University of Memphis

Major in Computer Science

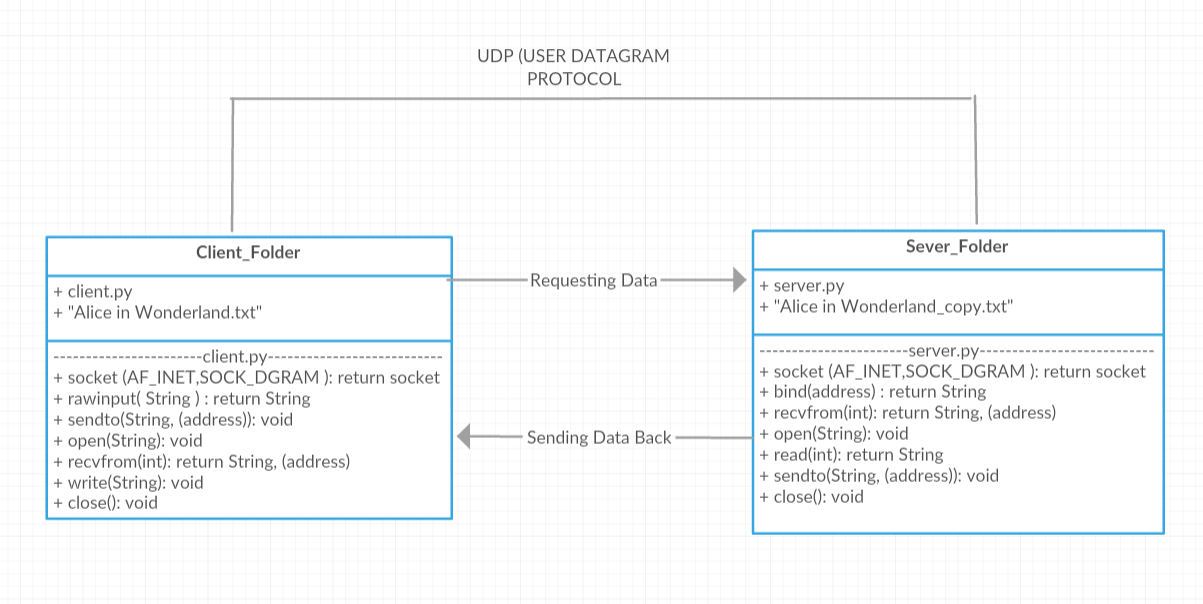
Author - Dung Le

Project Proposal

Client-Server File Transfer Using UDP Protocol

Instructor – Dr. Lan Wang

03/29/2018

1. Abstraction: The project will implement UDP file transfer protocol using python socket API for basic file transfer; however, UDP is not a reliable data transfer protocol, the packages can be lost during the transmission. For that reason, there has to be a protocol that guarantee for the reliable data transfer. The Alternating Bit protocol is implemented in the program to retransmit corrupted or lost data using FIFO. 

Repeate until receive FIN = 1 from Server

Repeate until the end of file

**Server**

**Client**

Socket()

Close()

Write()

Sendto()

Read()

Open()

Recvfrom()

Bind()

Recvfrom()

Open()

Sendto()

Rawinput()

Socket()

Close()

(Block util receive data)

Open file to read

Data Request

Data Reply

Flow-Chart

Timeout

Sendto(sendMsg)

start\_timer

Wait for call 0 from above

Wait for call 1 from above

Wait for ACK 1

Wait for ACK 0

Server

Sendto(data)

packet\_num = ‘0’

sendMsg = read(1024) + packet\_num

sendto(sendMsg)

start\_timer

Sendto(data)

packet\_num = ‘1’

sendMsg = read(1024) + packet\_num

sendto(sendMsg)

start\_timer

recvfrom(1024), ACK == 0

stoptimer

2. Code Frame for Client and Server Programs

This project will use Python 2.7 as framework to develop client – server application to transfer text file between client and server reliably using UDP protocol.

+ clientAlterBit.py: The source code for client side, that send the initial request for “text.txt” file to server, and when it receives the respond from server, it will check Seq number, and increase counter to 1 if the Seq number is the same as previous received package. If the Seq number is different, set counter = 0. When counter > 0, ignore data, send back to server the same ACK number as Seq, and wait for new data. When counter == 0, it will save the data into a text file in client side, “Alice in Wonderland.txt”, and send back the Acknowledgement number (ACK) to server to let server know it has received the message and wait for new data.

+ serverAlterBit.py: Server source code. The program will wait for request from client, send the data to client, set time out, and receive the ACK number from client. The sent data has sequence numbers alternate between 1 and 0. If the received ACK number is same as the previous sent data sequence number then flips the sequence number and send the next chunk of data with that sequence number. If the server has not received ACK number from client when timeout exception occurs, then retransmit data. If the server receives multiple ACK numbers because of premature timeout event, then only react to the first ACK and send data. Do nothing to other duplicate ACK numbers.

+ socket: Socket module is a python API or low-level networking interface used to build socket servers and clients.

+ sys: System-specific parameters and functions, which is a module that provides access to functions and parameters that can interact with kernel space to alter or see the information in kernel space, where the operating system resides.

+ time: Time is module in python that provides functions for working with time.

+ json: This module converts the in-memory Python objects to a serialized representation using jason.dumps() function, and convert back from a serialized representation to the in-memory Python objects using jason.loads() functions, which are similar to JavaScript Object Notation (JSON).

+ socket.socket(AF\_INET, SOCKET\_DGRAM): Create and return socket for IPv4 and using UDP transport protocol

+ socket.sendto(data[flags], addr): Send data to intended address through socket

+ socket.recvfrom(buflen [flag]): Receive incoming data into a buffer with predefined length. Return address and port of the remote host.

+ raw\_input([prompt]): The function reads a line from input, convert the line into string, and return it.

+ time.time(): Return time in second as a floating point numbers

+ socket.settimeout(value): Set a time to the system which is equal to value, starts to decrement the time, and throw socket.timeout execption when the countdown time reach 0.

+ open(name[, mode[, buffering]): Open a file and return object of file type

+ read(value): A function in file object, which is used to read a line of text.

+ write(buffering): A function in file object, which is used to write from buffer to the file object.