

IK2213 - Network Services and Internet-based Applications

Assignment 1 - Web mail

Ganapathy Raman Madanagopal, Tien Thanh Bui

1. Introduction

In this project, we have developed a webmail server for sending emails. The project mainly consist of two parts: a HTTP Server and a SMTP client.

The webmail server works as follows: on a user's request, the HTTP Server returns a simple web page containing a form by which the users can send emails. On the form, the user can specify the recipient address, the source address, the subject, the SMTP server address, the email content. The user can also specify the sending time for the email. Furthermore, the user does not really need to specify the SMTP server address, as it can be discovered based on the domain name of the recipient address. When the user submits the form, all the information will be sent back to the HTTP server. On reception of email information, the HTTP server forwards the information to a local SMTP client. The SMTP client then validates the information. If it is valid, the email will be sent on the specified time; otherwise, an error will be sent back to the user. The user can see a list of pending emails on a status page given by the HTTP server. A confirmation email will be sent back to the source address when the email is sent. The SMTP client supports UTF-8 and ISO-8859-15 character sets.

2. Problems & Our Solutions

As stated in the first section, the webmail server contains two main components: the HTTP server and the SMTP client. They were both implemented with Java. In this section, we will describe the problems we had to solve in those components and our solutions.

2.1. The HTTP Server

The HTTP server was implemented to listen and process HTTP requests from the users. A multithreading server was implemented to listen to requests on port 80. For each request from the user, a new thread will be created to handle the request. The server only processes three types of requests:

- **A HTTP GET request for the email form:** the HTTP server will return a HTML file containing the form to the user.
- **A HTTP POST request for submitting the information in the email form:** the HTTP server will parse the POST data from the request. The server then decodes the data using the ISO-8859-15 standard and sends it to the SMTP client. The SMTP client will consequently process the data to send the email.
- **A HTTP GET request for the status page containing a list of pending emails:** the HTTP server will get the list of pending emails from the SMTP client and returns it as a HTML web page to the user.

2.2. The SMTP Client

The problems that the SMTP client had to solve and our solutions are as follows:

- **Validate the information posted from the user:** the source and the recipient email address are validated by checking whether they are in the correct email format using regular expression. The delay time are validated by checking whether it is an integer and for negative numbers an error is thrown. If the SMTP server address is given, it will be looked up in the DNS records to check whether it exists or not. Otherwise, the SMTP client will look it up based on the recipient address. In that case, the information will be considered invalid if the SMTP server cannot be found.
- **Maintain a list of pending emails:** a list of email is created in the SMTP client. When a set of valid email information is sent to the SMTP client, it will create an Email object with the information and put it to the list. The email object will be removed from the list when it is sent.
- **Support emails in both English and Swedish:** the subject and the email content uses the ISO-8859-15 character set. ISO-8859-15 was chosen mainly because of its extensive support for western-european languages, support of € symbol and its popularity over recent times. For Multipurpose Internet Mail Extension (MIME) support, there are various encoding format available. For Content-Transfer-Encoding we chose Quoted-Printable encoding because it is fairly human readable and compact encoded results are obtained in compare to Base-64 and Binary formats. Any 8-bit value may be encoded using 3 characters, “=” followed by two hexadecimal characters. The length of the Quoted-Printable encoded line must not be greater than 76. If the length exceed the specified limit, then a soft line break consisting of “=” will be added at the end and it does not appear in the decoded text.
- **Connect to a SMTP server to send an email:** the SMTP client connects to the SMTP server on port 25 using socket. After successfully connecting to the SMTP server, a SMTP transaction with six command/reply sequences is performed: HELO, MAIL FROM, RCPT TO, DATA, Single dot (.) on a separate line and finally QUIT. The information for each sequence is generated from the email information. If all sequences are successfully performed, the email will be sent.
- **Schedule an email to be sent on the specified time given by the user:** for each email, the SMTP creates a timer that schedule to send the email after a given amount of time.
- **Send a confirmation email to the source address after sending an email:** when an email is sent, the SMTP client will try to send back a confirmation email to the source address. The SMTP server address of the source will be looked up in the DNS records, and it then be used to send the confirmation email to the source.

3. Discussion & Conclusion

Our webmail service performs basic email functionalities quite well. However, it is limited by the fact that we can only send emails within the ik2213.lab domain for now, because only SMTP is supported.

In order to send emails to other domains, like gmail.com, the SMTP connection needs to be secured with SSL or TLS. We will try to improve it in the future. However, although only basic functionalities were developed, this project has given us a solid background in some key areas, which will come handy in the future.