

Tatianna Robinson
CSCI379 Programming Assignment 1

The code in this program uses TCP as the transport layer protocol. `TCPServer.py` runs on the server and waits to receive a connection request from a client, using port 9999. Once `TCPClient.py` runs in a client end system, it will create a socket on the client side with the line of code `clientSocket = socket(AF_INET, SOCK_STREAM)` and attempt to establish a TCP connection with the host with `hostname` (IP address) stored in the variable `serverName` in the statement `clientSocket.connect((serverName, serverPort))`. The first message sent from the client contains no data, but its purpose is to initiate the handshake protocol.

The host running the server process receives the incoming connection request. It receives the source port number and IP address as well as the destination port number and IP address (its own IP address) from the client, and creates its a new socket with the line `connectionSocket, addr = serverSocket.accept()`. With this socket, the TCP connection will be established. The next line of code `sentence = connectionSocket.recv(1024)` waits for data to be sent to the server from the client side.

Meanwhile on the client side, the process requests a sentence from the user (user enters sentence and ends with carriage return) and the client sends the sentence through the socket and into the TCP connection with `clientSocket.send(sentence)`. In a similar fashion, the client process then requests and waits for the user to enter the `command` which determines how the server will transform the string, and sends this data to the server as well. The server now has values for `sentence` and `command`.

The client then waits to receive a response from the server with `modifiedSentence = clientSocket.recv(1024)`.

The user entered two strings, `sentence` and `command`, from the client side, sent this to the server, and with if-else statements, the server sends the correct modification of the string based on the user's selection, back to the client. After sending the modified sentence back to the client with `connectionSocket.send()`, the Server closes the TCP connection with the client `connectionSocket.close()`. Since `serverSocket` remains open, another client may request a connection.

Finally on the client side, `modifiedSentence` receives a value from the server and the result is printed to the client's screen. After this, the TCP connection with the server ends with the line `clientSocket.close()`.