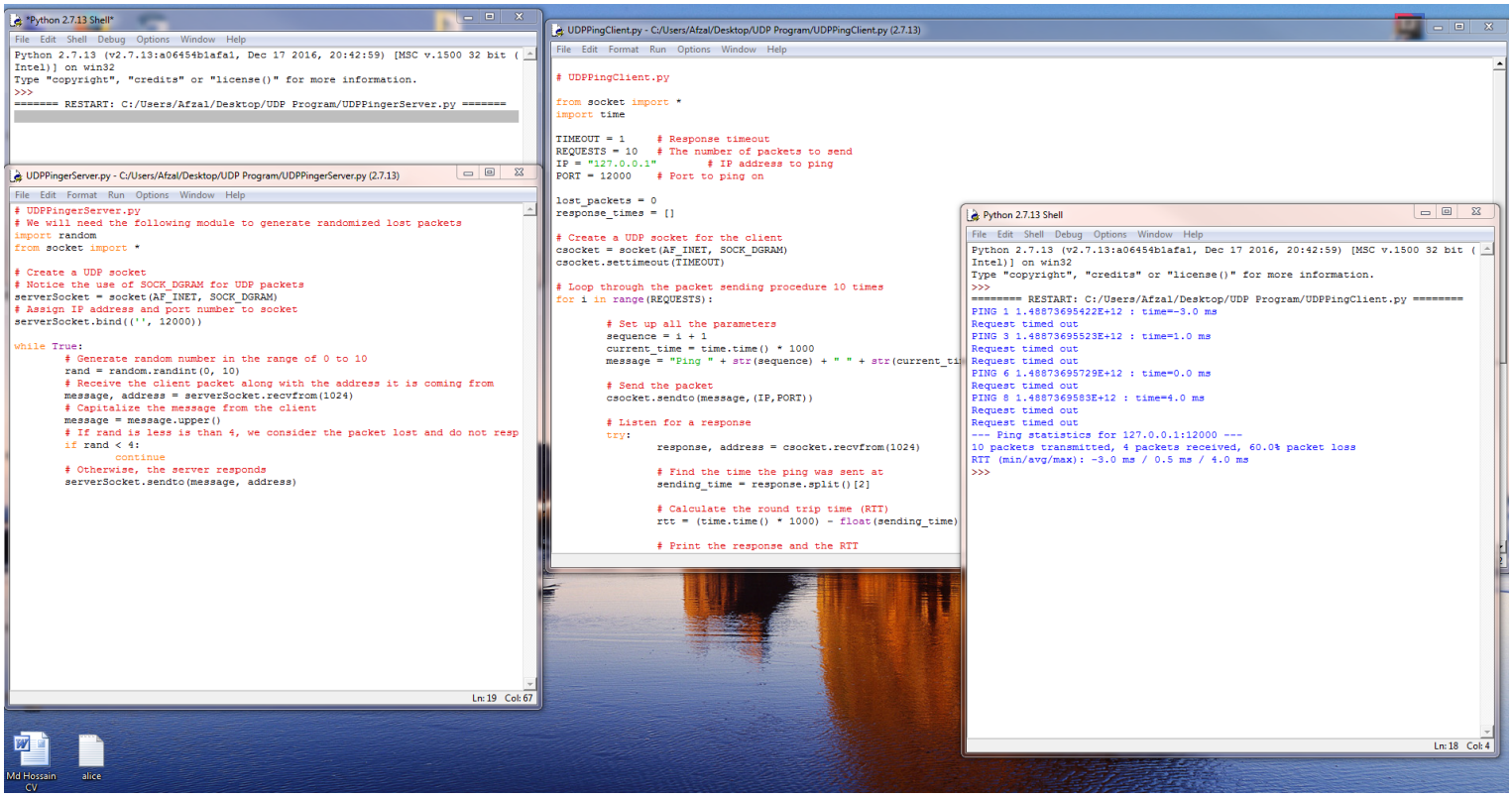
Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======= RESTART: C:/Users/Afzal/Desktop/UDP Program/UDPPingerServer.py =======
UDPPingerServer.py - C:/Users/Afzal/Desktop/UDP Program/UDPPingerServer.py (2.7.13)
File Edit Format Run Options Window Help
# UDPPingerServer.py
# We will need the following module to generate randomized lost packets
import random
from socket import *
# Create a UDP socket
# Notice the use of SOCK_DGRAM for UDP packets
serverSocket = socket(AF_INET, SOCK_DGRAM)
# Assign IP address and port number to socket
serverSocket.bind(('', 12000))
while True:
# Generate random number in the range of 0 to 10
rand = random.randint(0, 10)
# Receive the client packet along with the address it is coming from
message, address = serverSocket.recvfrom(1024)
# Capitalize the message from the client
message = message.upper()
# If rand is less is than 4, we consider the packet lost and do not resp
if rand < 4:
continue
# Otherwise, the server responds
serverSocket.sendto(message, address)
Ln: 19 Col: 67
UDPPingClient.py - C:/Users/Afzal/Desktop/UDP Program/UDPPingClient.py (2.7.13)
File Edit Format Run Options Window Help
# UDPPingClient.py
from socket import *
import time
TIMEOUT = 1 # Response timeout
REQUESTS = 10 # The number of packets to send
IP = "127.0.0.1" # IP address to ping
PORT = 12000 # Port to ping on
lost_packets = 0
response_times = []
# Create a UDP socket for the client
csocket = socket(AF_INET, SOCK_DGRAM)
csocket.settimeout(TIMEOUT)
# Loop through the packet sending procedure 10 times
for i in range(REQUESTS):
# Set up all the parameters
sequence = i + 1
current_time = time.time() * 1000
message = "Ping " + str(sequence) + " " + str(current_ti
# Send the packet
csocket.sendto(message,(IP,PORT))
# Listen for a response
try:
response, address = csocket.recvfrom(1024)
# Find the time the ping was sent at
sending_time = response.split()[2]
# Calculate the round trip time (RTT)
rtt = (time.time() * 1000) - float(sending_time)
# Print the response and the RTT
Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (
Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:/Users/Afzal/Desktop/UDP Program/UDPPingClient.py ========
PING 1 1.48873695422E+12 : time=-3.0 ms
Request timed out
PING 3 1.48873695523E+12 : time=1.0 ms
Request timed out
Request timed out
PING 6 1.48873695729E+12 : time=0.0 ms
Request timed out
PING 8 1.48873695583E+12 : time=4.0 ms
Request timed out
Request timed out
--- Ping statistics for 127.0.0.1:12000 ---
10 packets transmitted, 4 packets received, 60.0% packet loss
RTT (min/avg/max): -3.0 ms / 0.5 ms / 4.0 ms
>>>
Ln: 18 Col: 4
Md Hossain CV
alice