IOE Client Flow-Chart

The Client has a variable “server_address” to store the server_address and determine if the user has input the server address or not depending on the “server_address” is empty or not. “raw_input” prompts the user with the string inn the parameter and grabs the user input after the “Enter” key.

```
##define server address string
server_address = ''


# Send data
while True:
##    if the server address is empty then prompt the user to enter the server address
    if(server_address == ''):
        message = raw_input('Enter the CONNECT,IP Address,Port Number to establish connection.\n')
    else:
##    if there is a server address stored already, then prompt the user to enter the command
        message = raw_input('Enter your command.\n')
```

When the Command start with “CONNECT”, we call the functions “sock.connect(server_address” and “sock.sendall(message)” to send the input data to the server. “sock.connect” is a method from the Socket library that make the connection to the server address in the parameter. “sock.sendall” sends the message in the parameter to the server once the connection is made.

```
if message.startswith("CONNECT"):
    if (len(message.split(','))== 3):
          if command is connect, then try to make connect to the server
        try:
            server_address = (message.split(',')[1],int((message.split(',')[2])))
            sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            sock.connect(server_address)
            sock.sendall(message)
```

Then, wait for the response from the server and close the connection in case the server halts. ”sock.recv” receives any data from the server in the amount of bytes provided in the parameter. Here we also find the amount of data we received from the server by accmulating the length of the data. “sock.close” closes the connection with the server.

```
   wait to receive the connection messages from the server
data = sock.recv(1028)
amount_received += len(data)
print >>sys.stderr, 'received "%s"' % data
sock.close()
```

IOE Client Flow-Chart

If there are any error encountered during the connection, usually because the IP address or the port provided by the user is incorrect. Print out a feedback to the user stating the IP address or port is incorrect. “sys.stderr” prints out the string to the screen.

```
except:
      if anything went wrong, then return message to re-enter address
    print >>sys.stderr, "Please make sure the IP and port address are enterd correctly"
```