```
"""
Program written by: Cambrea Earley

This file establishes the connection between the raspberry Pi
and the AWS Server

This is for sending packets containing:
audio snippet, DOA, mic ID

This is for receiving packets containing:
audio snippet
"""

import socket, sys, signal, queue, pickle
from string import *
#import pymedia.audio.acodec as ascodec

#socket params
SERVER_IP = socket.gethostbyname(socket.gethostname())
PORT = 8
CHUNK = 512 #1024

#To exit the program on a keyboard interrupt(Eventually will use this to shut down system)
def signal_handler(signal, frame):
    s.close()
    sys.exit(0)

signal.signal(signal.SIGINT, signal_handler)



#compression
# sampleRate = 44100
# cparams= { 'id': acodec.getCodecID( 'mp3' ),
#             'bitrate': 128000/4,
#             'sample_rate': sampleRate,
#             'channels': 1 }
# ac= acodec.Encoder(cparams)

# msgToSend = [Audio, DOA, micID]




# set up socket
try:
    s = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)
    print("socket created")
```

```
except socket.error:
    print("Could not make socket. Error: ")
    sys.exit()

#set non blocking
s.setblocking(0)
#connect
s.sendto(str.encode("Hello Server") ,(SERVER_IP, PORT))

fail = 0
retry = 1
while retry == 1:
    retry = 0
    try:
        data, addr = s.recvfrom(CHUNK)
    except BlockingIOError:
        retry = 1
        fail = fail + 1
        if fail == 20:
            print("Could not connect")
            exit()
print("Message in client: " + data.decode())

#wait for Mic ID from serverp
mic_id = int(data.decode())
print(mic_id)


audioInQ = queue.Queue()
audioOutQ = queue.Queue()

### Used for Testing
audioOutQ.put_nowait(["D1", 10, 0])
audioInQ.put_nowait(["D2", 11, 0])

#Start sending and receiving
while 1:
    #Getting new data
    try:
        dataByte, addr = s.recvfrom(CHUNK)
        data = "!"
    except BlockingIOError:
        data = ""

    #decode data
    if data == "!":
        try:
            data = pickle.loads(dataByte)
        except:
```

```
            data = ""


    outMsg = []
    if not audioOutQ.empty():
        outMsg = audioOutQ.get_nowait()
        print("Got new message from Q")
        print(outMsg)

    if len(data) > 0:
        #place input audio in queue to be played
        audioInQ.put_nowait(data)
        print("DATAIN")
        print(data)


    if len(outMsg) > 0 :
        #send audio
        outMsg[2] = mic_id
        s.sendto(pickle.dumps(outMsg), (SERVER_IP, PORT))
        print("SENDING: ")
        print(outMsg)
```