

TCP File Transfer Practical Work 1

Vu Ngoc Minh

1 Introduction

The purpose of this practical is to implement a simple file transfer system using TCP/IP. The work involves socket programming in Python, with one laptop acting as a server and the other as a client.

2 Protocol Design

The file transfer protocol involves:

- Establishing a connection between the client and the server.
- Sending the file name from the client to the server.
- Transferring the file in chunks of data.
- The server saving the file after all data is received.

3 System Interaction

Laptop 1 (Server)

Laptop 2 (Client)

4 Implementation

The implementation involves two Python scripts:

```
PS C:\Users\DELL> python -u "c:\DS\ds2025\ds2025-main\practical\practical 1 _ ba12-128\server.py"
Server is listening on 192.168.162.60:33333...
Connection accepted from ('192.168.162.60', 50825)
Receiving file: example.txtThis is the example file of Distributed system practical 1
File example.txtThis is the example file of Distributed system practical 1 received successfully.
PS C:\Users\DELL> python -u "c:\DS\ds2025\ds2025-main\practical\practical 1 _ ba12-128\client.py"
```

Figure 1: Enter Caption

```
PS C:\DS\ds2025\ds2025-main\practical\practical 1 _ ba12-128> & "c:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe" "c:\Users\DELL\vscode\extensions\ms-python.debugpy-2024.12.0-win32-x64\bundle\libs\debugpy\adapter\..\..\debugpy\launcher" "50823" "--" "c:\DS\ds2025\ds2025-main\practical\practical 1 _ ba12-128\client.py"
File example.txt sent successfully.
PS C:\DS\ds2025\ds2025-main\practical\practical 1 _ ba12-128>
```

Figure 2: Enter Caption

4.1 Server Code

The server listens for incoming connections and saves the file:

```
import socket

def start_server():
    server_ip = '0.0.0.0' # Listen on all available interfaces
    server_port = 33333 # Port to listen on

    # Create a socket object
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind((server_ip, server_port)) # Bind to the specified IP and port
    server_socket.listen(1) # Allow one client connection at a time
    print(f"Server is listening on {server_ip}:{server_port}...")

    # Accept a connection from the client
    conn, addr = server_socket.accept()
    print(f"Connection accepted from {addr}")

    # Receive the filename from the client
    filename = conn.recv(1024).decode()
    print(f"Receiving file: {filename}")

    # Receive the file content and save it
    with open(filename, 'wb') as file:
        while True:
            data = conn.recv(1024) # Receive data in chunks
            if not data:
                break
            file.write(data)

    print(f"File {filename} received successfully.")
    conn.close() # Close the connection
    server_socket.close() # Close the server socket

if __name__ == "__main__":
    start_server()
```

4.2 Client Code

The client connects to the server and sends the file:

```
import socket

def send_file(server_ip, server_port, filename):
    # Create a socket object
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect((server_ip, server_port)) # Connect to the server

    # Send the filename to the server
    client_socket.send(filename.encode())

    # Send the file content
    with open(filename, 'rb') as file:
        while (data := file.read(1024)): # Read the file in chunks
            client_socket.send(data)

    print(f"File {filename} sent successfully.")
    client_socket.close() # Close the client socket

if __name__ == "__main__":
    # Replace with the actual file name in the same directory
    server_ip = '192.168.1.20' # Server's IP address
    server_port = 33333 # Server's port
    filename = 'example.txt' # File to send

    send_file(server_ip, server_port, filename)
```

5 Results

The file transfer system was successfully tested. The following results were obtained:

- File transferred: 'example.txt'
- File size: 58 B
- Transfer time: 1 second

6 Roles

Contributed to this project:

- Vu Ngoc Minh: Developed the server script (laptop 1).
- Vu Ngoc Minh: Developed the client script (laptop 2).
- Vu Ngoc Minh: Prepared the report in LaTeX.