International Conference on Network Applications, Protocols and Services 2008 (NetApps2008)
21 - 22 November 2008
Executive Development Center, Universiti Utara Malaysia

A New Approach to Enhance E-mail Performance through POP3 Protocol

Hussein Al Bazar[1], R. Sureswaran[1], O. Abouabdalla[1]
[1]National Advanced IPV6 Center
School of Computer Science
University Sains Malaysia
Penang, Malaysia.
{hussein, sures, omar}@nav6.org

Abstract- **E-mail system has become the most popular computer based method used to send and receive E-mail messages between the network clients inside or outside the local network. The most important point in E-mail system processes is to send and retrieve the E-mail message in the best delivery time without any delay or latency. POP3 (Post Office Protocol) is an E-mail protocol used to retrieve the E-mail messages from the E-mail server to the E-mail client. One of the most important issues is the E-mail server performance during the sending and retrieving process. In this paper, we will explain and show the latency locations in POP3 protocol which occurs during the E-mail messages retrieving process. We also proposed new client/server architecture to improve the current POP3 standard to enhance the E-mail server performance and to reduce the latency in retrieve time.**

Key Words: **- E-mail server, POP3 protocol, Latency, Client/server.**

I. INTRODUCTION

The POP3 protocol is used to retrieve the E-mail messages from the E-mail server to the E-mail client. Using the POP3 protocol; E-mail messages are carried out from the E-mail server and passed over through the network to the recipient's mailbox. These processes are done through client/server command dialogue. Once the E-mail messages are retrieved, it will be deleted from the E-mail server database using a specific command sent from the E-mail client [2, 6]. The POP3 client/server communication procedure is achieved through a set of steps; these steps can be divided into three main states (AUTHONTOCATION state, TRANSACTION state and UPDATE state). Each one of them has a specific function to accomplish. This procedure commences by establishing a communication session between the client and the server, and then proceeds by performing specific client's commands. This is followed by retrieving the E-mail message to the E-mail client. Upon the transportation of E-mail message to the client, this process comes to an end and the client proceeds by retrieving a new E-mail message or quitting the communication with the E-mail server [2, 4].

POP3 protocol provides the best delivery time compared to Simple Mail Transfer Protocol (SMTP) and X.400 protocol. This is due to the client/server handshake for each subsequence message and the security aspects in the SMTP protocol. Moreover, POP3 has been proved to be the fastest and most efficient protocol although its functionality is limited in the case of end-to-end connection which relates to the load on the client and the server, and the intermediate between them [4].

However, E-mail server performance in a real-life environment is limited due to the commands dialogue between the E-mail client, the E-mail server, and the E-mail server response time, which means that the largest latency occurs during the client/server commands dialogue and during the operations of transferring the E-mail messages data. Furthermore, neither the increase in the bandwidth, nor the increase of the E-mail server response time can reduce the latency time for the client/server procedure [1].

For POP3 protocol, the best delivery time and high performance are the core important points. Thus to achieve E-mail messages transportation without any delay or latency, the POP3 has to perform the best transportation functionality. As a conclusion, the enhancement of the POP3 protocol will lead to create a better E-mail environment with more efficiency and high E-mail server performance.

This paper is organized as follows: In sections **II** and **III**, we presented a full description of POP3 client/server procedure and showed the latency locations in POP3 protocol during the client/server procedure. In section IV we had discussed and proposed a possible architecture to enhance POP3 procedure in order to improve E-mail performance and to reduce latency delivery time. Finally we provided the conclusion of this paper.

II. POP3 CLIENT/SERVER PROCEDURE

The POP3 client/server procedure starts when the E-mail client has any incoming E-mail message and wants to retrieve it from the E-mail server. The procedure starts from the client by creating a new TCP connection between the client and the E-mail server on port 110. This connection is used to transfer the control commands and the E-mail message data to the POP3 E-mail client. The POP3 client uses a Domain Name System (DNS) to lookup the address of the E-mail server if it is known. Once the TCP connection is created, the client waits for an acknowledgment from the E-mail server that contains +OK greeting message which informs the client that the E-mail server is ready to start commands dialogue between them. The client now goes through the first session which is referred as "AUTHONTICATION State". In this state, the

client must identify himself to the POP3 E-mail server by using USER command which contains the user-name. If incase the user-name that was sent to the E-mail server is correct, which is a positive case, the server must respond using +OK command as an acknowledgment from the server side or send -ERR command. The next step in this state is to issue the password to the user-name which is done using PASS command. If the password is correct, which is a positive case, the server must respond using +OK command, or otherwise the server will respond with –ERR command and then the client must resend the password to the server. Once the USER and PASS commands are performed with positive responds from the POP3 server, the client has an access to an appropriate *maildrop*. The POP3 client now has entered into the "TRANSACTION State" and can start commands dialogue process.

The first command performed from the POP3 client in this state is STAT command. This command is used to ask the E-mail server if there are any new E-mail messages in the *maildrop*. In positive cases; the E-mail server responses with +OK command followed by information for the *maildrop*. Morever; the information sent from the POP3 server contains the number of E-mail messages in the *maildrop* and the message size of the *maildrop*. The next command performed from the POP3 client is LIST [msg] command, where it is used to ask the POP3 server about each E-mail message in the *maildrop*. Moreover, in the LIST command the argument [msg] is optional, where it is the message number which the client needs information about, and the command can be performed with or without the argument. In case the command was performed without any argument, the E-mail server responds by using +OK command followed by a line or multi-lines information about all messages in the *maildrop*. The information contains each message number and the message size. Furthermore, this command can be repeated more than once depending on the POP3 client's situation. If the client requires asking about specific E-mail message information, then the client will use the LIST command with argument; the server responds using +OK command and the information that is related to that message's number. The next command sent by POP3 client is RETR msg command, which is used to retrieve the E-mail message from the *maildrop*. The client sent RETR command with the message's number which requires to be retrieved to the client's mailbox. In positive cases, the POP3 server responds using the initial +OK command followed by a multi-line which is the E-mail message data. This command can be repeated many times, depending on the number of messages which the POP3 client wants to retrive. Once the retrieving process is accomplished, the POP3 client performs the DELE msg command which is used to mark the message as deleted. Moreover, if the client generates any future reference related to the message number, then the POP3 server responds using error message –ERR without deleting the message. This command can be repeated many times depending on the number of messages required to mark. After this command, the POP3 command will perform a NOOP command. In this case, the server must respond using the initial +OK command without taking any further

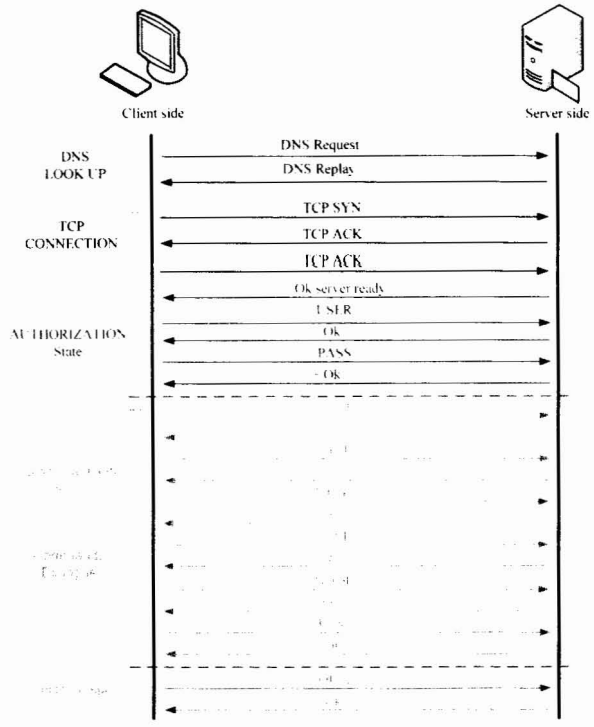

Fig. 1 POP3 Client/Server Procedure.

action. The last command in "TRANSACTION State" is RSET command, which is used by POP3 client to inform the E-mail server to unmark the selected E-mail messages by using the DELE command. In this case, the POP3 server. Again replies with +OK positive response. Figure 1 shows the illustration of the POP3 Client/Server procedure

The last state for the client/server procedure is the "UPDATE State"; where the POP3 client issues the QUIT command to exit from the "TRANSACTION State". Furthermore, this command must be performed in this state, otherwise the POP3 server will terminate the session without removing any message from the *maildrop*. In case the POP3 server removes all the E-mail messages in this state, then the server releases the *maildrop*, and closes the TCP connection.

As conclusion from Figure 1, the POP3 Client/server procedure, we can summarize that the minimal commands required to retrieve an E-mail message are as follows:

a. TCP connection establishment
Wait for connection on TCP port 110 in the server side.
Open connection by client.

b. *AUTHORIZATION State*
USER name valid in the AUTHORIZATION state
PASS string
QUIT

c. TRANSACTION State
STAT valid in the TRANSACTION state
LIST [msg]
RETR msg
DELE msg
NOOP
RSET

d. UPDATE State
QUIT

III. POP3 PROCEDURE LATENCY

The delay time is the time required to complete retrieving an E-mail message from the E-mail server to the E-mail client. E-mail latency is the main reason that directly affects the E-mail performance and the delivery time. By referring to Figure 1, the POP3 client/server procedure, which performs between the E-mail client and E-mail server, we find that the E-mail performance in high-speed network connections is limited due to the commands dialogue and POP3 server response time.

By analyzing the POP3 procedure, we can find the latency locations which directly affect the E-mail performance. Moreover; the first delay time comes from the greeting message which is sent from the E-mail server after the TCP connection is created. This latency time is shorter than the greeting time required in SMTP procedure. We will refer to this time using *Tgreeting*. The second location is the time required for the LIST command which strongly affects the E-mail receiving process while the POP3 client repeatedly polls the server to see if any new E-mail messages have arrived. If the size of an E-mail message is large, the latency comes from the POP3 server which replies to LIST commands. We will refer to this time using *Tlist*. The last latency location comes from the RETR commands which is the time required to send the E-mail message from the E-mail server to the E-mail client; we will refer to this time using *Tretr*. As a conclusion, the main reason leading to affect the E-mail server performance and delay time problem is the POP3 commands dialogue procedure. Figure 2 illustrates the POP3 client/server latency locations.

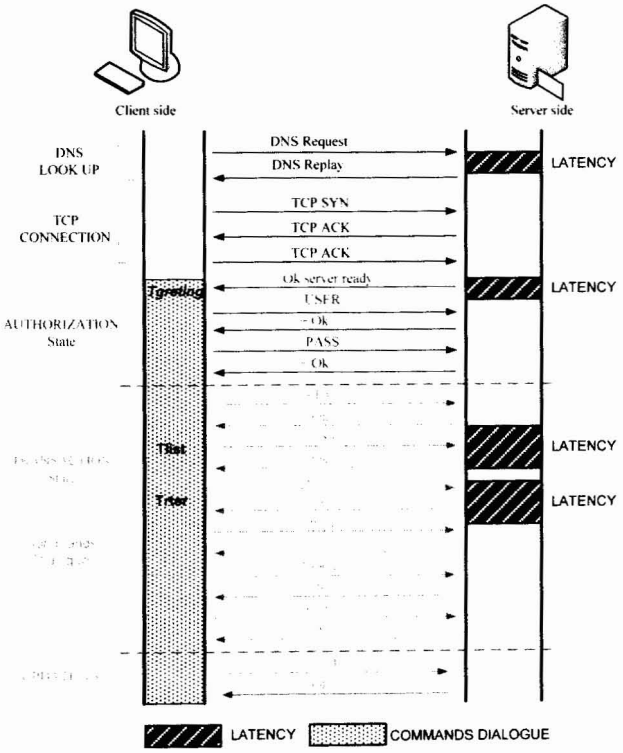

Fig. 2 POP3 Client/Server Latency Places.

IV. PROPOSED POP3 ENHANCEMENT

In this section we propose a new procedure for POP3 client/server to improve the current procedure and enhance POP3 protocol, and also to reduce the latency time. The first step is by cancelling the +OK greeting message, which comes before sending the USER and PASS commands in AUTHONTICATION State. This message is sent from the E-mail server to the E-mail client to inform the client that the server is ready to start commands dialogue process. Moreover, for the POP3 client to access an appropriate *maildrop*, we proposed to use APOP command rather than sending USER and PASS commands. APOP command is used as an alternative method of authentication but does not involve sending a password in a plaintext over the network. The POP3 client performs APOP command after it receives the +OK greeting message, which contains a fully-qualified domain-name corresponding to the host where the POP3 server is running, and the server verifies the digits which are sent with the APOP command. If the digits are correct, then the client/server procedure enters the "TRANSACTION State". We have to propose setting-up the TCP connection by using the T/TCP (Translation TCP), which is an extension of the TCP protocol to provide transaction oriented services, and to allow the transfer of data segments during the processes of connection establishment which is the standard TCP three-way handshake. However, in this session, the client's first packet will contain the SYN -bit-, the requested data itself, and the FIN -bit-, where the SYN is used to establish the connection with the E-mail server. The request is the data which is sent from the client to the E-mail server "the data digits" will be in the same first packet. The FIN is used to release the connection once the transaction process of the E-mail message is finished. For additional illustration, we can write the T/TCP process as follows: the client requests to establish a connection, the authentication process digits are sent, and once the client finishes, the client will inform the server that the client is done. After the E-mail server receives the client's request and verifies the data digits, it sends an acknowledgment which contains the greeting message and the permission to access the *maildrop* to the client. The client receives the server's acknowledgment, and sends –SYN, FIN acknowledgement. Thus, the client directly enters the TRANCACTION State.

In this case, the total time needed for creating TCP connection transaction process for the greeting message connection and the AUTHORIZATION state will be reduced. All the processes will be accomplished during the creation of T/TCP connection and the session will enter to the TRANSACTION state.

On the other hand, the transaction commands' dialogue will be implemented using the pipelining approach. The pipelining is used to minimize the sessions' time which is opened for the transaction procedure, and to improve the performance of delivery time with any E-mail server built to support this approach of pipelining.

We will use the pipelining in the TRANSACTION state by dividing the commands into two sessions. The first

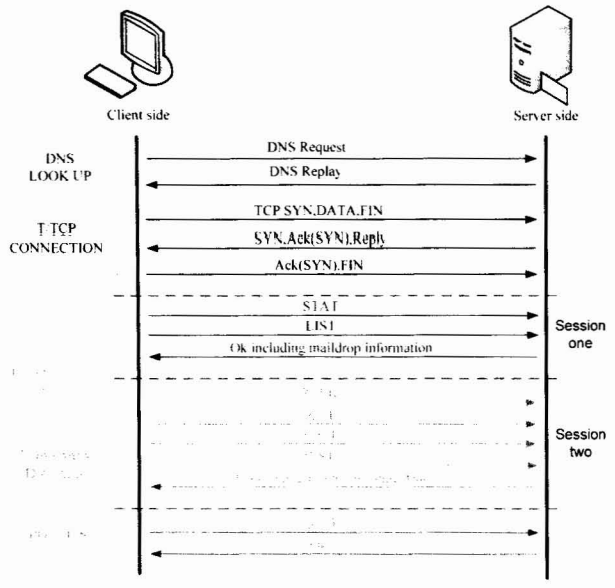

Fig 3. The Proposed Client/Server Procedure .

session will be for the commands which are STAT and LIST where both commands will be sent sequentially as a group without waiting an +OK replies for each command. Thus, the client will have to wait for one acknowledgment which contains the information for any new E-mail message arriving, and information on the number of the E-mail messages and their size. Additionally, a special field will be added to the +OK acknowledgment to inform the client about the error's location if there is any, during the sending process of the commands.

The second session will be for the commands which are RETR, DELE, NOOP, and REST. All these commands will be sent sequentially as a group without waiting an +OK reply for each command. The server, in the positive case will send the multi-line E-mail message and any other information related to that group of command. Also, a special field will be added to the +OK reply to inform the client about the error's location, if there is any, during the sending process of all commands, and the number of +OK replies will be reduced to one +OK acknowledgment only. These steps will enhance the total time required for the commands dialogue procedure. Figure 3 shows the proposed client/server procedure.

As a conclusion for the pipelining approach procedure, the total number of the +OK E-mail server replies will be reduced to three +OK messages, where in the standard POP3 procedure the total number of +OK messages is ten. Thus, we proposed a new POP3 procedure to reduce the total commands dialogue time and to enhance the E-mail server performance.

V. CONCLUSION AND FUTURE WORK

POP3 protocol is used to retrieve the E-mail messages from the E-mail server to the client, the latency which occurs in several places in the POP3 protocol, are referred to the E-mail greeting message and commands dialogue between the clients and the E-mail server.

To improve POP3 protocol and the delivery time, we proposed an enhancement for the POP3 client/server procedure by cancelling the E-mail server greeting message and using the T/TCP and APOP command during the process of creating the TCP connection. A modified pipelining approach in the POP3 procedure is also proposed as a step to reduce the time required to retrieve the E-mail message from the E-mail server and improve the E-mail performance. For the future work, a real-life simulation for the proposed procedure will be implemented and a comparison with the existing procedure will be conducted to examine the efficiency and the possible modifications.

REFERENCES

[1] J.Charzinski "Observations in E-Mail Performance" ITC Specialist Seminar, Würzburg, Germany, Jul. 2002, pp. 133-142

[2] J. Myers, M. Rose "Post Office Protocol- version 3" RFC 1939, May 1996.

[3] M. H. Haggag "ENHANCED DELIVERY THROUGH A SMART SMTP CLIENT".The International Journal on Intelligent Cooperative Information, VOL. 4, NO. 1, January 2004.

[4] P. Tzerefos, C. Smythe, I. Stergiou, and S. Cvetkovic, "A Comparative Study of Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP) and X.400 Electronic Mail Protocols", Proceedings of the IEEE 1997 22nd Conference on Local Computer Networks - LCN, pp 545-554, 1997.

[5] R.Braden "Extending TCP for Transactions – Concepts",FRC 1379, Novemver 1992.

[6] R. Gellens, C. Newman, L. Lundblade "POP3 Extension Mechanism" RFC 2449 November 1998.