Assignment 7

Implementation of TCP/UDP Socket Programming

NAME: Shirish Manoj Bobde

Reg. No.: 812

Roll No.: ECE/21152

Problem Statement

Write a TCP socket program (in C/C++/Java/Python) to implement a client-server program using TCP sockets. The client will send a message to the server, and the server will perform a cyclic redundancy check (CRC) on the message to detect errors. The server will then send the result back to the client. Display appropriate messages to the user indicating the status of the connection and the result of the CRC check.

Code:

**Server**

import socket

def xor(a, b):

    result = []

    for i in range(1, len(b)):

        if a[i] == b[i]:

            result.append('0')

        else:

            result.append('1')

    return ''.join(result)

def mod2div(dividend, divisor):

    pick = len(divisor)

    tmp = dividend[0: pick]

    print("\nDivisor:", divisor)

    print("Dividend:", dividend)

    while pick < len(dividend):

        if tmp[0] == '1':

            tmp = xor(divisor, tmp) + dividend[pick]

        else:

            tmp = xor('0'\*pick, tmp) + dividend[pick]

        pick += 1

    if tmp[0] == '1':

        tmp = xor(divisor, tmp)

    else:

        tmp = xor('0'\*pick, tmp)

    checkword = tmp

    return checkword

def receiveData(client\_socket):

    # Receive data and CRC key from the client

    received\_data = client\_socket.recv(1024).decode()

    key, data = received\_data.split(';')

    l\_key = len(key)

    appended\_data = data + '0'\*(l\_key-1)

    checksum = mod2div(appended\_data, key)

    print("\nKey:", key)

    print("Original Data:", data)

    print("Checksum:", checksum)

    print("Data Sent by Client:", data + checksum )

    print("Verification Result:", "Error Detected!" if checksum != '0'\*(l\_key-1) else "No Errors Detected")

# Set up the server socket

server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server\_socket.bind(('localhost', 12345))

server\_socket.listen(1)

print("Server is listening for incoming connections...")

# Accept a client connection

client\_socket, client\_address = server\_socket.accept()

print("Connection established with", client\_address)

# Receive and process data from the client

receiveData(client\_socket)

# Close the sockets

client\_socket.close()

server\_socket.close()

**Client**

import socket

def xor(a, b):

    result = []

    for i in range(1, len(b)):

        if a[i] == b[i]:

            result.append('0')

        else:

            result.append('1')

    return ''.join(result)

def sendData(server\_socket, data, key):

    l\_key = len(key)

    appended\_data = data + '0'\*(l\_key-1)

    print("\nKey:", key)

    print("Original Data:", data)

    # Send CRC key and data to the server

    server\_socket.send((key + ';' + data).encode())

# Set up the client socket

client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

client\_socket.connect(('localhost', 12345))

# Input the CRC key from the client

key = input("Enter the CRC Key: ")

# Input data to be sent

data = input("Enter the Data to be Sent: ")

# Send CRC key and data to the server

sendData(client\_socket, data, key)

# Close the socket

client\_socket.close()

Output

