Gary Williams

CS 576 Spring 2017

Program 2 Due 4/3/17

**Program 2: UDP Socket, Client - Server**

server.py:

#!/usr/bin/python

# Gary Williams

# CS 576

# Prog 2 Due 4/3/17

# This is a basic UDP socketed server. It connects to the

# local host and listens to port '5432' as specified in

# the assignment. If a message is received, it

# concatenates a reply message and then send that reply

# over port '5433'. Because UDP is a connection-less

# protocol, the server merely listens and replies.

#

# referenced: https://pymotw.com/2/socket/udp.html

import socket

import sys

gibberish = "- Roger Roger. Read you loud and clear"

#Create socket and save port in address before binding

sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

server\_address = ('localhost', 5432)

print >>sys.stderr, 'Starting up on %s port %s' % server\_address

sock.bind(server\_address)

#Receive loop; when data received, concat and reply

while 1:

print >>sys.stderr, 'Waiting to receive message'

data, address = sock.recvfrom(5433)

print >>sys.stderr, 'Received %s bytes from %s' % (len(data), address)

print >>sys.stderr, data

if data:

data = data + gibberish

sent = sock.sendto(data, address)

print >>sys.stderr, 'Sent %s bytes back to %s' % (sent, address)

client.py:

#!/usr/bin/python

# Gary Williams

# CS 576

# Prog 2 Due 4/3/17

# This is a basic UDP client socketed connection.

# It prompts the user for the message to be sent,

# echos that message before sending it to the

# and printing the reply. Msg sent to server over

# port 5432 and reply is heard on 5433.

#

# referenced: https://pymotw.com/2/socket/udp.html

import socket

import sys

#Prompt for input and validate

msg = raw\_input("Enter short message: ")

while not msg:

msg = raw\_input("Enter a valid message: ")

#Create socket before trying to send

sock = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

server\_address = ('localhost', 5432)

try:

print >>sys.stderr, 'Sending: "%s"' % msg

sent = sock.sendto(msg, server\_address)

#Msg sent, program will block until recv

data, server = sock.recvfrom(5433)

print >>sys.stderr, 'Received: "%s"' % data

finally:

sock.close()

screenshots:



