**University of Bahrain ITCE 352**

**College of Information Technology**

**Department of Computer Engineering Lab experiment no. 3**

TCP Client-Server application

# Objectives:

1. Learn how to create and deal with TCP sockets at client and server sides.
2. Learn how to connect to a remote socket.
3. Learn how to list and retrieve information about local files and folders

# Equipment and Software Tools Required:

1. PYTHON
2. socket module

# Introduction:

Transmission Control Protocol (TCP) is a connection-oriented protocol. It provides reliability, flow control, congestion control, and other services. In TCP the sockets with type SOCK\_STREAM should be generated and used. The server should create a passive socket and binds a port number to it to listen to connection establishment requests. Furthermore, the server needs an active socket to exchange the messages between the client and the server. A connection must be established before data messages exchange. Both client and server use *send()* and *recv()* to send and receive data messages. After sending the last data message, the connection should be concluded, and resources should be released. Therefore, at the end, the *close()* method should be used to tear down the connection.

# Requirements

Write two Python scripts (client and server) that implement TCP to communicate over the network. The client, on the other hand, establishes a connection with the server and sends several requests. The server, on the other hand, can list local files at a folder (the folder is available on the server side) and provide the list of files, and read text file contents and send them to the client.

The client has the following options:

* + Ask for list of files in the server folder
  + Select a file from the list provided by the server to retrieve its content
    - receive the content of the file and save it in a new local file (in the client side)
  + Quit and print out a “good bye” message
    - Keep the script running until the used choose to quit The server does the following:
  + keep listening for requests
  + Send the list of local files to the client (to select from)
  + Open the selected file, read its contents, and sends the content to the client.
    - Close and release the file
  + Quit when it receives “quit” message from the client



Lab Experiment no. 3 1 | P a g e

# Note:

Beside the functionality of your code, you will be marked also on the neatness of the code, and neatness of the output.