LAB NO. 11

TCP/IP IMPLEMENTATION USING PYTHON SOCKET PROGRAMMING



DATA COMMUNICATION AND COMPUTER NETWORKS LAB

Submitted by:

Name : Muhammad Musa

Reg no. : 22PWCSE2157

Class Section : C



Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_

Date: June 01 2025

Submitted to: Dr. Yasir Saleem Afridi

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

**CSE 303L: Data Communication and Computer Networks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Demonstration of Concepts** | **Poor (Does not meet expectation (1))**  The student failed to demonstrate a clear understanding of the assignment concepts | **Fair (Meet Expectation (2-3))**  The student demonstrated a clear understanding of some of the assignment concepts | **Good (Exceeds Expectation (4-5)**  The student demonstrated a clear understanding of the assignment concepts | **Score**  **30%** |
| **Accuracy** | The student mis-configured enough network settings that the lab computer couldn't function properly on the network | The student configured enough network settings that the lab computer partially functioned on the network | The student configured the network settings that the lab computer fully functioned on the network | **30%** |
| **Following Directions** | The student clearly failed to follow the verbal and written instructions to successfully complete the lab | The student failed to follow the some of the verbal and written instructions to successfully complete all requirements of the lab | The student followed the verbal and written instructions to successfully complete requirements of the lab | **20%** |
| **Time Utilization** | The student failed to complete even part of the lab in the allotted amount of time | The student failed to complete the entire lab in the allotted amount of time | The student completed the lab in its entirety in the allotted amount of time | **20%** |

**Credit Hours: 1**

LAB #11: TCP/IP Implementation using Python Socket Programming

# Objectives of Lab

* To understand the implementation of TCP/IP using Python socket programming.
* To establish a basic client-server communication system.
* To explore the functionality of Python’s socket library.

# Introduction to Python Programming

Python is a powerful, high-level programming language renowned for its readability and simplicity. One of its strong suits is built-in support for network programming via the socket library. This lab aims to introduce the core functions of socket programming and demonstrate how TCP/IP-based communication works in a client-server model using Python.

# Introduction to Python Socket Library and Its Various Functions

Python’s socket module provides a standard way of networking in Python and is used for implementing clients and servers. It supports both TCP and UDP protocols.

# Client-Server Communication Using Socket Library

## Server Code (TCP Server)

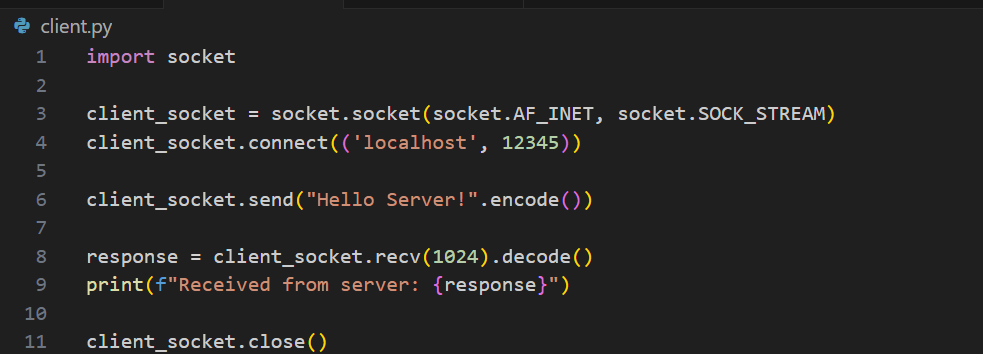
import socket  
  
# Create socket object  
server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
server\_socket.bind(('localhost', 12345))  
server\_socket.listen(1)  
  
print("Server is waiting for client connection...")  
  
# Accept connection  
client\_socket, address = server\_socket.accept()  
print(f"Connected to {address}")  
  
# Receive message  
message = client\_socket.recv(1024).decode()  
print(f"Received from client: {message}")  
  
# Send response  
client\_socket.send("Hello Client, Message received!".encode())  
  
# Close sockets  
client\_socket.close()  
server\_socket.close()

## Client Code (TCP Client)

import socket  
  
# Create socket object  
client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
client\_socket.connect(('localhost', 12345))  
  
# Send message  
client\_socket.send("Hello Server!".encode())  
  
# Receive response  
response = client\_socket.recv(1024).decode()  
print(f"Received from server: {response}")  
  
# Close socket  
client\_socket.close()

# Flowchart of Client-Server Communication Using Python Socket Library

CLIENT SERVER  
 | |  
 | socket() | <- Create socket  
 | connect() | <- bind(), listen()  
 |-----------------------------> | <- accept()  
 | send() |  
 |-----------------------------> |  
 | | <- recv()  
 | | <- send()  
 | recv() |  
 |<----------------------------- |  
 | close() | <- close()  
 | |



A computer screen shot of text

AI-generated content may be incorrect.

## Sample Output

**Server Output:**

A computer screen with white text

AI-generated content may be incorrect.

**Client Output:**

A screenshot of a computer

AI-generated content may be incorrect.

**Conclusion**

This lab successfully demonstrated the fundamentals of TCP/IP communication using Python's socket programming. We implemented a basic client-server model where a client sends a message to a server and receives a confirmation in response. This lays the foundation for understanding more advanced network communication systems.