

# Practical Work 1: TCP File Transfer

Le Ba Hai Long

December 12, 2024

## 1 Introduction

The goal of this practical work is to implement a one-to-one file transfer system over TCP/IP using a client-server architecture. The implementation uses Python's socket library to establish a connection and transfer files between the client and server.

## 2 System Design

The system is designed as follows:

1. A server listens for incoming connections on a specified port.
2. A client connects to the server and initiates the file transfer.
3. The server receives the file and saves it to the local disk.

### 2.1 System Architecture

## 3 Protocol Design

The protocol design involves the following steps:

1. The client sends the name of the file to the server.
2. The client reads the file in chunks and sends each chunk to the server.
3. The server writes the received data to a file with the specified name.
4. The connection is closed after the file transfer is complete.

## 4 Implementation

The implementation of the file transfer system is done in Python. Below are the code snippets for the server and client.

### 4.1 Server Code

Listing 1: Server Code

```
import socket

HOST = '127.0.0.1'
PORT = 47392

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((HOST, PORT))
server.listen(1)

conn, addr = server.accept()

name = conn.recv(1024).decode()

with open(name, 'wb') as f:
    while True:
        data = conn.recv(1024)
        if not data:
            break
        f.write(data)

print(f"Complete")
conn.close()
server.close()
```

### 4.2 Client Code

Listing 2: Client Code

```
import socket

HOST = '127.0.0.1'
PORT = 48383
```

```

client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.connect((HOST, PORT))
print(f"server {HOST}:{PORT}")

name = 'prac1.txt'
client.send(name.encode())

with open(name, 'rb') as f:
    data = f.read(1024)
    while data:
        client.send(data)
        data = f.read(1024)

print("complete.")
client.close()

```

## 5 Conclusion

This practical work successfully demonstrates a simple file transfer system using TCP sockets. The implementation can be extended to include error handling, encryption, and support for multiple clients.