Requirement description

This document describes which files in the source code satisfy each of the specific requirement.

Protocol requirement:

1. Stateful:

1. Main flow

i) akh\_server\_main.c: this file contains the main function for implementing the DFA on the server side.

ii) akh\_client\_main.c: this file contains the main function for implementing the DFA on the client side.

(2) Connection

i) connection.c: defines functions that handle setting up connection between the client and server.

ii) connection.h: the header file

(3) file transfer:

i) file\_transmission.c: handles receiving and sending file segments

ii) file\_transmission.h: the header file

(4) disconnection:

i) disconnection.c: handles close connection and requesting retransmission of missing segments

ii) disconnection.h: the header file

2. Concurrent:

i) thread\_util.c:

Enable the server to support multi-client. Each client is assigned a server thread. These server threads are isolated from the socket. A monitor thread is responsible for socket receive operations. Original timer\_recvfrom() function is replaced with read\_with\_timeout. The communication between monitor and each server thread is done through fifo pipes.

ii) thread\_util.h: the header file

iii) hashmap.c: use hashmap to store the writing end of the pipes which are associated to each thread, such that monitor knows where to pass the incoming packet, and the right thread can receive the packet.

iv) hashmap.h: the header file

v) akh\_server\_main.c: m: implements the monitoring thread.

3. Service:

i) Makefile: in this file, we can specify a specific port number (default 9190) as the command arguments to the server.

ii) akh\_server\_main.c: in this file, we bind the server to the port number defined in Makefile.

4. Client:

i) Makefile: in this file, we can specify a specific port number (default 9190) and the host name of the server as the command arguments passed to the client.

ii) akh\_client\_main.c: in this file, the client can send messages to the host (IP) and port number defined in Makefile.

5. UI:

We use a command line UI. Some specific command line arguments can be set in the “Makefile”.

i) Makefile:

Command line arguments that can be set:

IP: IP address of the server

PORT: Port number

UFILENAME: the file to be uploaded (upload a file from client to server)

DFILENAME: the file to be downloaded (download a file from server to client)

Command Line examples:

To run the client (then nput arguments according to the instructions on screen):

$make run\_client

To run the client for downloading file:

$make run\_client\_download

To run the client the uploading file:

$make run\_client\_upload

To run the server:

$make run\_server