CSN-341: Computer Networks

Interactive POP3
Final Project Report

I. Members

Group-4

1. Aarush Gupta (16114002) aarush98@gmail.com
2. Aniruddha Mahapatra (16115021) aniruddha26398@gmail.com
3. Dakshit Agrawal (16114022) neelagrawal08@gmail.com
4. Deep Jethloja (16114023) deep.jethloja@gmail.com
5. Vipul Jain (16114071) jain.vipul2498@gmail.com

II. Aim

The objective of the project is to build a simplified version of POP3 protocol for educational purpose. We plan to demonstrate the basic POP3 functionalities listed below by client-server programs to simulate the interactions. We will also display the messages sent at the network layer, basically the IP packets with the IP headers and all the specific details that are actually involved in message communication. We also plan to create a beautiful User Interface to demonstrate these interactions.

III. POP3 Protocol

POP (Post Office Protocol) is an internet standard that defines an email server (the POP server) and a way to retrieve mail from it (using a POP client). POP3 is the most recent

version of the standard protocol for receiving email. This version includes mechanisms to expand the protocol for new actions and, for example, authentication mechanisms. It is built into the most popular e-mail products, such as Eudora and Outlook Express. It's also built into the Netscape and Microsoft Internet Explorer browsers.

POP3 is designed to delete mail on the server as soon as the user has downloaded it. However, some implementations allow users or an administrator to specify that mail is saved for some period of time. POP can be thought of as a "store-and-forward" service.

1. Working:

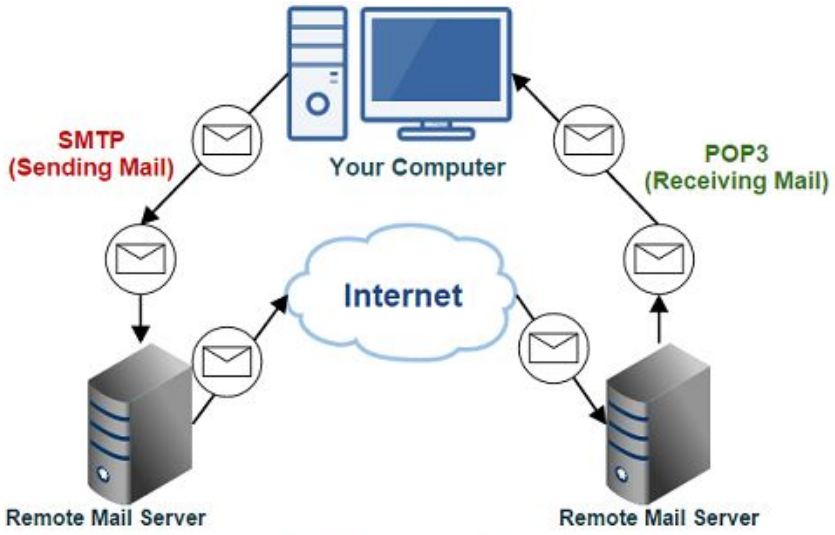
Incoming messages are stored at a POP server until the user logs in (using an email client) and downloads the messages to their computer. While SMTP is used to transfer email messages from server to server, POP is used to collect mail with an email client from a server.

2. POP Compared to IMAP:

POP is the older and much simpler standard. While IMAP allows for synchronization and online access, POP defines simple commands for mail retrieval. Messages are stored and dealt with locally on the computer or device alone. POP is, therefore, easier to implement and typically more reliable and stable.

3. Disadvantages:

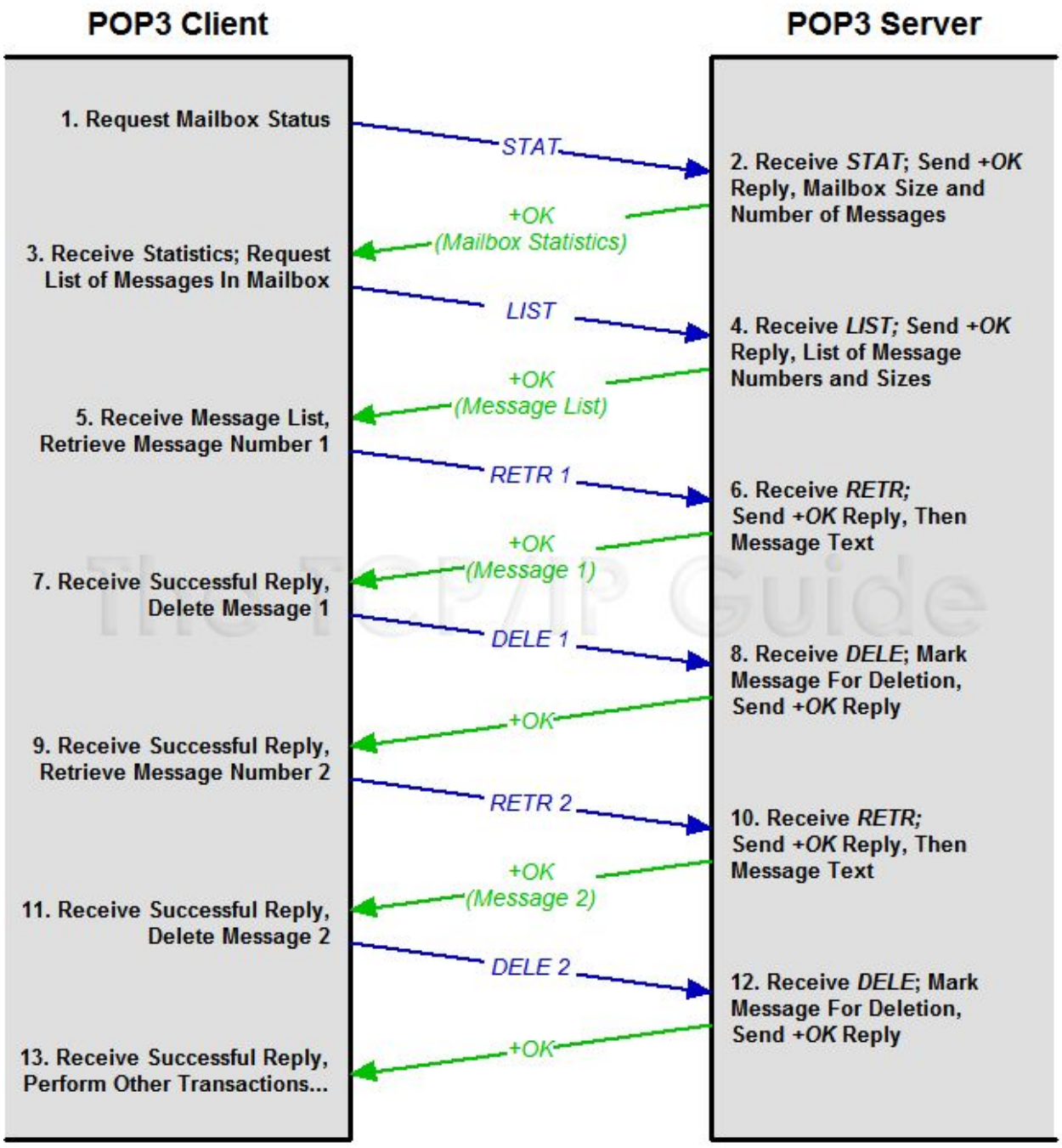
- POP is a limited protocol that lets your email program do nothing but download messages to the computer or device, with an option to keep a copy on the server for future download.
- While POP lets email programs keep track of which messages have been fetched already, sometimes this fails and messages may be downloaded again.
- With POP, it is not possible to access the same email account from multiple computers or devices and have actions synchronize between them.



POP Protocol Communication Process

III. FLOW DIAGRAM :

The below flow diagram demonstrates all the standard functionalities(commands) of POP3 protocol in particular order for obtaining a specific email from POP3 server by the POP3 client.


POP3 Client
POP3 Server
1. Request Mailbox Status
STAT
2. Receive STAT; Send +OK Reply, Mailbox Size and Number of Messages
+OK
(Mailbox Statistics)
3. Receive Statistics; Request List of Messages In Mailbox
LIST
4. Receive LIST; Send +OK Reply, List of Message Numbers and Sizes
+OK
(Message List)
5. Receive Message List, Retrieve Message Number 1
RETR 1
6. Receive RETR; Send +OK Reply, Then Message Text
+OK
(Message 1)
7. Receive Successful Reply, Delete Message 1
DELE 1
8. Receive DELE; Mark Message For Deletion, Send +OK Reply
+OK
9. Receive Successful Reply, Retrieve Message Number 2
RETR 2
10. Receive RETR; Send +OK Reply, Then Message Text
+OK
(Message 2)
11. Receive Successful Reply, Delete Message 2
DELE 2
12. Receive DELE; Mark Message For Deletion, Send +OK Reply
+OK
13. Receive Successful Reply, Perform Other Transactions...

IV. FUNCTIONALITIES :

1. USER command:

This is the first command a user may issue to the server. If a user's logon id is Dakshit, it should send:

```
USER Dakshit<CRLF>
```

Assuming there is a known user named Dakshit. So, the server sends an affirmative response to the client:

```
+OK Hello Dakshit- now send PASS<CRLF>
```

Or if Dakshit is unknown to the server, it may send:

```
-ERR Sorry Dakshit could you please introduce yourself ;)<CRLF>
```

2. PASS command:

After receiving a successful response of USER command, the client should send PASS command in the following format:

```
PASS secret<CRLF>
```

The server now checks its record and if it is satisfied, it sends:

```
+OK Dakshit- welcome and proceed<CRLF>
```

If password is wrong, the server sends -ERR response like:

```
-ERR password is wrong.<CRLF>
```

If password is correct, the server sets session state to transaction state.

3. STAT command:

This command is valid in the transaction state. On response server sends the following:

```
+OK nn mm
```

Here “nn” is number of messages and mm represents total size of all messages.

4. LIST command:

This command is issued to get the list of message information. It can optionally send message id following list command. In that case, information of that message is sent. Without message id, the server sends a positive response, then message id and message size in each line and finally sends a period:

```
+OK 3 messages
1 1229
2 15203
3 23105
.
```

The client can use the message id to reference the message to server.

5. RETR command:

This command followed by a (required) message id is used to get the entire message from the server. If the message id is valid, the server sends +OK on the first line. Then the entire message and then<CRLF>. <CRLF> sequence to indicate end of message. For a "RETR 1" command, a server may send the following response:

```
+OK 1229 octets<CRLF>
<Now server sends 1229 octets here><CRLF>
.<CRLF>
```

Or if the message id is not valid, the server sends an -ERR response:

```
-ERR Message not found.
```

6. DELE command:

The DELE command followed by message id requests the server to delete specified message in mailbox. On success, the server sends a +OK response. The message is marked as deleted and is deleted at the end of the session - when the session enters UPDATE state. User can undo the deletion of message if she/he wants before that.

7. QUIT command:

Last command from user. It closes the TCP channel and server sets the session to update state and deletes all messages that are marked as deleted.