جامعة خليفة
Khalifa University

Khalifa University of Science and Technology
Department of Electrical and Computer Engineering
ECCE 356: Computer Networks
Spring 2022

Deadline: Monday, April. 4, 2022 @12:00 Noon

Assignment 2: Simple Mail Transfer Protocol (SMTP)

Send Emails

Specifications and Requirements:

The goal of assignment 2 is to enhance the SMTP protocol started earlier. The student has to build on the SMTP implementation started in Assignment 1 where a message was constructed and shared with the mail server for validation. By the end of assignment 2, students should be able to modify the TCP multi-threading code, in order to implement most of SMTP functionalities including message construction and validation, clients' validation and email forwarding between clients through a mail server. Also, they should be able to differentiate between different mail server's reply codes and share them accordingly. The particular problem that you have to resolve consists of **transferring mail** between multiple clients **connected** to a server at **different** computers.

SMTP Background:

Simple Mail Protocol (SMTP) is a protocol that handles email transmissions between hosts. Client connects to a mail server to send an email, such that the message has the following header and body.

Header	To (email)	From (email)
	Subject	Time-Stamp
Body	Email Content	

The mail server verifies the TO and FROM fields of the header before pushing the email to the queue. If successfully verified, the server replies with a *"250 OK"* message. Otherwise, the reply is *"501 error"*. 501 error code implies that an emails header argument is not verified. Also, another code that the server can reply with is *"550 error"* which mean that the email address does not exist.

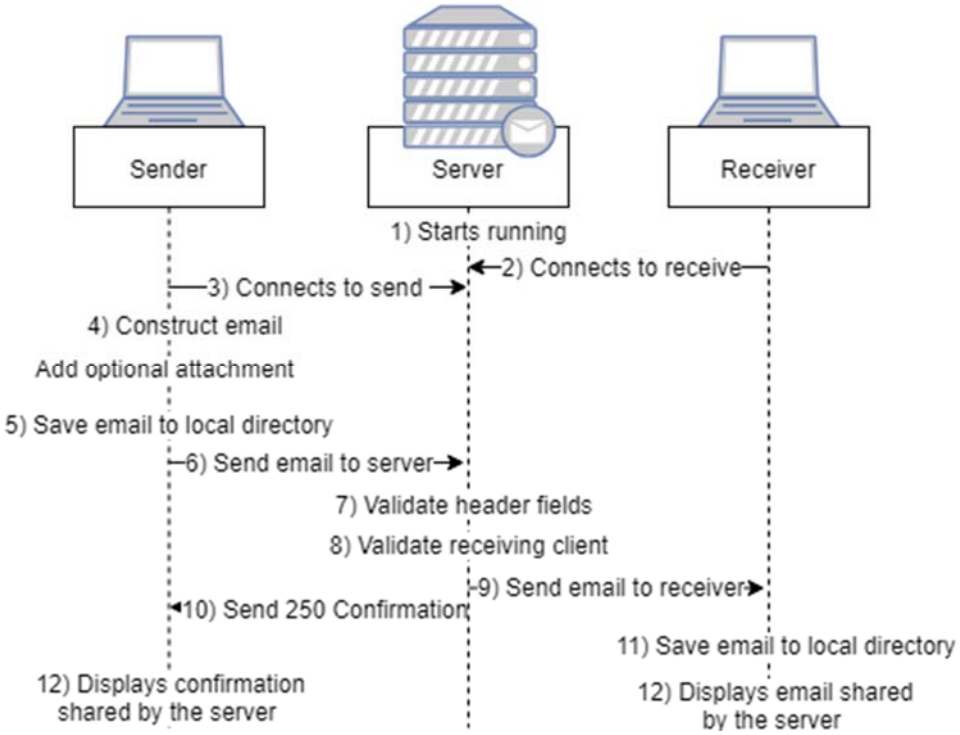
We have to do signup in thing??

Assignment 1 Functionalities:

In assignment 1, you were required to implement some SMTP protocol functionalities. As a first step, the client will create the message with its header and body and send it to the server. The server receives the message header, displays it and replies with a confirmation message including the time-stamp. The same functionalities will be extended on during this assignment as the following scenario indicates. Hence, you will use your assignment 1 codes to implement assignment 2.

Assignment 2 Scenario:

The below figure illustrates the scenario which you are required to implement in this assignment where a server acts as an intermediary between the sender and receiver clients.



The scenario steps are detailed as described below:

1. Server starts running while displaying the server hostname
2. Receiver Client connects to the server and listens to incoming emails
3. Sender Client connects to the server and receives a prompt to construct a new email
4. Sender constructs the email and is given the choice to add an attachment to the email (ex. Pdf, jpg, mp4, etc)
5. Once the email is constructed, it's saved to a file at the local sender directory
6. Sender sends email to Server
7. Server validate the header fields
    - Server **sends** error message with code *501 error* if email headers are not valid
8. Server has a mapping file to maps an email to a client host name, server validated the receiving clients by checking the (email, hostname) mapping in the mapping file:

    - If mapping does not exist, Server sends a reply message with code *550 error*
9. If mapping exist, Server sends the email to the receiving client
10. Server sends confirmation message to sender client with code *250 OK*
11. Receiving Client saves emails to files
12. Sender client and receiving client displays the reply messages and emails sent to them by the server (ex. Sender client displays to screen the server reply message "250 OK").

Assumptions and Specifications:

- The server that clients connect to can change its name. *→ (use ip ??)
- File format: Subject_Time.txt (Subject and Time change depending on the email sent, i.e. each email is saved in a separate file) (is it for both sender & receiver)
- If an attachment is sent along with an email it should be saved in the same directory as the email with the same name. However, the extension (.png, .pdf) will be different. => how different?
- *TO* and *FROM* fields are email format and **NOT** host name
- Sender Client can send to different email IDs. (Do not restrict a single To and From emails) *→ cc?
- There should be input check aside from the server validation to ensure that the input fields are valid. on sender & receiver
- All the existed email addresses are stored in the mapping file on the server
- Assume one Client Sends emails and another Client Receives emails.
- All running codes should display its status (ex. Client print to screen, "email has been sent to the server", whenever it sends the email)
- Server can only send emails to a **connected** receiving client

Supplementary Codes:

To develop the applications required in this assignment, please use the previously given programs in assignment 1 and the codes you developed in assignment 1.

Groups

For this assignment, you are allowed to strictly work in groups of 2 students unless you have requested otherwise and received an email confirmation regarding your group formation. Every member must explain his/her contribution to the assignment. Every group will be given 10 mins to demo and 10 mins to answer questions related to the assignment.

Deliverables

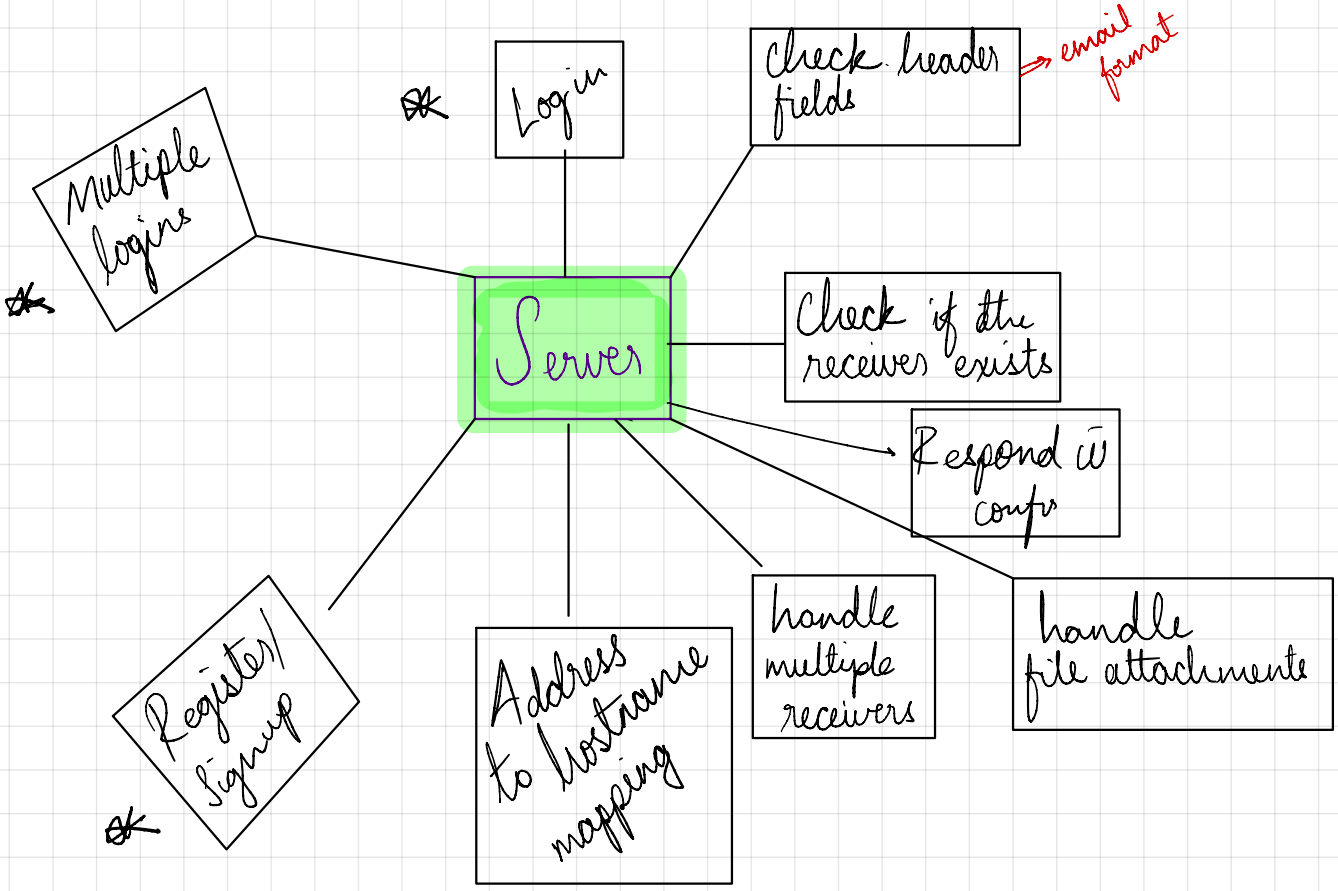
You will be required to submit the following by the deadline. Failure to submit by the deadline will result in deduction of marks.

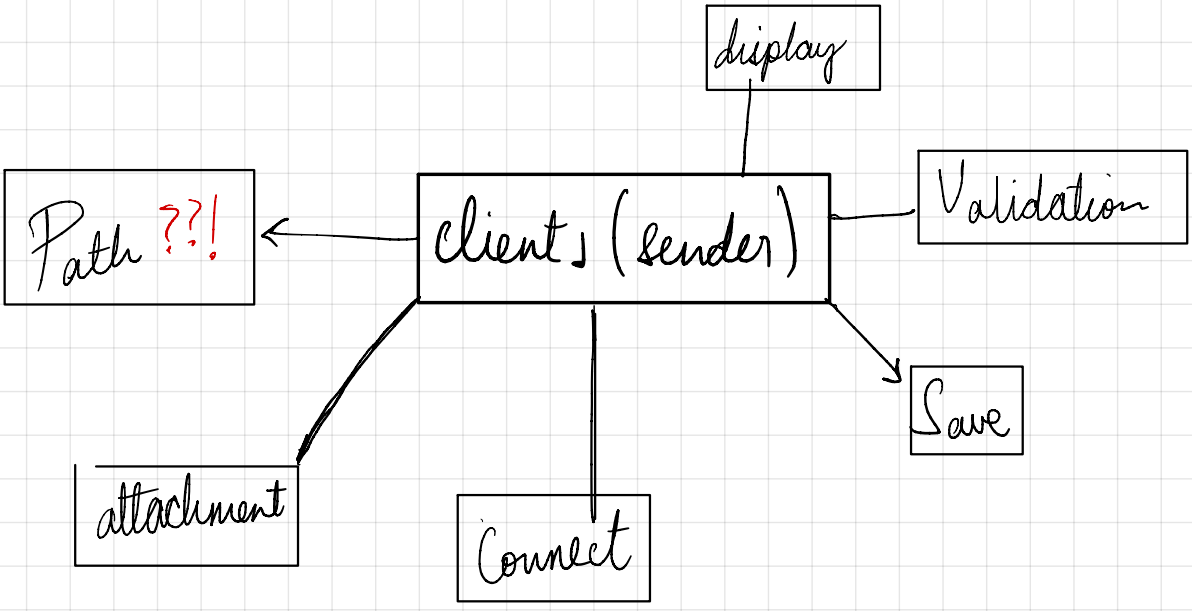
- Word/PDF document which contains your C/C++ program. The document should include all the file you created and modified (cpp and header files).

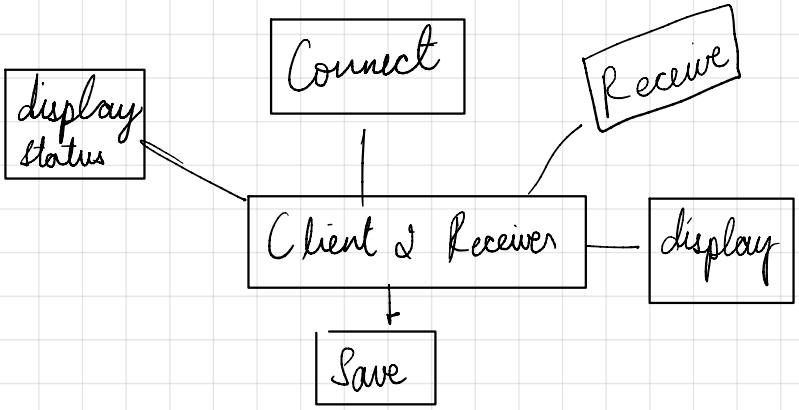
- A ZIP folder which includes the used projects (server, sender and receiver). You will be required to download the folder from Moodle in your demo slot and run it directly. Hence, make sure you submit a correctly running version.

Demo:

On demo day, each team should run the server and client codes on 2 or 3 PCs. one for the Server, one for Sender Client and the third is for the receiving client. One of the Clients running (Sender **OR** Receiver) on the same machine with the server. In addition, you will be asked questions about the functioning of the program; any student of the group must be able to answer any question. Part of the marks will be assigned for demonstrating compliance with requirements at this demonstration. Marks assigned to each member of the group may be different, depending on ability to answer the questions.


Server
Login
Multiple logins
Register/ Signup
check header fields
email format
Check if the receivers exists
Respond w/ confo
handle multiple receivers
handle file attachments
Address to hostname mapping


display
client, (sender)
Path ??!
Validation
Save
attachment
Connect



* use an array to store the sockets of connecting clients
* use status to identify as a sender or receiver ....