De La Salle University - Manila

College of Computer Studies

Machine Project #1:

TFTP Client Implementation

In Partial Fulfillment of the
Course Requirements for

Network Application Protocols

First Term, AY 2024 - 2025

Submitted by:

NSCOM01 - S12 - Group 5

Maristela, Kyle Gabriel A.

San Luis, Owen Phillip C.

Submitted to:

Ms. Katrina Ysabel Solomon

February 19, 2025

I. Introduction

This project is a **TFTP (Trivial File Transfer Protocol) client** implemented in **Python**. It allows users to **upload (put)** and **download (get)** files, including binary files, from a **TFTP server** using UDP, following **TFTP v2 (RFC 1350)** and extensions from **RFCs 2347**, **2348**, and **2349** for block size (blksize) and transfer size (tsize) negotiation.

II. Features

- Upload (put) and Download (get) support
- Customizable block size (blksize) negotiation
- Transfer size communication (tsize) for uploads
- Timeout handling with retries (max 3 attempts)
- Duplicate ACK handling
- File sequencing to ensure correct transfer
- Handles TFTP error messages properly

III. Requirements

- Windows OS
- Python 3.x
- tftpd32 by Ph. Jounin (TFTP Server for Windows)

IV. Installation

1. **Clone or download the project repository**

   ```
   git clone https://github.com/your-repo/tftp-client.git
   cd tftp-client
   ```

2. **Ensure Python 3 is installed**

   ```
   python --version
   ```

3. **Navigate to the project directory and run the client**

   ```
   python main.py
   ```

V. Setting up tftpd32 (Windows)

1. **Download tftpd32**

- Go to tftpd32.jounin.net

- Download tftpd32 (64-bit or 32-bit, depending on your system)

2. **Configure the TFTP Server**

- Open tftpd32.exe
- Under Current Directory, set the folder where files will be stored
- Ensure the TFTP Server tab is selected
- Set the Server Interface to 127.0.0.1

3. **Create a test file for download**

- Open Notepad, type "This is a test file", and save it as test.txt
- Place it in the Current Directory of tftpd32

VI. Usage

Running the TFTP Client

Run the script and follow the prompts:

```
python main.py
```

Example input:

```
TFTP Client Application

Write 'exit' to close the program.
Enter TFTP server IP address: 127.0.0.1
Connecting to TFTP server at 127.0.0.1:69...

Write 'exit' to disconnect from the server.
Enter operation (get for download, put for upload): get

Write 'exit' to return to main menu.
Enter filename: FileA.jpg
Enter blocksize (leave blank to skip):
```

Example Usage

Download a File from the Server (get)

```
Write 'exit' to disconnect from the server.
Enter operation (get for download, put for upload): get

Write 'exit' to return to main menu.
```

```
Enter filename: FileA.jpg
Enter blocksize (leave blank to skip):
```

Expected Output:

```
Negotiated blocksize: 512
Starting download of FileA.jpg...
Download complete!
```

Upload a File to the Server (put)

```
Write 'exit' to disconnect from the server.
Enter operation (get for download, put for upload): put

Write 'exit' to return to main menu.
Enter filename: FileA.jpg
Enter blocksize (leave blank to skip):
```

Expected Output:

```
Negotiated blocksize: 512
File size (tsize): 7944 bytes
Starting upload of FileA.jpg (7944 bytes)...
Upload complete!
```

VII. Error Handling

The client gracefully handles:

- Timeouts (retries up to 5 times before aborting)
- File not found errors
- Duplicate ACKs

1. Handling Timeout (Server Not Responding) and Errors

Test Case: Server is unreachable

Input:

```
Write 'exit' to disconnect from the server.
Enter operation (get for download, put for upload): put

Write 'exit' to return to main menu.
Enter filename: testing.txt
Enter blocksize (leave blank to skip):
```

Expected Output:

```
Negotiated blocksize: 512
Starting upload of testing.txt (None bytes)...
Error: Illegal TFTP operation
Warning: Timeout occurred, retrying 1/5...
Error: Illegal TFTP operation
Warning: Timeout occurred, retrying 2/5...
Error: Illegal TFTP operation
Warning: Timeout occurred, retrying 3/5...
Error: Illegal TFTP operation
Warning: Timeout occurred, retrying 4/5...
Error: Illegal TFTP operation
Warning: Timeout occurred, retrying 5/5...
Error: Maximum retries reached, aborting upload.
```

2. **Handling File Not Found (Server-Side)**

   Test Case: Attempting to download a non-existent file

   Input:

```
Write 'exit' to disconnect from the server.
Enter operation (get for download, put for upload): get

Write 'exit' to return to main menu.
Enter filename: dne.txt
Enter blocksize (leave blank to skip):
```

   Expected Output:

```
No OACK received. Using default values.
Starting download of dne.txt...
Warning: Timeout occurred, retrying 1/5...
Error: File not found
```

3. **Handling Duplicate ACKs**

   Test Case: Server sends duplicate ACKs\

Expected Output:

```
Warning: Unexpected packet received, retrying...
Warning: Timeout occurred, retrying 1/5...
Warning: Unexpected packet received, retrying...
Warning: Timeout occurred, retrying 2/5...
Warning: Unexpected packet received, retrying...
Warning: Timeout occurred, retrying 3/5...
Warning: Unexpected packet received, retrying...
Warning: Timeout occurred, retrying 4/5...
Warning: Unexpected packet received, retrying...
Warning: Timeout occurred, retrying 5/5...
Error: Maximum retries reached, aborting upload.
```