Networks assignment 2

# TCP over UDP sockets

Mazen Elmesery

12th December, 2019

# Server Part

## Overview

The server is on by creating a UDP socket for it, then bind it with some port number assigned by user, and starts listening for incoming datagrams on this socket.

Once there's a message on the socket, the server delegates it to a child process to handle it.

Each process is responsible for receiving the request from the client and determines the name of the requested file and starts to collect the data and store it into packets and sends it to the client

## Major methods and data structures

**handle_request(client_fd, client_addr, packet)**

This is the min method of the process which call the rest of the methods needed in order to deliver the data for the client. It starts by checking the file name packet sent by the client (checksum) and if it's valid it open the specified file and call **get_data** method to collect the data into **vector<vector<char>> data** which is the container for the data, then it calls **send_data** method which is responsible for creating and sending the packets to the client.

**get_data(file_name)**

This method opens the requested file and collect its data into some container **vector<vector<char>> data** where each element size of data is less than or equal to 500 Bytes.

**send_data(client_fd, client_addr, data)**

This method loops over the data vector and creates a packet for each element then tries to send it (with some PLP) and waits for an acknowledgment to send the next one (stop and wait).

# Client Part

## Overview

The client starts by sending a packet contains the name of the requested file, then waits for the data packets to come and send acks for them. The received data are stored in ordered map to be written into a file in order.

## Major methods and data structures

**create_data_packet(string file_name)**

This method creates a data packet contains the name of the requested file.

**send_file_name_packet(buffer)**

This method sends the file name packet to the server and waits for the data to come to ensure that the file name packet was sent successfully.

**extract_data(packet)**

This method extracts the data part from the packet and push it into ordered map.