Efficient E-mail Management System with Bandwidth Reduction and 3-Level Message Prioritization

O. Said
Department of Computer Science, College of Science, Menoufia University, Egypt
Department of Computer Science, College of Computers and Information Technology,
Taif University, KSA

ABSTRACT

In this paper, a new e-mail management system is demonstrated in which all messages are prioritized. Each type of priority message is handled with a special method to enhance the efficiency of the proposed e-mail system. Also, a new management technique for user inbox is introduced. In addition, a minor upgrade in TCP (Transmission Control Protocol) message confirmation (ACK) is demonstrated, to reduce the bandwidth consumption. Finally, the network simulator ns-2 is used to evaluate the new proposed e-mail management system. The obtained results show that the proposed system outperforms the traditional ones from points of view of bandwidth, overall system efficiency, packet loss, and delay.

Keywords

Internet Protocols, E-mail System, SMTP, POP3, UDP, TCP.

1. INTRODUCTION

Sending and receiving of emails is one of the most important and useful processes of the Internet. If it is free to send e-mail, it also free to send millions e-mails. This makes the email an ideal medium of abuse. There are approximately 50% of the sent e-mails are spam or trivial messages. It is well known that the spam messages consume a huge bandwidth that may be needed at internet bottleneck states. In addition, important messages may be received in spam or bulk and this may cause a message deletion without being seen. So, email is important, and spam affects all users and networks. The famous treating for the above problems is to make a block or a spam filtering. These trials can’t be considered a long term solution, because the spam filtering makes the filtered messages travel through the networks and stop at the e-mail box. So, the message overload still affects the message trip networks. The address block solution forbids the urgent and trivial messages to be received. So, the solution is to upgrade the management system of traditional e-mail such that; the sender can determine the degree of message importance that will be sent, and the receiver can determine the importance of sender e-mail address that the message comes from. This technique is called message prioritization [1], [2], [3], [4], [5], [6], [7].

This paper proceeds as follows; the traditional e-mail system analysis is demonstrated at Section 2. The related work is discussed at Section 3. The new management e-mail system with detailed flowcharts to describe its processes is demonstrated at Section 4. The TCP upgrades are demonstrated at Section 5. Algorithms of the new proposed system are demonstrated at Section 6. Enhancements of the new e-mail system are shown at Section 7. Real test environment with some output screens is introduced at Section 8. Finally, the simulation setup and the results are discussed at Section 9. Conclusion and future work terminate the paper.

2. TRADITIONAL E-MAIL SYSTEM

This section gives a brief overview of the basic idea of the e-mail system [8] and how the e-mail server operates.

2.1 Basic idea

When you write an e-mail message details and send it, it goes on internet trip. The e-mail message trip can be stated as follows: the access provider's network encapsulates the message in an electronic "envelope" like a real envelope, and sends it out. The Internet protocol that is used here is called SMTP (Simple Mail Transport Protocol), which is a member of the TCP/IP protocol suite. Hence, the message finds its way to the recipient's network, where an e-mail computer routes it to the person's mailbox [8], [9].

2.2 E-mail server processing

This sub-section aims to give a brief overview of how the email server works. Email servers are complicated servers to set up because there is mass number of security implications to run your personal mail server. The MTA (Mail Transport Agent) has the mainly security issues. Many modern MTAs make it quite difficult to configure an open relay. The MDA (Mail Delivery Agent) is the next step. The MDA function is delivering the email to the user. This means that the MDA places the email in mail-box or mail-dir and accomplishes a mail filtering process [8], [9]. The traditional e-mail system components and protocols are showed at Fig. 1.

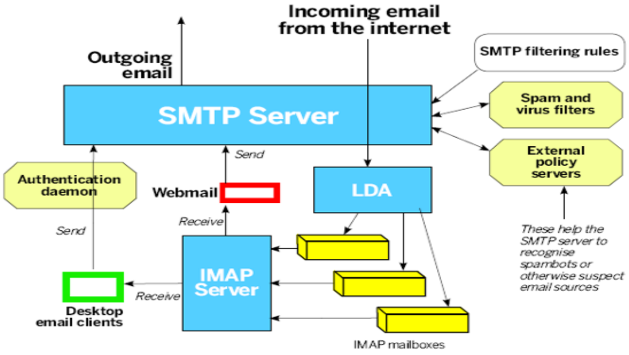


Fig. 1: The traditional e-mail system components.

3. RELATED WORK

In this section, the trials to enhance the e-mail system management and efficiency are evaluated. These trials are the traditional e-mail servers (yahoo, hotmail, etc.) [10] and the Marsono model. The traditional e-mail servers just upgrade the application and neglect the infrastructure of the e-mail protocol that will enhance the entire system. The last trial in the e-mail message prioritization is Marsono model [11]. In the Marsono model the e-mails are classified into two categories, spam and non-spam. Non-spam messages are given higher service priority than the spam messages. It uses two queues to filter the e-mails before handling them. To decrease the delay and loss probability for the non spam messages, Markov chain is used. The main disadvantage of the Marsono model is that it can't be considered a long term solution. This is because; it gives a priority to the non spam messages, but didn't try to enhance the environment service like bandwidth (i.e. what will be done if the e-mails face a network bottleneck during the trip from the sender to the receiver). So, the Marsono model extension can be used as a second solution level after upgrading and enhancing the e-mail system environment level.

4. PROPOSED E-MAIL MANAGEMENT SYSTEM

This section demonstrates how the proposed system works, the server database additional entities, the server reports and messages, and the additional message header fields.

4.1 How the proposed E-Mail management system works

In this system, the messages are prioritized in three levels: high, medium, and low. The user specifies the level of his message before sending it. Each client (user) has a maximum number of high and medium priorities that should be sent within a predetermined interval (in this paper, this interval is assumed to be one month). Within the first month after the e-mail creation, the e-mail server accepts the message from a client without restrictions to test the client behavior. If a client sends a number of prioritized messages greater than the predetermined maximum number for this type of priority, the e-mail server restricts the message sending process.

Each priority level has a special confirmation method. The high priority message should be confirmed using the TCP as a transport layer protocol to guarantee the reliability. For the medium priority messages confirmation, a new simple version of TCP is used as a transport protocol. This version replaces the regular TCP confirmation messages (ACK), which are sent when the sender message reaches the receiver, with a cumulative confirmation message. The low priority messages may be confirmed or not depending on the network state (overloaded or not).

The client has the authority to determine the number of high, medium, or low priority messages that should be received from other clients. For example: a client A can determine Z of high priority messages that should be received from clients B, C, and D. All of these data are stored in the client table that should be constructed with the e-mail creation at the server side. Also, the e-mail server can restrict the messages automatically by reviewing the client tables at the sending and the receiving processes. This strategy decreases the bad effect resulting from complete e-mail addresses blocking.

Every time interval (one month) the server sends a cumulative report for each client. This report contains a summary of client activities on the system like the number of high priority messages that are sent and the busy areas in the inbox, the bulk, and the spam. Upon this report, client can adjust the sending rate of each priority type (i.e., the client takes a decision as regards the messages that will be sent at each time interval).

It is possible that the urgent massages may be automatically directed to the spam or the bulk. This makes the message vulnerable for deletion. So, a new management technique for client inbox is demonstrated. A prioritization system helps us to arrange the messages and delete the lower priority message. There are three factors that should be considered in automatic message deletion. These factors are; 1) the status of message (read or not); 2) the size of the new message is greater than the inbox free area; 3) the new message should have a priority type higher than the message(s) selected for deletion.

The e-mail system contains three main boxes: inbox, bulk, and spam. As a client opens an account on the server; the server gives initial size of each e-mail box. The size of inbox is constant for all clients. But the size of spam and bulk may be changed depending on the history of the client operations (sending, receiving, and blocking). If the server system detects that the client has more activities, it can automatically increase the spam, or the bulk size, and vice versa.

The maximum number of messages for each priority type is adaptable value and not fixed for all servers. The owner of e-mail server can determine the number of messages for each priority type. But, there are other factors that must be taken in consideration when these values are calculated. The factors that control these initial values determination process include the number of clients, the available bandwidth of the e-mail server, the average size of sent messages, and the activities history for each client within a time interval.

The client connects to the server using SMTP protocol [12]. The client message, with a specific priority type, plus the basic e-mail header fields and the message data, is created and sent. The message arrives to the e-mail server and all the entries of the client table (see subsection 4.2) are filled. Consequently, the server checks if the flag value equals zero or one. If zero, this means that the interval from e-mail creation time till the sent message time didn't pass a month (the month is selected as an initial interval), and the message is wrapped and sent. Hence, the counter of the sent message priority type is increased by one. If the flag value equals one, this means that the interval from the e-mail creation till the sent message passed a month. Hence, before the message is sent, the system should test the limitations of each priority type and compare it with the real number of sent messages. If the real number of sent messages is less than the upper limit, the message will be sent, else the message is failed, and a failure message, called priority overflow, is sent to the client. The flowcharts that explain each process of the proposed e-mail management system are shown at Figs. 2, 3, 4, and 5.

Regarding the relation between our new system and other e-mail systems, it is determined by the number of messages that should be received from other e-mail system clients due to our e-mail system restrictions. In addition, clients of other systems can determine the priority level in the message title. So, our server should test the message title that comes from other e-mail servers, to restrict or permit the sending process.

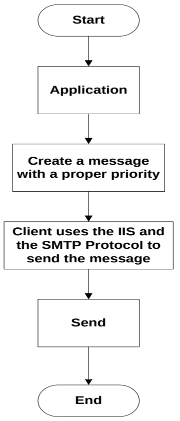

Start
Application
Create a message with a proper priority
Client uses the IIS and the SMTP Protocol to send the message
Send
End

Fig. 2: The sending message technique.

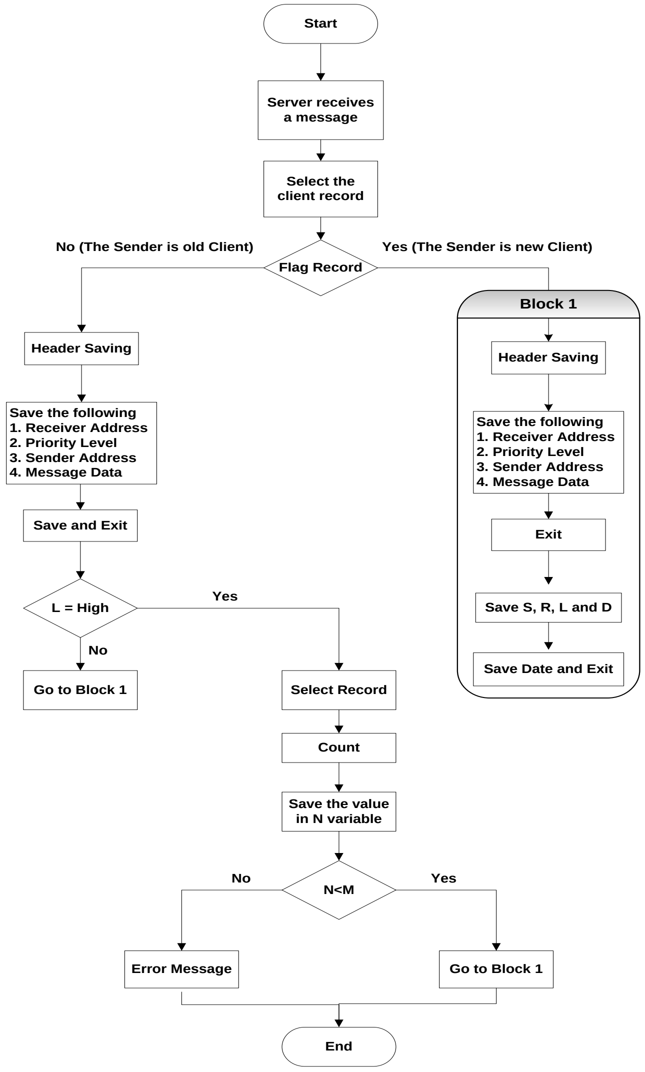


Fig. 3: The message registration (server side).

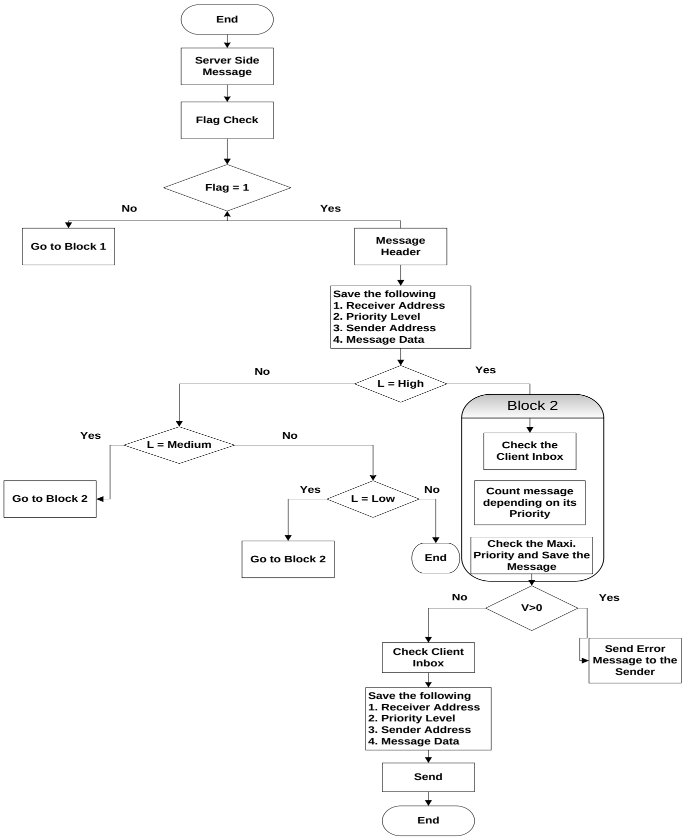


Fig. 4: The message receiving (server side).

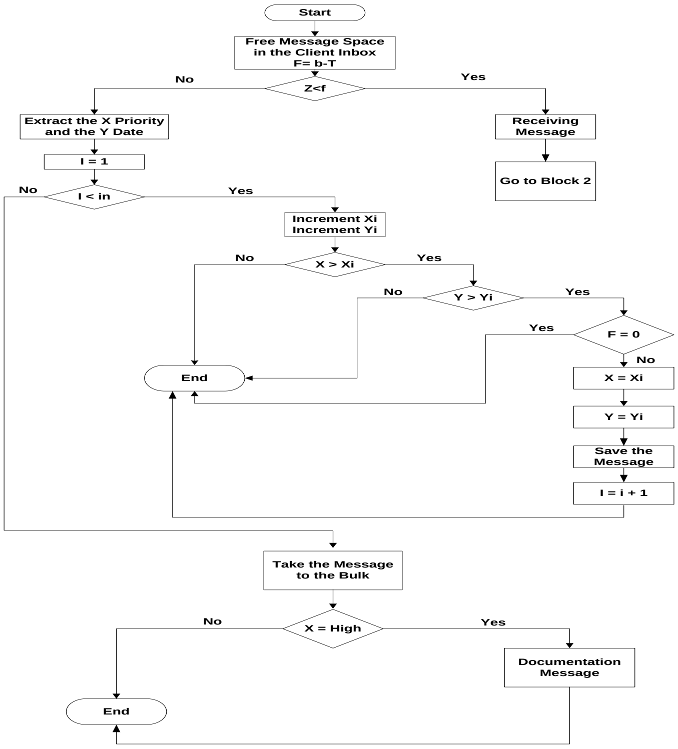


Fig.5: The receiver side

The flowchart symbols description is listed at table 1.

Table 1: The flowcharts symbols description

Symbol	Description
S	Sender address.
R	Receiver address.
L	Priority level.
D	Sent date.
Z	Size of new message.
F	Free size of inbox.
I	Counter.
X[i]	Buffer to test the priority level for each old message.
B	Total inbox size.
T	Busy inbox size.
Y	Received date
O	Maximum number of messages that can be received from a specific client with a special priority level.
V	Real number of messages that are received from a specific client with a special priority level.
M	Number of permitted high priority messages for the client within a time interval (Server side).
N	Number of sent messages with high priority to the client within a time interval.

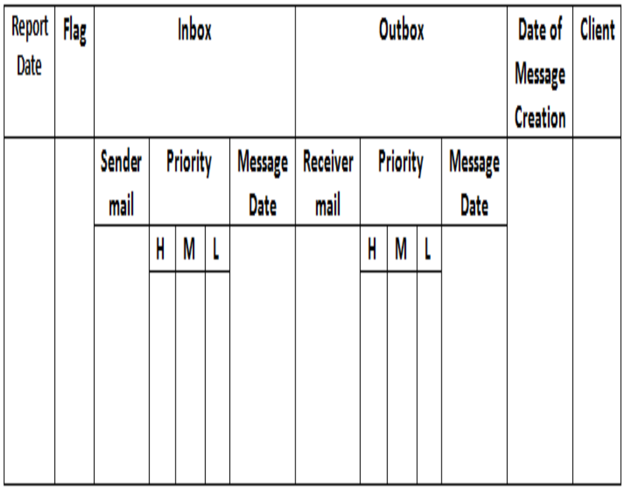
As regards the rogue clients, in our system, we have two sides that can prioritize the e-mail messages, the MTA (e-mail server) and the user. So, the user can control a rogue client by deleting his message from the e-mail box automatically or by blocking his e-mail address. In addition, even if a rogue client sends all messages with high priority level, he can send a limited predetermined number of messages. Hence, the system can be adapted to decrease this number of messages by the time as it knows that he is a rogue client.

4.2 The E-Mail server database additional entities

In addition to the original entries, the e-mail database server contains the following entries for each client, see table 2:

1. Inbox: this entry contains the messages that are received by the client. Each message has a priority type and a sent date.
2. Outbox: this entry contains all messages that are sent by the client. Each message has a priority type and a sent date.
3. Date of a client e-mail creation.
4. Client registration information.
5. Report: containing the number of messages that are already sent and the maximum number of messages that client has a permission to send as regards each priority type.
6. Client report date.
7. Flag: if its value equals zero, the interval from the e-mail creation till the last sent e-mail didn’t pass a month. If its value equals one, the interval time passed a month.
8. Client upper limits: the number of messages for each priority type that can be received from other clients.
9. Server upper limits: The number of messages for each priority type that can be sent by a client.

Table 2: The client database entities on the server



Report Date	Flag	Inbox					Outbox					Date of Message Creation	Client
		Sender mail	Priority			Message Date	Receiver mail	Priority			Message Date		
			H	M	L			H	M	L			

4.3 Server reports and messages

1. Initial Report: The server sends this report to the client after the e-mail creation. This report contains the upper limits of high, medium, and low priority messages that can be sent within the first month.
2. Regular Report: The server sends this report to the client monthly. This report contains the following entries: the maximum number of high, medium, and low priority messages that can be sent within a month, in addition to the real number of messages for each priority type that were sent within this month.
3. Initial Message: The server sends this message to the client at the beginning of e-mail creation to inform the client with the e-mail size (inbox, spam, bulk).
4. Sender Failure Message: The server sends this message to the client when a sending message process is failed due to a priority limitation. For general view of our system see Fig. 6.

4.4 Additional fields in the E-mail message header

For simplicity, only one field will be added to the message header. This field is called priority field. The length of this field is two bits. The value of this field is 00 for the low priority, 01 for the medium priority, and 10 for the high priority.

5. MINOR UPGRADES IN TCP

To enhance the e-mail system efficiency, the bandwidth requirement of e-mail server should be decreased. To decrease this bandwidth, the number of messages that are used in our proposed e-mail system should be decreased. It's noted that the e-mail application layer protocols run over a TCP as a transport layer protocol. Also, it's clear that the TCP is connection oriented and reliable protocol. Hence, the TCP may cause a transmission delay due to the confirmation process for each sent message (ACK). In addition, these ACKs consume bandwidth from the network capacity [13], [14], [15]. The suggested upgrade is to pause the confirmation process of medium priority message for an interval in case of network overload. Within this interval, the server can collect the unconfirmed medium priority messages till the network state is improved and goes to steady state. The techniques at [16], [17] can be used to scale a network bandwidth dynamically. Consequently, the server sends a cumulative report that contains the unconfirmed messages to the sender. The sender uses this report to detect the medium priority messages that have not reached to the receiver. Hence, the sender can retransmit it. If the server waits a long interval, the old messages are deleted from the report. The deletion process will continue until the report becomes full and not overflowed.

This simple upgrade in TCP can be adapted using the options field in the TCP header message [18]. Two bits from the options field can be used as follows: 00 for high priority message, 01 for medium priority message, 10 for low priority message, and 11 for future use. These two bits are taken from the unused option bits [19].

6. PROPOSED E-MAIL SYSTEM ALGORITHMS

A. Message preparing algorithm

1- The client creates an account on the proposed system.

2- The client opens the e-mail system application using its GUI (Graphical User Interface).

3- The client writes his message on the application editor.

4- The client determines the message priority level.

5- The client clicks the send button.

B. Message processing algorithm

1- The client data and the header are capsulated in an e-mail message.

2- The SMTP protocol is used to complete the communication between the client and the server.

3- The server receives the client message and checks the flag field at the message header.

4- If the flag value equals zero

4.1 The server considers the client as new.

4.2 The server acquires the sender address, the receiver address, the message priority, and the sending date from the message header.

4.3 The server stores the acquired data at its database especially in the outbox field related to that client.

4.4 The server increments the number of messages for a priority type that the sent message has been taken.

4.5 The server tests if the number of messages, which is permitted by the receiver to be sent with this type of priority, passes the upper limit or not.

4.5.1 If yes, the server confirms the incrementing process that is mentioned at step 4.4. Then, the message is sent.

4.5.2 If no, the message sending process is restricted, step 4.4 is neglected, and a failure message is sent to the client.

5. If the flag value equals 1.

5.1 The server considers the client as old.

5.2 The server acquires the sender address, the receiver address, the message priority, and the sending date from the message header.

5.3 The server tests if the number of messages that have the same priority as the sent message is less than the permitted number of messages of that priority.

5.3.1 If yes, go to step 4.

5.3.2 If no. go to step 4.5.2.

C. Inbox management algorithm

1- The inbox free area should be calculated by subtracting the total message size from the inbox size.

2- If the coming message size is greater than the subtraction results.

2.1 The server searches for the message that satisfies three factors, 1) is opened, 2) has a priority lower than the coming message; 3) its deletion makes a sufficient area for the new message.

2.2 The server deletes the old message and receives a new message

3. Else, IF the number of messages that should be received from the sender (each client determines this number) is greater than the upper limit of sent messages from that sender

3.1 The server passes the new message directly to the receiver.

3.2 Else, the message is rejected and the server sends a failure report to the sender.

7. THE NEW E-MAIL SYSTEM ENHANCEMENTS

1- New inbox management system.

2- New full message prioritization system.

3- Bandwidth reduction.

4- Provides the client with a new control method.

5- Adaptive inbox, bulk, and spam sizes.

6- New TCP confirmation method.

7- Keep simplicity.

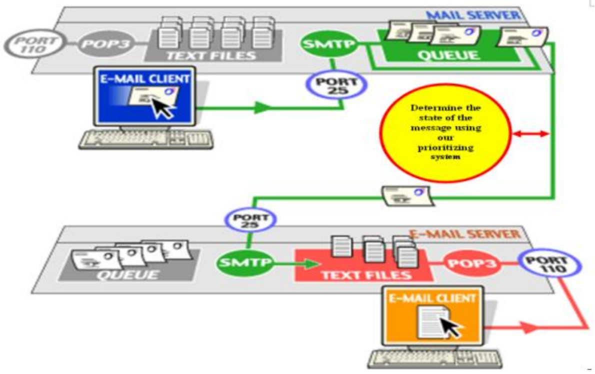
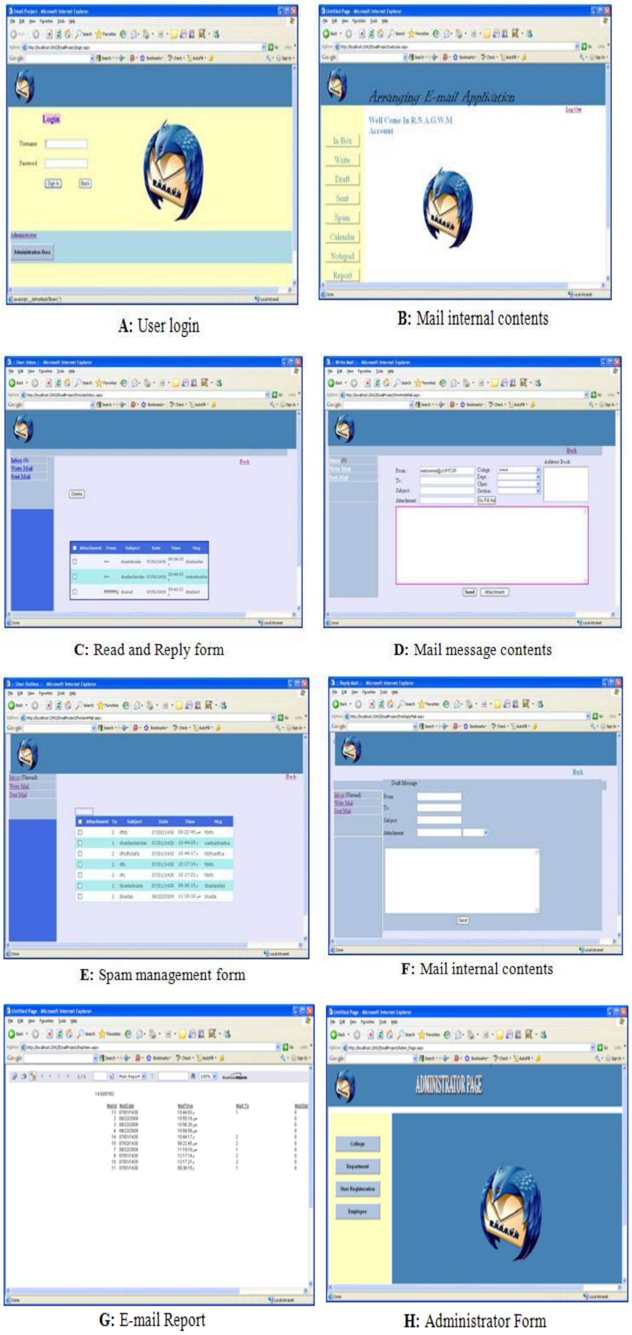


Fig. 6: General view of our proposed e-mail system

8. IMPLEMENTATION

To test our system practically, a simple environment should be installed and programmed. This environment contains the following items; two labs each contain 25 PCs (Personal Computer), 10 laptops, and a powerful server to install our proposed e-mail system. These components are connected via client/server network and have 1 MB internet connection. C# and ASP.net (Active Server Page) [20] are used in our proposed system programming. The code of the e-mail system is found at [21]. Some of output screens are shown in Fig 7.



A: User login

B: Mail internal contents

C: Read and Reply form

D: Mail message contents

E: Spam management form

F: Mail internal contents

G: E-mail Report

H: Administrator Form

Fig. 7: Some output forms from our simple application

9. SIMULATION AND EVALUATION

9.1 Simulation Environment

The ns2 package [22] is used to build our simulation environment. The simulation environment should recover important aspects to obtain accurate results and show how our proposed system enhances the traditional e-mail system. These aspects are the internet simulation and the traffic generation with stressing on SMTP, POP3 and our system modeling.

9.1.1 Internet model and traffic generation

In our implementation heavy-tailed ON/OFF intervals mechanisms are used in addition to heavy-tailed packet sizes to guarantee a self-similar system attitude. In order to make sure that a realistic mixture of different protocols is used during simulations, eight different traffics are defined, which are based on changeable transport protocols: TCP, UDP (User Datagram Protocol), and ICMP (Internet Control Message Protocol). Table 3 shows these profiles with their transport protocols and selection probabilities. The e-mail traffic percentage is notably upgraded.

Before starting our simulation, the host systems, which are defined on the router level topology, are divided into two classes: client system and server system. Client system continuously generates a traffic flow consisting of numerous on/off intervals by randomly choosing one of the existing traffics. This process depends on their choose probability, see table 3. Then, the profiles of traffic randomly decides if this traffic flow is predetermined for a server system through the client's (AS) Autonomous System or away from the AS boundaries. Consequently, the active client sends traffic depending on the traffic profiles to the chosen server. Servers just answer the requests of clients according to their specific role. So, the servers, in contrast to the clients, are considered as passive objects. For more details of internet modeling, simulation, and traffic generation see [23], [24], [25], [26].

Table 3: Mixture of different protocols with different traffics

Traffic Profile Label	Transport Protocol	Selection Probability
Backup traffic	TCP	2.17
Interactive Traffic	TCP	5.31
Web Traffic	TCP	12.61
Mail Traffic	TCP	45.19
Streaming Traffic	UDP	13.11
Name server Traffic	UDP	2.03
Misc Traffic	UDP	11.02
Ping Traffic	ICMP	8.56

9.1.2 SMTP, POP3, and proposed system traffic modeling

The SMTP and POP3 (Post Office Protocol) traffic are both modeled on the connection level. Our experiments contain the complex state of SMTP connection in which emails have a message in the body, three attachment files, sender and receiver’s addresses and a long word subject. The data payload is more than 500 bytes in its TCP packets. All SMTP connections with payloads less than 500 bytes are discarded before modeling. Simulation model of POP3 [27] connections restricts that a POP3 client would always issue several commands, such as LIST and UIDL, after the connection with the server is established and the authorization passed. Hence, the information about the mail-drop on the server is collected. The server responds to each command from the client; however, the length of the response depends on how many e-mail messages of the user account exist on the server. Also, our simulation environment considers a successful POP3 dialog with a remote server, which has an empty mail-drop, has a connection payload contains 90 bytes of data. The POP3 connections are classified into three types: invalid connections (payload < 90 bytes) and cannot accomplish the simplest conversation; unloaded connections (90 bytes < payload < 1000 bytes); and loaded connections (payload > 1000 bytes). Unloaded connections can log in to a mail server, and check the data of the mail-drop, but cannot download any e-mail. Loaded connections can create all events of the unloaded connections in addition to access at least one email from the mail server. Some connections may have payload more than 1000 bytes but cannot access any e-mail. In this situation, loaded connections should be classified as unloaded connections. In practice, TCP header information cannot distinguish them from the real loaded connections (it is feasible to think them as real loaded connections). The following random parameters are used in SMTP and POP3 traffic modeling: connection arrivals of SMTP; SMTP connection transferred bytes; connections arrivals of POP3; POP3 connection transferred bytes. Poisson distribution is used to adapt the processes of SMTP and POP3 connection arrivals [23], [24], [25], [26], [28].

Most of our system traffic model is acquired from the SMTP and the POP3 models. The role file that installed on the server for each client is upgraded to restrict the clients with our system specifications and restrictions. In addition to SMTP and POP3 parameters, the following random parameters are needed to model our system traffic: Client maximum number of sent messages; Client maximum number of received messages; Maximum number of e-mail users; Maximum number of H, M, L messages; and Network bandwidth, see table 4. The xml file that determines the restriction of new prioritized model is found at [21].

Table 4: The simulation parameters required for our new e-mail system

N#	Parameter Name	Value
1	Client maximum number of sent messages.	10000
2	Client maximum number of received messages.	20000
3	Maximum number of e-mail users.	1000
4	Maximum number of H messages.	1000
5	Maximum number of M messages.	2000
6	Maximum number of L messages.	7000
7	Network bandwidth	1 Mb

9.2 Simulation Results

In our simulation the following parameters are scaled and discussed; delay, system efficiency, packet loss, overhead computations, and bandwidth consumption. Fig. 8 shows the delay time of our new e-mail system and the traditional e-mail system to make sure that the number of restrictions didn't affect the sending and receiving of system messages. It's clear from Fig. 8 that the delay difference between the new and the old systems is too few to be considered. Fig. 9 shows the efficiency of the new e-mail management system as regards the old one. The system efficiency is calculated by the number of messages that are sent and received correctly with new prioritized model divided by the total number of messages within a simulation time. Fig 9 shows the notable efficiency improvement for our new e-mail system. This is due to the decrease of real message number that is important for the client. Fig 10 shows the packet loss ratio. It's noted that the number of packet loss under the new e-mail system is less than the old one. This is due to the reduction of bandwidth that is consumed by the new e-mail system packets. It's very important to make sure that the overhead computation didn't notably affect the whole system attitude specially in case of overloading state. So, Fig. 11 is found to show the effect of the overhead computation that resulted from the new prioritized model. The overhead computation is calculated by the number of computations handled at the server and required for each message within the simulation time. It's noted at Fig. 11 that a minor difference is found between our new e-mail system and the old one as regard to the computational overhead parameters. Fig. 12 shows the bandwidth consumption in the new and the old systems. The bandwidth parameter should be scaled to verify the reduction. It's noted that the bandwidth that is taken by the new system is lower than the traditional one. This is due to the limitation of sending the trivial messages for each client and the efficient e-mail box management.

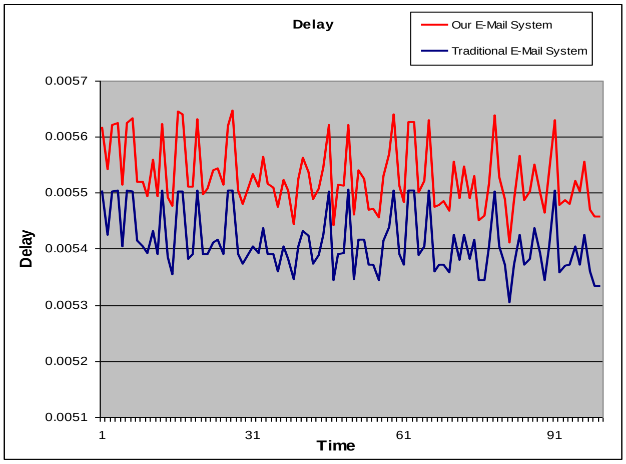


Fig. 8: delay time of the new and the traditional e-mail systems.

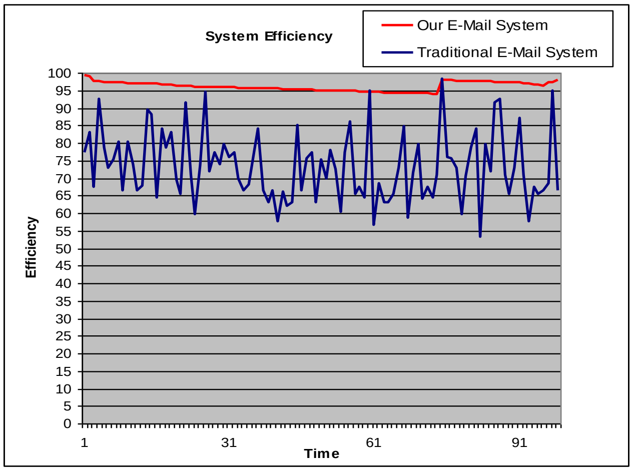


Fig. 9: Efficiency of the new and the traditional e-mail systems

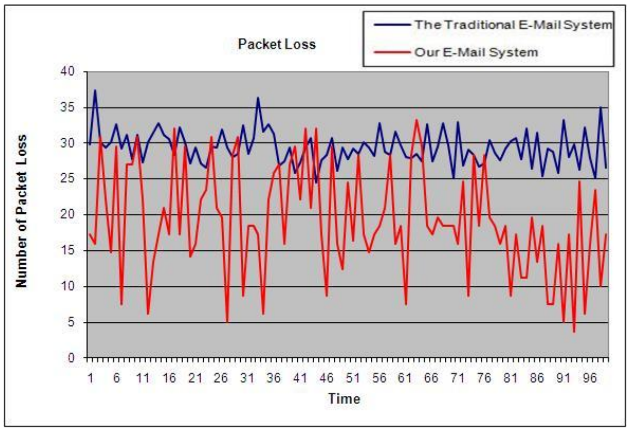


Fig. 10: packet losses of the new and the traditional e-mail systems.

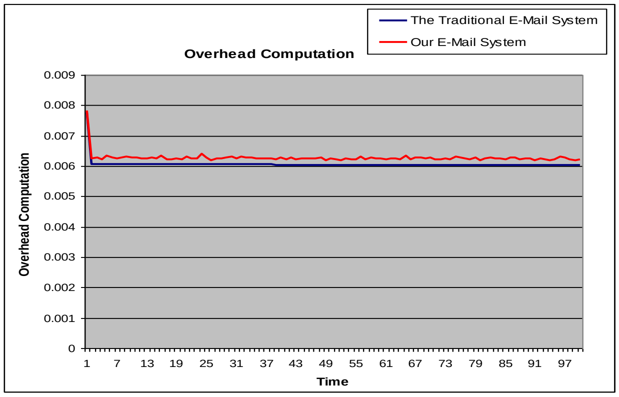


Fig. 11: Overhead computations of the new and the traditional e-mail systems

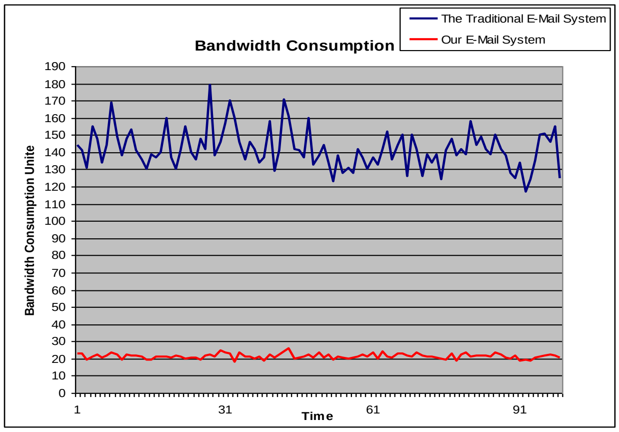


Fig. 12: Bandwidth consumption of the new and the traditional e-mail systems

10. CONCLUSION

In this paper, a new e-mail management system is introduced. In this system the bandwidth assigned for management, sent, and received message is minimized. This is due to the message prioritization technique that handles each e-mail message with a special method. The user can manage his inbox in flexible way. The proposed system recovered the traditional e-mail systems management faults at the server and the user sides. In addition, a simple application output is demonstrated. Minor upgrades are done at the TCP to enhance the system efficiency. A simulation environment is constructed to test our new e-mail system. Finally, extracted results proved that the proposed system outperformed the traditional ones. The following parameters are enhanced; bandwidth, overall system efficiency, packet loss, and delay.

11. FUTURE WORK

There are several open problems in our e-mail system. First, as stated at [11], the Marsono model uses two queues model to provide the non spam message high priority. In our suggested model we have three levels of priorities high, Medium, and Low. So, the high priority message can be considered as a non spam message and other two priorities messages can be considered as spam messages. Our idea is to apply the Marsono model in two stages. The e-mails are filtered into two classes. The first class contains the high priority messages and the second class contains the medium and the low priorities messages. Consequently, the second class is classified into two sub classes. The first subclass is for the medium priority messages and the second subclass is for the low priority messages. The result of this classification is three levels of services. The first level is for the high priority messages and the second level is for the medium messages and the third level is for the low messages, see Fig. 13.

Second, our system should be tested with more number of priorities (4, 5, etc.) till n levels to go towards the system generalization. Third, our system should be tested with next-generation of TCP and compare the results

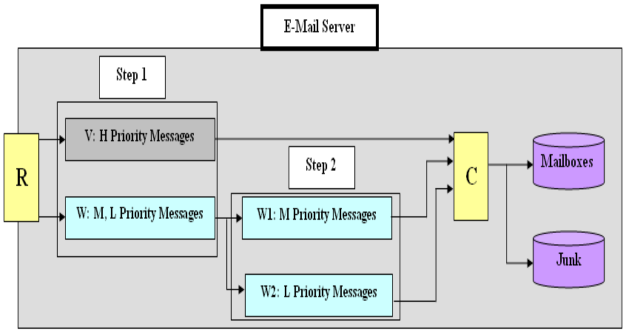


Fig. 13: The adapted two queues prioritized e-mail servicing scheme

12. REFERENCES

[1] “Email system part #1, the main components.” http://codeworks.gnomedia.com/westhost-introduction/email-system-part-1-the-main-components/

[2] Pedram Hayati, Vidyasagar Potdar, “Evaluation of spam detection and prevention frameworks for email and image spam: a state of art”, Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services, Pages: 520-527, Linz, Austria, 2008.

[3] Gordon V. Cormack, Aleksander Kolck, “Spam filter

evaluation with imprecise ground truth" Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, 604-611,Boston, USA, 2009.

[4] Mengjun Xie, Heng Yin, Haining Wang, "Thwarting E-mail Spam Laundering" ACM Transactions on Information and System Security, Volume 12, Issue 2, Article No 13, December 2008.

[5] Lise Getoor, Elena Zheleva, Aleksander kolcz, "Trusting spam reporters: A reporter-based reputation system for email filtering", ACM Transactions on Information Systems (TOIS), Volume 27 , Issue 1 , Article No 3 December 2008.

[6] Miklós Erdélyi, et. al., "Web spam filtering in internet archives" Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web, Pages: 17-20, Madrid, Spain, 2009.

[7] Grieser, et. al., "E-Mail Bounce Management Using Text Mining", HICSS '09. 42nd, Pages 1-10, Hawaii, USA, 2009.

[8] "How Does eMail Work?" http://www.crazysquirrel.com/computing/debian/mail.jspx

[9] Paul Mcfedries, "A Brief E-Mail Primer How Does the Internet E-Mail System Work?" http://www.mcfedries.com/Ramblings/email-workings.asp

[10] "Review and evaluation of Windows Live Mail", Web Developer Notes, http://www.webdevelopersnotes.com/tips/windows_live_mail/windows_live_mail_review.php

[11] Muhammad Marsono, et. al., "Prioritized e-mail serving to reduce non-spam delay and loss a performance analysis", IEEE international Journal of Network Management, Volume 18 , Issue 4, Pages 323-342, August 2008.

[12] Jonathan B. Postel, "Simple Mail Transport Protocol", RFC 821, August 1982. http://www.faqs.org/rfcs/rfc821.html

[13] Xiuchao Wu; Mun Choon Chan; Ananda, A.L.; Ganjihal, C., "Sync-TCP: A New Approach to High Speed Congestion Control", 17th IEEE International Conference on Network Protocols (ICNP) 13-16 Oct. 2009.

[14] Transmission Control Protocol (TCP) Parameters, 2010. http://www.iana.org/assignments/tcp-parameters/tcp-parameters.xml

[15] Jesper D. Brouer, Jørgen S. Hansen, "Experiences with Reducing TCP, Performance Problems on ADSL", DIKU - Technical Report 04/07, 2004. http://web.taranis.org/drraw/papers/DIKU-04.07.pdf

[16] Marshall Crocker , Georgios Lazarou, Julie Baca, Joe Picone, " A Bandwidth Determination Method for IPv6-based Networks", International Journal of Computers and Applications, volume 31 (1), Jan 2009.

[17] "Defining Network Performance". http://media.wiley.com/product_data/excerpt/12/04714330/0471433012.pdf

[18] R. Hay, W. Turkal, Google Inc. "TCP Option to Denote Packet Mood", RFC 5841, 1 April 2010.

[19] http://www.iana.org/assignments/tcp-parameters/tcp-parameters.xml

[20] Danijel Arsenovski "Professional Refactoring in C# & ASP.NET (Wrox Programmer to Programmer)", Wrox, April 20, 2009

[21] "The Proposed Email Project Code" http://www.zshare.net/download/794888471fac1205/

[22] "The Network Simulator - ns-2", 2008. http://www.isi.edu/nsnam/ns/

[23] Songjie Wei, Jelena, Mirkovic, "A realistic simulation of internet-scale events", Proceedings of the 1st international conference on Performance evaluation methodologies and tools, Article No.: 28, Pisa, Italy, 2006.

[24] Oliver, et. al., "On realistic network topologies for simulation", Proceedings of the ACM SIGCOMM workshop on Models, methods and tools for reproducible network research, Pages: 28 – 32, Karlsruhe, Germany, 2003.

[25] Song Luo, Gerald A., Marin, "Realistic internet traffic simulation through mixture modeling and a case study", Proceedings of the 37th conference on Winter simulation, Pages: 2408 – 2416, Orlando, Florida, USA, 2005.

[26] Thomas Gamer, Michael Scharf, "Realistic simulation environments for IP-based networks", International Conference on Simulation Tools and Techniques for Communications, Networks and Systems, Article No.: 83, Marseille, France, 2008.

[27] R. Gellens,et. al., "POP3 Extension Mechanism", RFC 2449, May 1998. http://www.faqs.org/rfcs/rfc2449.html

[28] Fabian Menges, Bud Mishra, Giuseppe Narzisi, "Modeling and simulation of e-mail social networks: a new stochastic agent-based approach", Proceedings of the 40th Conference on Winter Simulation, Pages: 2792-2800, Miami, Florida, USA, 2008.