جامعة خليفة
Khalifa University

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

Deadline: Saturday, March. 3rd , 2022

Assignment 1: Simple Mail Transfer Protocol (SMTP)

Header Implementation

Specifications and requirements

The goal of this assignment is to acquaint you with the basics of network programming within the **TCP/IP** Internet family environment, and its socket application programming interface (API) and also a little bit of **multi-threading**. You will use the TCP protocol as a "delivery service", to carry your packets. The particular problem that you have to resolve consists of **transferring Mail** between **multiple clients and one server** logged in 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.

Task 1: Deadline: End of lab session (21st -23rd of Feb,2022)

- Run the sample code according to the provided instructions in the Assignment manual document and understand the code and its functions.
- The code and manual are uploaded to your lab sections

Task 2: Deadline: 3rd,March, 2022

In task 2, you have to implement part of the SMTP protocol. As a first step of SMTP protocol, you will implement the **SMTP send-message**. 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 server also has to verify the header's fields, for example verifying that the content of TO field is an email. If successfully verified, server reply with "250 OK", otherwise, "501 Error" is the reply message.

Client-Server Scenario:

1) **MAIL SERVER**: one partner logs at one computer and starts one application program to act as the mail server. At startup, program displays the following messages:

→ Mail Server Starting at host: [ServerHostName]
waiting to be contacted for transferring Mail...

2) MAIL CLIENT:

2.1) The other partner logs into the second computer and starts the other application program to act as the first client. At start up, program displays the following messages:

→Mail Client starting on host: [ClientHostName]
Type name of Mail server: **ServerHostName**
Creating New Email.

To: **Client2HostName**
From: **Client1HostName**
Subject: **Networks Lab**
Body: **Network Lab is Awesome..**

Mail Sent to Server, waiting...

3) **MAIL SERVER**: the server receives the mail, displays the Mail Header Fields:

Mail Received from [Client1Hostname]

FROM: [Client1HostName]

TO: [Client2HostName]

SUBJECT: Networks Lab

TIME: WED. Feb. 23, 2022 12:00

Network Lab is Awesome…

4) The server receives the email form Client 1, verifies the header's fields and sends back a confirmation message ("250 OK or "501 Error") with a time-stamp, and returns to the waiting state.

5) MAIL CLIENT:

The Client receives the confirmation message from the server displaying the following message:

Email received successfully at [TimeStamp]

6) **MAIL SERVER**:

You will have to enter control-C to kill the program.

Implementation Hint:

- The following figure shows the message structure, that you have to modify when modifying the message headers. Each variable is a message header's field and the buffer variable is the message's content. Be careful when you modify the *Msg* structure, you have to modify the *MSGHEADERSIZE* as well.

```
typedef struct
{
    Type type;
    int  length; //length of effective bytes in the buffer
    char buffer[BUFFER_LENGTH];
} Msg; //message format used for sending and receiving
```

Figure 1: Message Structure

```
#define HOSTNAME_LENGTH 20
#define RESP_LENGTH 40
#define FILENAME_LENGTH 20
#define REQUEST_PORT 5001
#define BUFFER_LENGTH 1024
#define TRACE 0
#define MSGHDRSIZE 8 //Message Header Size
```

Figure 2: Message Header Size Variable

Groups

For this assignment (and for the ones to come), a team of **two to three** is permitted, and no bonus is givenfor working alone. Every member must explain his/her contribution to the assignment.

Each team must be formed from the same lab section. Every groupwill be given 10 mins to demo and 10 mins to answer questions related to the assignment.

Deliverables

1. You are required to implement the Mail transfer, from the Client to the Server, and from the Server to the Client.
2. You must write your program in C/C++, and you must be prepared to demonstrate your program. Submit your assignment online to Blackboard by the deadline. The demonstration will be performed using the submitted version.
3. At the time of demonstration, 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.
4. Penalties will apply for late submission.