeft = timeout
    while 1:
        startedSelect = time.time()
        whatReady = select.select([mySocket], [], [], timeLeft)
        howLongInSelect = (time.time() - startedSelect)
        if whatReady[0] == []:  # Timeout
```

```
            return "Request time out"
        time_rec = time.time()
        recPacket, addr = mySocket.recvfrom(1024)
        # Fill in start
        # Fetch the ICMP header from the IP packet
        icmpHeader = recPacket[20:28]
        Type, code, revChecksum, recId, recSequence = struct.unpack('bbHHh', icmpHeader)
            print "Type : ",Type, " Code : ", code, " ID : ", recId, " Sequence: ", recSequence
        if ID == recId:
            bytesInDouble = struct.calcsize('d')
            timeData = struct.unpack('d', recPacket[28:28 + bytesInDouble])[0]
            RTT.append(time_rec - timeData)
            Recieve += 1
            return time_rec - timeData
        else:
            return "ID is different!"
        # Fill in end
        timeLeft = timeLeft - howLongInSelect
        if timeLeft <= 0:
            return "Request time out"


def sendOnePing(mySocket, destAddr, ID):

    global sent
    # Header is type (8), code (8), checksum (16), id (16), sequence (16)

    myChecksum = 0

    # Make a dummy header with a 0 checksum.
    # struct -- Interpret strings as packed binary data
    header = struct.pack("bbHHh", ICMP_ECHO_REQUEST, 0, myChecksum, ID, 1)
    # Calculate the checksum on the data and the dummy header.
    data = struct.pack("d", time.time())
    myChecksum = checksum(header + data)
    # Get the right checksum, and put in the header
    if sys.platform == 'darwin':
        myChecksum = socket.htons(myChecksum) & 0xffff
        # Convert 16-bit integers from host to network byte order.
    else:
        myChecksum = socket.htons(myChecksum)
    header = struct.pack("bbHHh", ICMP_ECHO_REQUEST, 0, myChecksum, ID, 1)
    packet = header + data
    mySocket.sendto(packet, (destAddr, 1))
    sent += 1
    # AF_INET address must be tuple, not str
    # Both LISTS and TUPLES consist of a number of objects
    # which can be referenced by their position number within the object
```

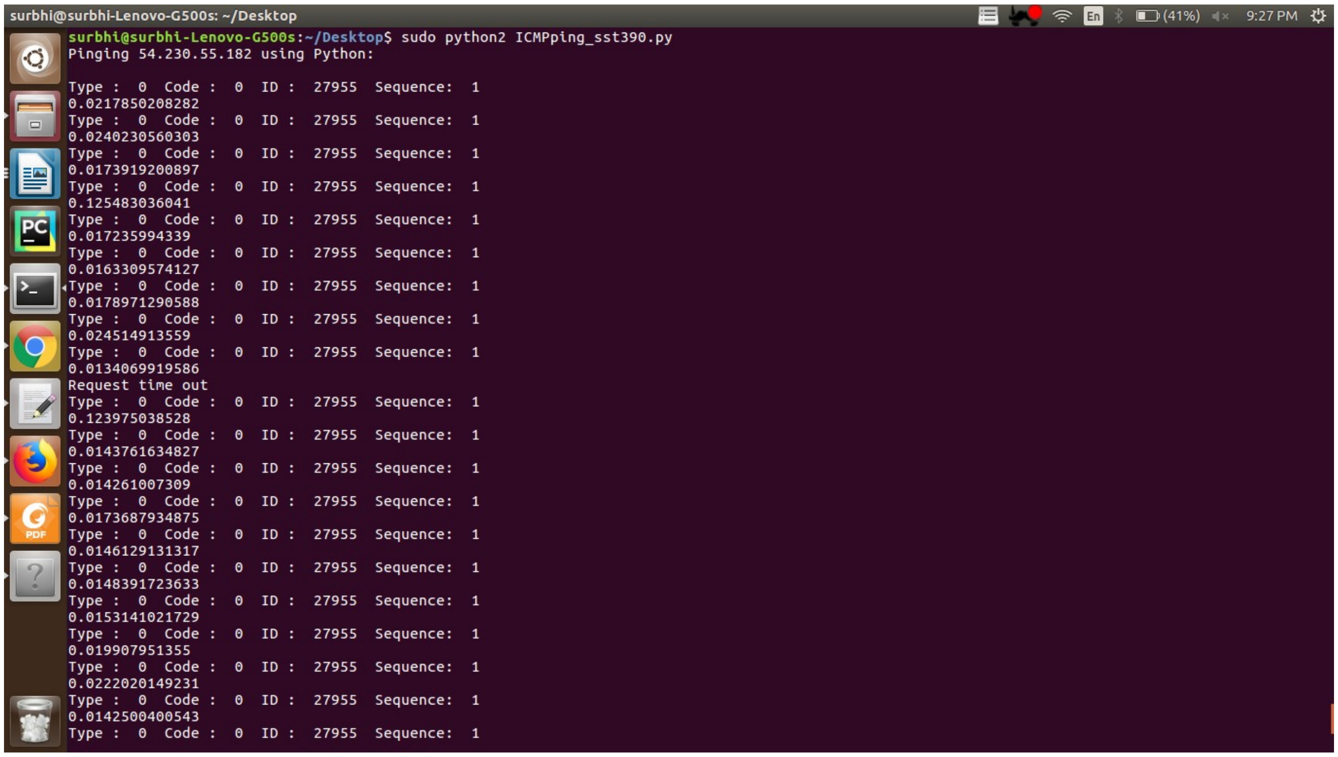
```
def doOnePing(destAddr, timeout):
    icmp = socket.getprotobyname("icmp")
    # SOCK_RAW is a powerful socket type. For more details see:http://sock-raw.org/papers/sock_raw
    # Fill in start
    # Create Socket here
    try:
        mySocket = socket.socket(socket.AF_INET, socket.SOCK_RAW, icmp)
    except socket.error, (errno, msg):
        if errno == 1:
            raise socket.error(msg)
    # Fill in end
    myID = os.getpid() & 0xFFFF  # Return the current process i
    sendOnePing(mySocket, destAddr, myID)
    delay = receiveOnePing(mySocket, myID, timeout, destAddr)
    mySocket.close()
    return delay


def ping(host, timeout=1):
    # timeout=1 means: If one second goes by without a reply from the server,
    dest = socket.gethostbyname(host)
    print "Pinging " + dest + " using Python:"
    print ""
    # Send ping requests to a server separated by approximately one second
    while 1:
        delay = doOnePing(dest, timeout)
        print delay
        time.sleep(1)  # one second
    return delay


ping("www.bom.gov.au")
```

1) Asian Continent: ping("www.mu.ac.in")



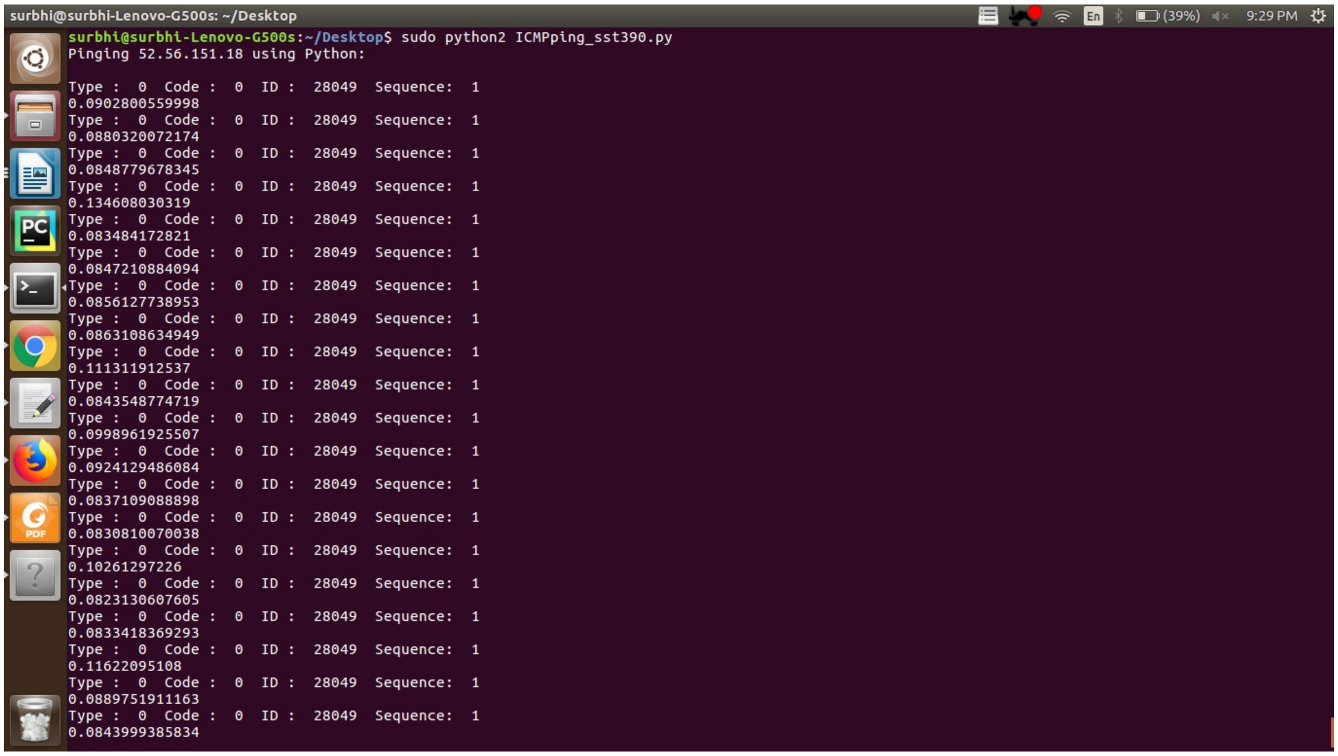
```
surbhi@surbhi-Lenovo-G500s: ~/Desktop
surbhi@surbhi-Lenovo-G500s:~/Desktop$ sudo python2 ICMPping_sst390.py
Pinging 121.241.25.1 using Python:

Type :  0  Code :  0  ID :  27847  Sequence:  1
0.204503059387
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.211550951004
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.202692985535
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.208584070206
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.221907138824
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.205801963806
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.219782114029
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.212030172348
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.202657938004
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.206707954407
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.207376003265
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.204321146011
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.207986116409
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.216785907745
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.211627960205
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.214236974716
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.207092046738
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.212134122849
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.203992128372
Type :  0  Code :  0  ID :  27847  Sequence:  1
0.214469194412
```

2) North – American Continent: ping("www.amazon.com")



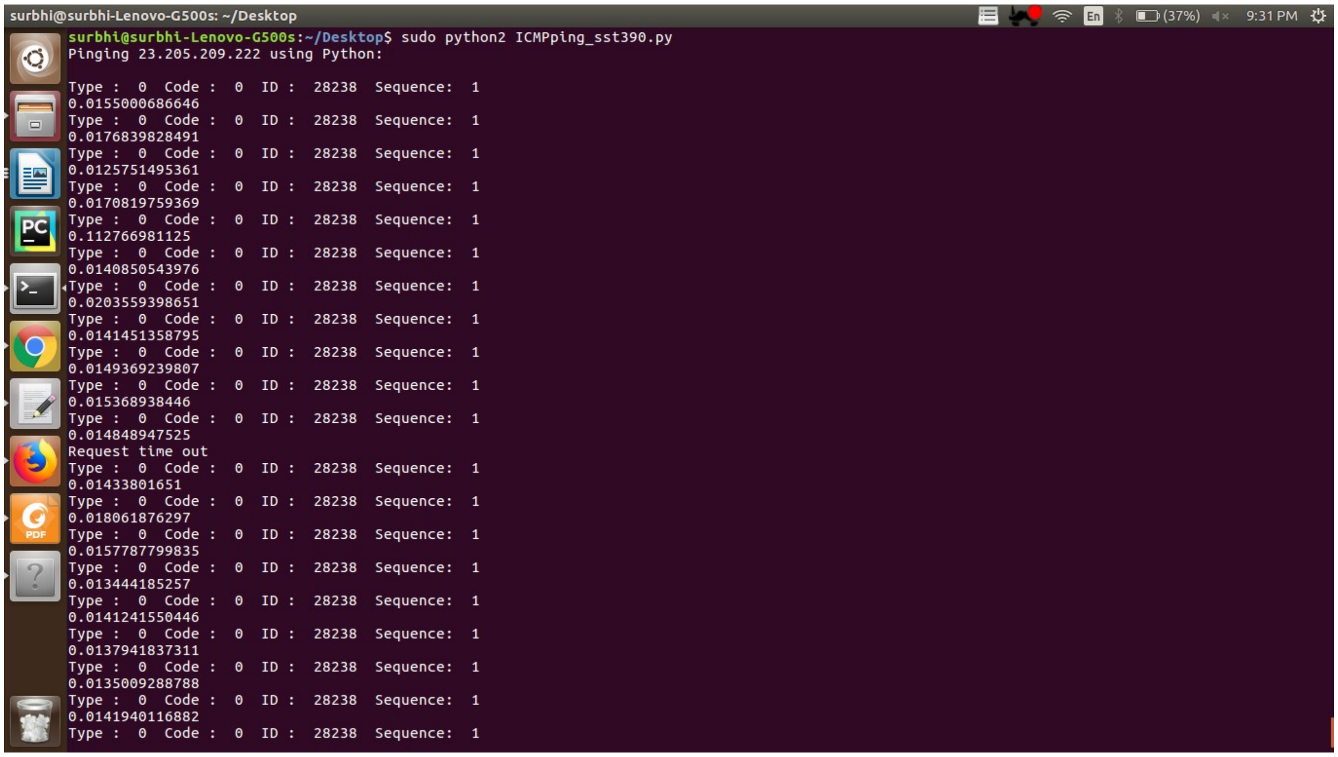
```
surbhi@surbhi-Lenovo-G500s: ~/Desktop
surbhi@surbhi-Lenovo-G500s:~/Desktop$ sudo python2 ICMPping_sst390.py
Pinging 54.230.55.182 using Python:

Type :  0  Code :  0  ID :  27955  Sequence:  1
0.0217850208282
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.0240230560303
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.0173919200897
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.125483036041
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.017235994339
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.0163309574127
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.0178971290588
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.024514913559
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.0134069919586
Request time out
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.123975038528
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.0143761634827
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.014261007309
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.0173687934875
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.0146129131317
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.0148391723633
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.0153141021729
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.019907951355
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.0222020149231
Type :  0  Code :  0  ID :  27955  Sequence:  1
0.0142500400543
Type :  0  Code :  0  ID :  27955  Sequence:  1
```

3) European continent : ping(“www.politico.eu”)



```
surbhi@surbhi-Lenovo-G500s:~/Desktop$ sudo python2 ICMPping_sst390.py
Pinging 52.56.151.18 using Python:

Type :  0  Code :  0  ID :  28049  Sequence:  1
0.0902800559998
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.0880320072174
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.0848779678345
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.134608030319
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.083484172821
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.0847210884094
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.0856127738953
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.0863108634949
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.111311912537
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.0843548774719
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.0998961925507
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.0924129486084
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.0837109088898
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.0830810070038
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.10261297226
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.0823130607605
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.0833418369293
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.11622095108
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.0889751911163
Type :  0  Code :  0  ID :  28049  Sequence:  1
0.0843999385834
```

4) Australia : ping(“www.bom.gov.au”)



```
surbhi@surbhi-Lenovo-G500s:~/Desktop$ sudo python2 ICMPping_sst390.py
Pinging 23.205.209.222 using Python:

Type :  0  Code :  0  ID :  28238  Sequence:  1
0.0155000686646
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.0176839828491
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.0125751495361
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.0170819759369
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.112766981125
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.0140850543976
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.0203559398651
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.0141451358795
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.0149369239807
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.015368938446
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.014848947525
Request time out
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.01433801651
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.018061876297
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.0157787799835
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.013444185257
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.0141241550446
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.0137941837311
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.0135009288788
Type :  0  Code :  0  ID :  28238  Sequence:  1
0.0141940116882
Type :  0  Code :  0  ID :  28238  Sequence:  1
```