

A New Approach to Enhance E-mail Performance Through SMTP Protocol

Hussein A.Al bazar[†], Sureswaran Ramadass^{††}, and Omar Abuabdalla^{†††}

[†]National Advanced IPv6 Center, School of Computer Science, University Siance Malaya, Penang, Malaysia

^{††}National Advanced IPv6 Center, School of Computer Science, University Siance Malaya, Penang, Malaysia

^{†††}National Advanced IPv6 Center, School of Computer Science, University Siance Malaya, Penang, Malaysia

Summary

E-mail usage has become the most popular computer based method on the internet. The primary goal in E-mail processes is to decrease E-mail delivery time which is the key important point in order to improve E-mail performance for the duration of sending an E-mail message. SMTP (Simple Mail Transfer Protocol) is the E-mail transport protocol used to transfer E-mail messages to and from the mail server and used to transfer messages between the local hosts in the local network. This paper will explain and show the latency places in SMTP protocol which occurs during the SMTP client/server communication procedure and propose a new architecture for the client/server procedure to improve the current SMTP standards which will enhance E-mail performance and reduce the latency in delivery time.

Key words:

E-Mail Server, SMTP, Latency, Client/Server.

1. Introduction

The SMTP transport protocol used to transport the E-mail messages from the client to the E-mail server and from the E-mail server to the destination address. Using the SMTP protocol; the messages are carried out from the sender and are passed over through the local network to the sender's E-mail server, then it starts routing over the internet network till arrives to the recipient's E-mail server [6]. The SMTP transportations step outlines the E-mail transportation from the sender to the recipient, and forces reliability, efficiency and provides quick services [4]. The SMTP client/server communication procedure is achieved via a set of steps; each of them is responsible for accomplishing specific functions. This procedure commences with establishing a communication session between the client and the server, proceed by performing specific client commands, followed by transferring the E-mail message to the SMTP E-mail sever, upon the transportation of E-mail message; the client proceeds by resending a new E-mail message or quitting the communication with the E-mail server [4,5,6]. For SMTP

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

The paper is organized as follows. In the next section we review the related work on SMTP protocol. In section 3 and 4, we present a full description of SMTP client/server procedure, and show the latency places in SMTP protocol during the client/server procedure. Sections 5 present the contribution of this paper which is to enhance SMTP procedure in order to improve E-mail performance and reduce latency delivery time. Section 6 will provide the conclusion of this paper.

2. Related Work

This section review some of previous work related to the SMTP protocol, and the characteristics of its performance. P.Tzerefos et al. have done a comparison between SMTP protocol, Post Office Protocol (POP3), and X.400 Electronic Mail Protocol. In this work the SMTP shows relatively high weak-performance due to SMTP client/server handshake for each subsequent message, as a conclusion, the POP3 protocol has been proved to be more efficient than SMTP protocol; one reason behind this, is that the SMTP does not provide security aspects for sender authentication and does not guarantee E-mail message delivery, SMTP just allows to verify the existence of the recipient of the E-mail message, in this case, an error message will be returned to the sender to resubmit the E-mail message. Moreover, the SMTP protocol produce the highest delay in end-to-end connections.[7]. Joachim Charzinski, presented an observation of protocol performance for E-mail transfer protocol in real-life environment. This work has evaluated the traffic characteristics, and the performance measurement for SMTP, POP3 and IMAP. Multiple traces has been used which through using it showed that SMTP client/server

performance is limited, that is due to the commands dialogue and E-mail server response time, which means that the largest latency in SMTP protocol occurs during the commands dialogue between the clients and the E-mail server, and during the operation of transferring the E-mail message data. The result shows that neither the increase in the bandwidth nor, the increase of the E-mail server response time can reduce the latency time for the client's requests. [3].

In Saket Kaushik et al. work; the SMTP recipients are given permission to control the recipient of the E-mail messages, and the connection with the sender by proposing three new type of policies, which are the Service Level Agreement Policy (SLAP), Message Scheduling Policy (MSP), and Mailbox Resource Allocation Policy (MRAP). SLAP addresses how the receiver of an E-mail service provider interacts with a given sending E-mail service provider. MSP -which is the output from SLAP evaluation- shows how to deal with each message in the sending E-mail service provider. MRAP is used to determine how the messages are presented to the actual human recipient. Moreover this work have proposed an extension to the SMTP protocol architecture by extending the handshake procedure and adding SHLO command, and add extra headers for the policy support which are X-Attributes, X-HdrHash, and X-Hash. However, all of the previous proposed extension of the SMTP protocol leads to increase the processes and expand the commands dialogue which takes place before the E-mail message transfer to the recipient E-mail address.[5]

M. H. Haggag. Enhance the E-mail delivery by proposing smart SMTP protocol which force reliability, highly performance, and aims to minimize the delivery time by cutting the time consumed at the queuing phase through processing all the messages belonging to the same E-mail server as a bulk. The proposed SMTP contains two smart agents which are Mail Transfer Agent (MTA) and Delayed Delivery Agent (DDA), these agents use some other components to improve and enhance the delivery of the E-mail message. The threading controller takes the responsibility of creating and controlling a group of threads, the grouping generator groups the messages by their destination address, and store the groups according to their priorities. The commands manager is used to enable the communication with the destination E-mail server and verifies the state of each command. The messages manager in its turn is used to read, move, copy and delete the messages. Finally, the communication controller is the one responsible to control the network communication. [6]. Alka Nand et al. proposed a compression extension and insert it into the E-mail system to enable embedded data compression mechanisms; by doing this, the SMTP and POP3 protocol efficiency can be raised and improved.

Using the embedded data compression lead to reduce the average time required to send an E-mail message but the number of processes increase before starting transaction, because it require using compression algorithm before starting sending E-mail message. [1].

3. SMTP Client/Server Procedure

The SMTP client/server procedure starts when a client in a local network wants to send an E-mail message to another address outside the local network using the SMTP E-mail server, or when the messages are transferred between the hosts in the same network. The procedure starts with the client by creating a new TCP connection between the client and the E-mail server on port 25. This connection is used to transfer the control commands and the E-mail data to the SMTP E-mail server. The SMTP client uses a Domain Name System (DNS) to lookup if the address of the E-mail server is known. After the TCP connection is created; the client waits to receive an acknowledgment from the E-mail server which contains a 220 replay greeting message. During this stage the sender cannot send any commands until receiving the acknowledgment, at that time; the client reply back to the acknowledgment by sending a greeting message using HELO or EHLO command. This greeting message contains the client's identity. The server confirms using 250 (OK) reply codes. The next three steps for the E-mail transactions start from SMTP client sending MAIL command to the E-mail server, which initiate a mail transaction and indicates the E-mail return-address, and the E-mail server confirms using 250 (OK) reply codes. The previous step is followed by sending RCPT command from the client to identify an individual recipient of the E-mail data. The same operation is repeated by the server, both of them are 250 (OK) reply codes, -note that this operation can be repeated many times if the E-mail is being sent to multiple recipients-. To start sending an E-mail message, the SMTP client sends DATA command and the E-mail server confirms by sending 354 reply codes, which indicate the beginning of the data transaction from the client to the SMTP E-mail server. When all the E-mail data are sent to the SMTP E-mail server, the client notifies the end of the E-mail data by sending a line containing only a special character ".". The server confirms by sending 250 (OK) reply codes. The client might repeat the same process if it has another E-mail to be sent to the server using the same TCP connection. If no more E-mails to be send to the E-mail server, the connection will be closed by sending QUIT command to the E-mail server, in this case the E-mail server must confirm this command also by 221 (OK) reply codes.

4. SMTP Latency

The delay time is the time slot required to complete sending an E-mail message from the client to the E-mail server. This slot begins with the initiation of TCP connection up to the delivery of the E-mail message to the SMTP E-mail server. E-mail latency is the main reason that directly affects the E-mail performance, and the delivery time. Figure1 shows the latency in the SMTP client/server procedure.

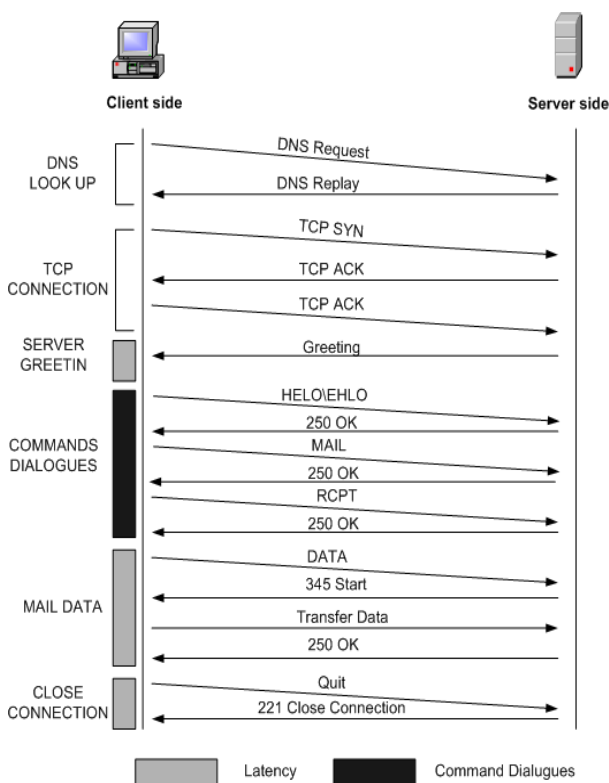


Fig. 1 The client/server procedure and the latency places in SMTP protocol.

By looking at the SMTP client/server procedure –which performs between the client and the server- We find that the E-mail performance in high-speed network connections is limited with the commands dialogues and the SMTP E-mail server respond time. Where the commands dialogue procedure leads to delay the E-mail transaction process. Moreover, the SMTP client must have a time-out for each of the commands, such time-out is used by the client in the interval between each pair of the commands mentioned in section three, and this time is exactly 5 minutes in the worst case, while for the DATA command and TCP completion it requires 15 minutes in the worst case. as well.

When the SMTP client fails to send an E-mail message, the client will have to wait 30 minutes before resending the same E-mail message to the same SMTP E-mail server. As a conclusion in a large network such as a residential network or an office network which contain a large number of clients (more than 100 clients), SMTP protocol have latencies in the communications which are referred to the server greeting time, the commands dialogues, and the time required to transfer the data of the E-mail message.

5. The Proposed SMTP Enhancement

In this, section we propose a new procedure for the client/server procedure to improve and enhance the SMTP protocol, and reduce the delay time. The first step is to cancel both of the greeting messages from the E-mail server side and the HELO or EHLO command from the client side. Since the greeting message is used to identify the E-mail server software and version information to the client and vice versa, we have to propose setting up the identification during the initiation of the TCP connection by using the T/TCP (Translation TCP), which is an extension of the TCP protocol to provide transaction-oriented services, and allow the transfer of data segments during the processes of connection establishment –the standard TCP three-way handshake-. However, in this case the client's first packet will contain the SYN -bit-, the request 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. Hence, the (HELO, EHLO) command information, -which should be sent from the client to the E-mail server to identify the client- will be in the same first packet. The FIN is used to release the connection after the transaction process of the E-mail message is finished. For more illustration; we can write the first client's packet's-information as the client tries to inform the E-mail server with the following: (I want to establish a connection, here is the data, and after the client finishes; informs the server that I am done). After the E-mail server receives the client's request, it sends an acknowledgment to the client containing the greeting message information to identify the E-mail server by the client. The client receives the server's acknowledgment, and sends –SYN, FIN- acknowledgement.

In this case, the transaction process for initiation 220 greeting message connection, will reduce the total time-out by 5 minutes which is the time that the client have to wait in the worst case of delay from the server until the client receive the greeting message from the E-mail server ,and starts the commands dialogue procedure.

In the other hand, the transaction commands will be implemented using the pipelining -an extension for SMTP protocol standards-, the pipelining is used to minimize the

time sessions which is opened for the transaction procedure, and to improve the performance of delivery with any E-mail server built to support this approach of pipelining.

We will use the pipelining with the data transaction commands (MAIL, RCPT, DATA) by sending all the commands sequentially as a group without waiting an acknowledgment for each command. Thus, the client will have to wait for one acknowledgment only for the three commands, which includes a special field to inform the client about the acceptance to start sending the E-mail message's data, and another field to confirm that all of the commands were sent correctly -no error in the E-mail server side-. Furthermore, a special field will be added to the server acknowledgment to inform the client about the error's location -in any of the three commands- if errors occur during the sending process of the transaction commands. Figure 2 shows the new client/server procedure.

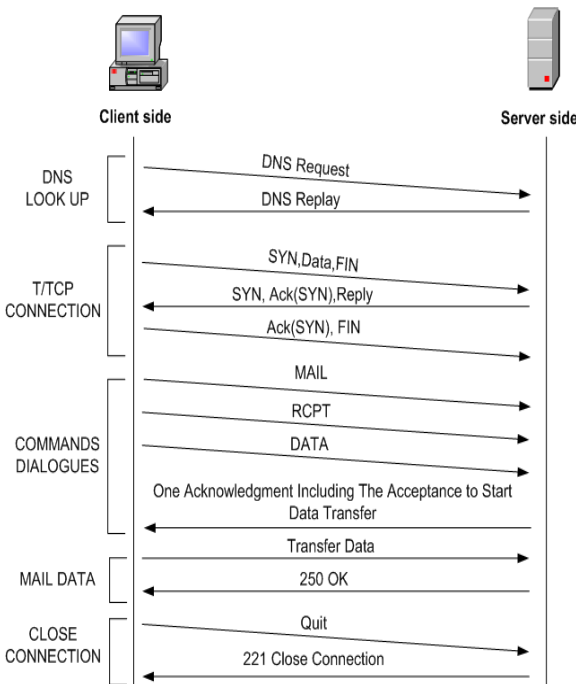


Fig. 2 The proposed client/ server procedure

In the pipelining approach the client message transfer agent has to wait 10 seconds for each acknowledgment, and in the failure situation; the time-out increases to 5 minutes for each command, furthermore, the server will send an acknowledgment for each command (MAIL, RCPT, DATA), this means that in case of failure situation, the total time-out for the commands is 15 minutes.

-as 5 minutes per each acknowledgment-. However, in the proposed pipelining approach; the E-mail server sends

only one acknowledgment for all three commands, where this will lead to reduce the time required for the command dialogue. Moreover, in the failure situation, the client waits at least 5 minutes between each command to receive an acknowledgment. By using this proposed enhancement, the client will wait only 5 minutes for resending the commands, where the commands is a group not a single, and the client will receive one acknowledgment for all the transaction commands.

Table1 shows the different time-out between the proposed pipelining, and the current pipelining approach regarding the total time-out in the failure situation, and the best case situation, which is required for each command at the client side before starting transaction of the E-mail message's data.

Table 1. Pipelining Approach

CMD	Current Pipelining Approach		Proposed Pipelining Approach	
	Worst case	Best case	Worst case	Best case
MAIL	5 m	10 s	5 m	10 s
RCPT	5 m	10 s		
DATA	5 m	10 s		
Total Timeout	15 m	30 s	5 m	10 s

By looking at table1; in the failure situation -worst case-, we can see that in the current pipelining approach; the total time-out is 15 minutes per commands (MAIL, RCPT, DATA), which means that the client have to wait 15 minutes before it starts transferring the E-mail message's data. In the other hand, the proposed pipelining Approach reduces the time-out by 10 minutes by waiting only for one acknowledgment from the E-mail server's side; in this case the total time the client has to wait is 5 minutes before starting to transfer the E-mail message's data. However, in the high performance response time -as we consider that the time-out is at least 10 seconds per command-, we can see that in the current pipelining approach; the total time-out for the three commands mentioned above is 30 seconds, which is the time that the client has to wait before starting to transfer the E-mail message's data. In the other hand, the proposed pipelining approach reduces this time-out by 20 seconds, where the total time the client has to wait is 10 seconds before it starts transferring the E-mail data.

$$\sum(T.CMD + T.DATA) \quad (1)$$

Where, (T.CMD) is the time per commands dialogue which includes (initial 220 message, MAIL command, RCPT command, DATA Initiation, DATA Block). (T.DATA) is the data termination time-processing to deliver a message to the recipient's mailbox.

Here we will show the calculating of the total time for both the proposed SMTP procedure and the current SMTP procedure in the case of delay responding from the E-mail server side.

Suppose we are looking to send message using the current SMTP protocol, in the worst case of the E-mail server delay of the response time, the total time required to send 5 messages is calculated as:

$$((5 + 5 + 5 + 3 + 2) + (10)) = 30 \text{ minutes.}$$

Similarly; to send the message using the proposed SMTP, the Total time required is calculated as:

$$((5) + (10)) = 15 \text{ minutes}$$

Where by referring to table1, we can see why we need only 5 minutes to the (T.CMD).

As a conclusion the proposed SMTP protocol will reduce the total time by 50%.

6. Conclusion

Sending the E-mail and its delivery time are the most important issues in the E-mail Performance. SMTP transaction protocol is used to transfer the E-mail message from the client to the E-mail server, the latency -which occurs in several places in the SMTP protocol- are referred to greeting messages and commands dialogue between the clients, and the E-mail server.

To improve SMTP standards and delivery time we proposed an enhancement to the SMTP client/server procedure by canceling the two greeting message commands dialogue using the T/TCP protocol, and use a modified pipelining approach in the SMTP procedure, as a step to reduce the time required to transfer the E-mail message from the client to the E-mail server.

References

[1] A. Nand, T. Lai Yu, "Mail Servers with Embedded Data Compression Mechanisms", IEEE Computer Society, Proceedings of the Conference on Data Compression, 1998, ISSN:1068-0314

[2] B.Dekens "Relations Between In- and Outbound Email Traffic", Proceeding of 4th Twente Student Conference on IT, Enschede, January, 2006.

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

[4] J. Klensin, "Simple Mail Transfer Protocol", RFC 2821, April 2001.

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

[6] 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.

[7] R.Braden " Extending TCP for Transactions—Concepts", RFC 1379, November 1992.

[8] Saket Kaushik, Paul Amman, Duminda Wijesekera, William Winsborough, Ronald Ritchey, "A Policy Driven Approach to Email Services, Fifth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'04), 2004. policy, p. 169-178.