Performance Evaluation of Mail Systems *

Maria Carla Calzarossa

Dipartimento di Informatica e Sistemistica
Università di Pavia
I-27100 Pavia, Italy
mcc@unipv.it

Abstract. Electronic mail is the core of modern communications. People rely on email to conduct their business and stay in touch with families and friends. The ubiquity and popularity of email make its QoS an important issue. Performance of mail systems is the result of the interactions between their hardware and software components and the user behavior, that is, how users exploit and exercise these components. This paper addresses the performance issues of mail systems by focusing on the characterization of their workloads and on benchmarking. The mail systems considered in this paper rely on Internet standard messaging protocols.

1 Introduction

Electronic mail is one of the earliest and most popular Internet applications that has had major societal and economical impacts [13], [22]. Email is an asynchronous communication medium that connects people across time zones and culture. Email is the core of modern communications and plays a central role for both personal and professional lives. It is used to stay in touch with families and friends and conduct business.

The ubiquity of email availability is one of the strengths of the Internet whose phenomenal growth and popularity are also due to the popularity of email. Email is fast, cheap and easy to distribute and is now as necessary as telephone and other communication media or even more necessary. Email services have become more and more elaborate and powerful and continue to evolve [20]. Messages are no longer limited to textual contents; they can include any type of multimedia contents, that is, images, sound, video and other applications.

Many people rely on email to get work done and information communicated across an enterprise or a research institution. Hence, they expect mail services with a good QoS. Service providers have to meet user requirements by designing and delivering mail systems characterized by high availability and reliability and good performance.

There is a variety of commercial products available for supporting email services. Most of them rely on Internet standard protocols to format, transport

* This work has been supported by the Italian Ministry of Education, Universities and Research (MIUR) under the FIRB and Cofin Programmes.

and retrieve messages, whereas only few use their own protocols, message formats and transports.

The performance of mail systems is the result of a complex set of interactions between their hardware and software components and of the user behavior, that is, how users exercise these components. Hence, it is important to assess and predict the performance of mail systems both at the design stage and during their entire life cycle.

Despite the importance of these topics, few papers deal with the design of mail systems and the analysis of their performance. In [19], the behavior of Lotus Notes email clients is modeled with the objective of balancing the workload on mail servers. The model is based on the distribution of the work generated by users. The characterization of the workload associated with Internet standard mail protocols is addressed in [2]. The models obtained in this study are used describe the input of benchmark experiments. A high–availability high–performance email cluster is presented in [16]. The design of the system is based on the analysis of user needs and behavior. Benchmarking is applied to test the system under heavy load conditions. Benchmarking of mail servers is also addressed in [3] and a prototype of a benchmark tool is presented.

The purpose of this paper is to present a survey of the performance issues typical of mail systems that rely on Internet standard protocols. We first introduce the architecture of mail systems by describing the Internet mail model and the corresponding protocols. We then focus on the performance of mail systems by presenting a methodology for the analysis of their workloads. Benchmarking is then introduced as a technique to assess and compare the performance of different mail systems.

The paper is organized as follows. Section 2 presents the Internet mail model and the protocols used to transport and access messages. Section 3 focuses on workload characterization of mail systems. Section 4 discusses the benchmarking techniques in the framework of mail systems. Finally, Section 5 summarizes the paper and outlines some open issues.

2 Internet Mail Model

Mail systems rely on a client/server architecture consisting of a large number of hardware and software components interacting together. The mail server is the component that provides mail services. The mail clients are the components used to access these services. The network is the communication medium between clients and server.

The architecture of mail systems is organized around the Internet mail model, namely, a collection of standard components whose goal is to provide a framework for carrying electronic messages between users. One main characteristic of these components is their interoperability, that is, their ability to operate on different platforms.

The framework of the Internet mail model consists of agents, mailstores and standards [11], [15]. The standards are the rules that define how to handle mes-

sages, that is, how to format, encode, transport and access the messages. The agents are the software programs responsible to perform a certain task on behalf of a human user or of another program. The mailstore is the filing cabinet of a mail system. User mailboxes are stored in the portion of the mailstore that contains the messages addressed to the specific user.
Mail messages are managed by three different types of agents, namely:

- Mail User Agent (MUA)
- Mail Transport Agent (MTA)
- Mail Delivery Agent (MDA).

The Mail User Agents, known also as mail readers, are the programs that run on the user clients, e.g., PCs, workstations, PDAs, and allow users to read, reply, save and compose messages. Current MUAs are provided with graphical interfaces and a rich set of features that allow users to send and view multimedia messages and attachments and to address security issues. The most popular MUAs are Microsoft Outlook, Netscape Messenger and Qualcomm Eudora.

The Mail Transport Agent is the program that sends and receives messages between mail servers and is responsible to know how to route them. The Mail Delivery Agent is the program that places in the mailstore the messages handed by the MTA.

Figure 1 sketches the hardware and software components and the main functions of the mail architecture. The Mail User Agent is a client/server program

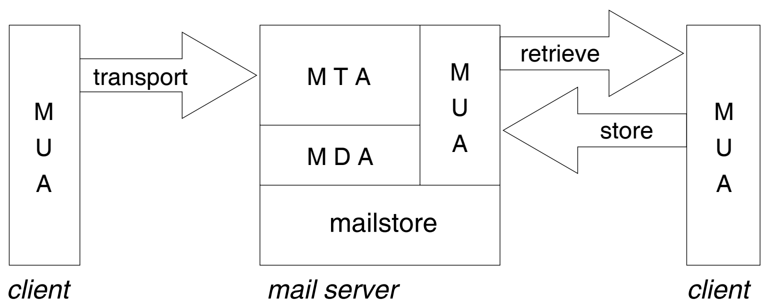


Fig. 1. Components of a mail architecture.

whose client component runs on the user client and whose server component runs on the mail server. Both the MTA and the MDA run on the mail server. An MTA is a client/server program that acts as a client when it sends messages to another MTA and as a server when it receives messages from another MTA. Note that both components of the MTA run on every mail server. The mailstore is located in the mail server. A mailstore can be organized as a traditional flat file or characterized by a more sophisticated organization, e.g., hierarchical, to cope with the increased volume and size of the messages and to allow mailbox sharing among users.

Figure 2 shows the flow and the components of the Internet mail model involved in transporting a message from a sender to a receiver. As can be seen, Mail User Agents are the source and the target of email. At the source an MUA collects messages to be transmitted from a user. At the destination, an MUA accesses and manipulates on behalf of a user the messages in the mailstore. A message originates from the sender's MUA and is transmitted to the MTA of the sender's mail server. This MTA determines the IP address of the receiver's mail server and sends the message to the receiver's MTA that hands it off to the MDA. The MDA deposits the message in the mailstore where it can be retrieved by means of the receiver's MUA. Note that the figure shows a simplified path as a message could travel through multiple mail servers.

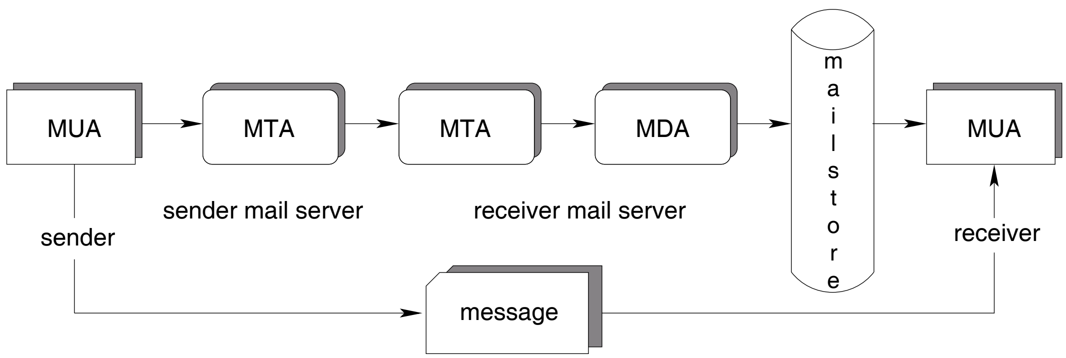


Fig. 2. Flow of a message in the Internet mail model.

Various standards and protocols are associated with the flow shown in Fig. 2. A message, before being transmitted, has to be formatted and encoded according to the Internet Message Format Standard [7] and to its MIME (Multipurpose Internet Mail Extensions) extensions [9], [10].
SMTP is the Internet standard protocol used to transport and deliver messages, whereas POP3 and IMAP4 are the standard mail access protocols. All these protocols are implemented within the corresponding agents. In particular, the MUAs implement the POP3 and IMAP4 protocols, whereas SMTP is implemented within the MTAs and MDAs.

To fully understand the performance and the workloads of mail systems, it is important to understand the characteristics and the behavior of the underlying transport and access protocols. Next sections present a brief survey of these protocols.

2.1 SMTP

The Simple Mail Transfer Protocol (SMTP) [14] is an Internet standard protocol whose objective is to transfer mail reliably and efficiently. The protocol defines the rules for exchanging messages between a client and a mail server and between mail servers. As already pointed out, SMTP is typically implemented by the Mail Transport Agent and Mail Delivery Agent running on the mail servers.

SMTP is a push protocol, that is, the sender's mail server pushes the message to the receiver's mail server. The SMTP protocol consists of two components: the SMTP client that executes on the sender's mail server and the SMTP server that executes on the receiver's mail server. Both components reside on each mail server. When an SMTP client has a message to transmit, it establishes a connection to an SMTP server. An SMTP session is initiated when a client opens a connection to a server and the server responds with an opening message. The connection uses any available transport service, e.g., TCP/IP. Once the connection is established, the client and the server perform some handshaking before the actual transmission of the message.

SMTP is a stateful protocol, that is, both the client and the server know about the state of the connection. The communication between the sender (i.e., client) and the receiver (i.e., server) is an alternating dialogue controlled by the sender. The sender issues a command and the receiver responds with a reply. Each mail transaction involves a sequence of commands and replies. A session is terminated when the client sends a `QUIT` command.

The SMTP client is responsible to guarantee the reliability of the mail transport and delivery. The client either properly delivers the message or reports a failure. For this purpose, the protocol supports queueing capabilities. A message is put in a mail queue whenever a retryable error is encountered by the SMTP client, as in the case of receiver's mail server or network temporarily unreachable.

`Sendmail` [1], [5] is the most popular program that implements the SMTP protocol.

2.2 POP3

The Post Office Protocol – Version 3 (POP3) [18] is an Internet standard mail access protocol whose objective is to allow users to access their mailboxes located in the mail server. Like the SMTP protocol, POP3 is a stateful protocol consisting of two components: a POP3 client running on the user clients as part of their Mail User Agents and a POP3 server running on the mail server. The protocol is very simple and because of its simplicity it provides limited functionalities.

A POP3 session starts when the POP3 client opens a connection to the mail server. The MUA running on the user client issues the commands to the mail server. The mail server responds to each command with a reply. Users can submit commands to retrieve information about their mailbox and to retrieve or delete messages. POP3 supports the off–line model, that is, once the messages have been stored in the client mailbox, the connection with the mail server is closed and the messages are manipulated on the user client. Moreover, messages will be deleted from the server mailbox depending on the option, i.e., `download-and-delete`, `download-and-keep`, specified by the POP3 client.

The POP3 protocol neither keeps any state information across sessions nor provides any mechanism to maintain consistency between client and server mailboxes. Thereby, it is the ideal protocol for users accessing their mailbox from a single client machine.

Today POP3 is the protocol of choice by ISPs as it consumes very little resources on the mail servers and puts the entire load on user clients.

2.3 IMAP4

The Internet Mail Access Protocol – Version 4 (IMAP4) [6] is an Internet standard protocol for mail access designed to overcome the POP3 limitations. IMAP4 is a very flexible and powerful protocol that provides complex messaging capabilities [12], [17]. Users can access and manipulate their mailbox directly on the mail server as it was a local resource.

The IMAP4 protocol supports off–line, on–line and disconnected models. As for POP3, the off–line model allows users to download and manipulate messages on their local machine. With the on–line model, users access their messages directly on the mail server. The disconnected model complements the off–line model in that it allows users to download messages on their client machine, manipulate them and then upload the changes to synchronize the server mailbox such that it is always consistent. Hence, users can access their mailboxes from everywhere using different client machines without the need to replicate their mailboxes.

An IMAP4 session consists of commands sent by the client and responses of the mail server. A session goes through four possible states. Commands and responses are valid only in certain states. The protocol supports a large variety of commands that provide users with many features to manipulate their mailboxes. Users can maintain multiple mailboxes on the server with a hierarchical organization. They can share their mailboxes, search them on the server and selectively choose which messages to download based, for example, on their headers. The IMAP4 protocol puts a significant load on the mail server. It heavily exercises mail server resources for what concerns processing power and disk space. Moreover, demand of bandwidth is high as many IMAP4 sessions can be simultaneously open on the mail server and the duration of these sessions can be quite long.

Because of its flexibility and powerful characteristics, today IMAP4 is the mail access protocol of choice by enterprises and universities willing to offer advanced mail services.

3 Workload Characterization

The workload of mail systems consists of the commands issued by users towards a mail server and of the mail server responses. Moreover, the interaction between clients and server generates packets on the interconnecting network. The

objective of workload characterization is to study the behavior of this composite workload, identify typical usage patterns and predict its future behavior.
Workload characterization can be addressed from different perspectives. The typical approach is experimental, that is, based on the analysis of measurements collected on the system under study [4]. Thereby, to characterize the workload of mail systems measurements of user clients, mail server and network have to be collected and analyzed.

Mail servers typically provide facilities to log all the requests associated with the Internet standard protocols. A large variety of tools is available to collect measurements on the network. The hardest part of the analysis is represented by the clients. Clients are usually "dispersed" and it is difficult to install and run logging facilities on each individual machine. Moreover, users are somehow reluctant to have their activities monitored even for statistical purposes. As a consequence, measurements on the client side are seldom available. The behavior of the users is typically inferred from the analysis of the load of the mail server. Note that mail servers are the main target of the analysis as they are the most critical resources of mail systems, whereas performance of user clients is seldom addressed as an issue.

The level of details and the granularity of the measurements collected vary and depend on the protocol and on the measurement tools. For example, the measurements collected for the commands processed by a mail server include time stamps and other information that is a function of the command type, e.g., size of the message being fetched, number of messages in the mailbox, IP address of the sender and of the receiver of the message. Let us remark that before any processing, log files have to be "sanitized" to remove all sensitive information they might contain.

The analysis of the measurements is based on a typical workload characterization methodology consisting of several steps, namely:

1. identification of the parameters describing the characteristics and the properties of the workload
2. exploratory analysis of the identified parameters
3. application of statistical, probabilistic and numerical analysis techniques to build workload models
4. validation of the workload models.

The parameters have to provide a qualitative and quantitative description of the workload. The choice of the number and type of parameters is driven by the objective of the study. Once the parameters have been identified, the exploratory analysis provides some preliminary insights into the properties and behavior of the workload. Techniques, such as, clustering, factor analysis, numerical fitting, time series, are applied to obtain models able to reproduce the static and dynamic characteristics of the workload. The validity of the identified models is assessed using the representativeness criterion.

Moreover, to represent the hierarchical nature of the workload, it can be decomposed in a hierarchy of layers, where the workload of each layer is transformed into the workload of next lower layer. At the top layer of the hierarchy,

we identify the *users* who are the source of the load of the mail server. Each user opens one or more sessions. Hence, the next lower layer is represented by the *sessions*. Each session consists of multiple *commands*. Each command involves a flow of *packets* on the network, whose number and type depend on the specific command that in turn depends on the mail protocol.
The workload characterization methodology has to focus on each layer of the hierarchy and take into account the properties and the characteristics of each layer together with the transformations between layers.

Note that due to the peculiarities of different mail protocols, workload characterization has to be performed on a per protocol basis, that is, on a per workload type. Next sections address the characterization of the three types of mail system workloads, namely, SMTP, POP3 and IMAP4 workloads.

3.1 SMTP Workload

The SMTP workload refers to the requests, i.e., commands, processed by a mail server to send and receive messages. The parameters used to describe the SMTP workload deal with its intensity and the characteristics of the messages being sent or received.

Workload intensity can be studied in terms of arrival rate, that is, number of requests arrived at the mail server per time unit, and exchange rate, that is, number of bytes sent/received by the mail server per time unit. Figure 3 shows the arrival rate of the SMTP requests over a time period of 24 hours. The measurements refer to a mail server of a small ISP. As can be seen, the

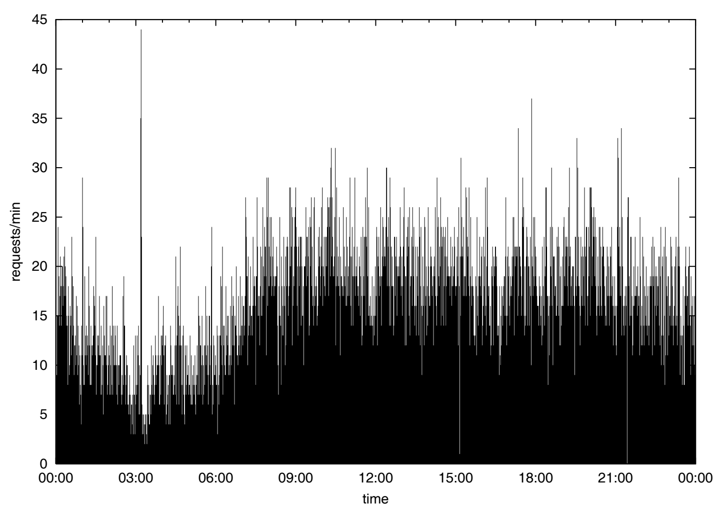


Fig. 3. Arrival rate over a 24-hours period.

arrivals exhibit large fluctuations and burstiness. The load is light during the early hours of the day, and then it slowly increases. The average arrival rate is 16.58 requests/min.

Messages can be characterized by quantitative parameters, such as, size, number of recipients, IP address of the sender and of the receiver. Figure 4 shows an example of a message size distribution. The diagram plots a zoom of the

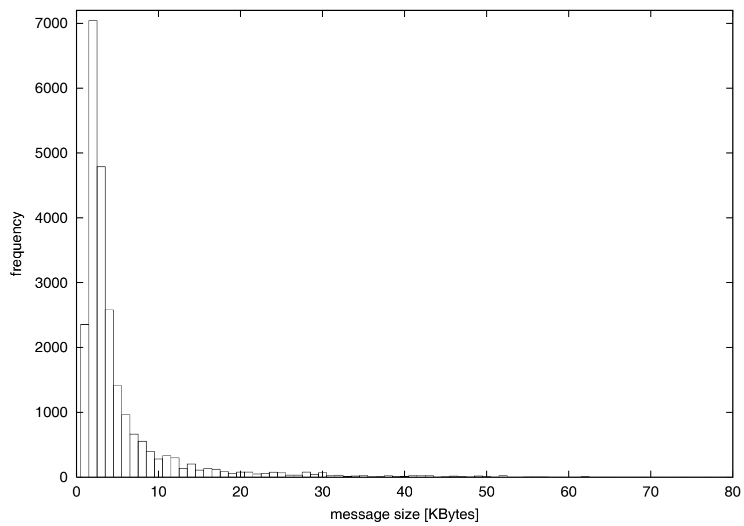


Fig. 4. Message size distribution.

distribution in the range 0-80Kbytes. As can be seen, the distribution is highly positively skewed. The average message size is equal to approximately 7.7Kbytes with a standard deviation more than six times larger than the corresponding mean. Most of the messages are short, even though there are messages as large as 5Mbytes.

Another important parameter that describes the SMTP workload is the interarrival time, that is, the time elapsed between two consecutive messages sent or received by the mail server. Figure 5 shows the interarrival time distribution obtained from measurements collected over a 12 hours period. The figure also shows the corresponding models obtained applying numerical fitting techniques. Two distributions, namely a Weibull distribution and a Pareto distribution, have been identified to represent the empirical data. A Weibull distribution, whose analytic expression is given by:

$$f(x) = \frac{b}{a} \left(\frac{x}{a} \right)^{b-1} e^{-\left(\frac{x}{a}^{b} \right)}$$

with shape parameter a equal to 3.027 and scale parameter b equal to 1.124, best fits the body of the empirical distribution. A Pareto distribution, whose analytic expression is given by:

$$f(x) = \alpha k x^{-(\alpha+1)}$$

with shape parameter α equal to 2.459 and location parameter k equal to 3.958, best fits the tail. The dashed vertical line shown in the figure denotes the threshold, identified by right censoring techniques, between the body and the tail of the empirical distribution. This value is equal to approximately 7 seconds.

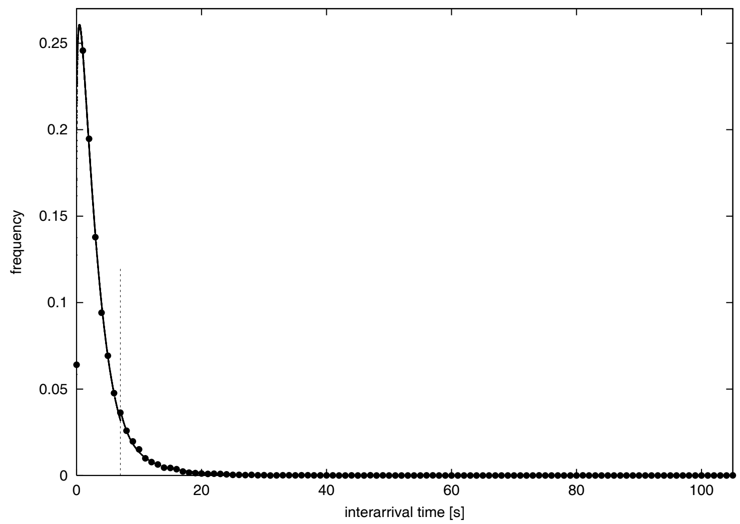


Fig. 5. Empirical interarrival time distribution (dotted curve) and fitting models (solid curves).

3.2 POP3 Workload

The analysis of the POP3 workload follows the hierarchical approach outlined in Sect 3. POP3 users are characterized by the number of sessions they open towards the mail server and by the size and number of messages in their mailbox. Each session is described by its length, that is, number of commands issued within the session, its duration, that is, the time elapsed between opening and closing the connection, and the time between two consecutive sessions. The POP3 protocol supports very few command types. Commands within a session are typically described in terms of their operations on the server mailbox, that is, number and size of the messages retrieved from the server mailbox and number and size of the messages deleted from the server mailbox. Moreover, to derive typical usage patterns, it is important to identify within each session commands issued by human users and commands automatically issued by mail readers to check, for example, for new messages or to keep the connection between client and mail server open.

As for the SMTP workload, models of the POP3 workload are obtained by applying statistical, probabilistic and numerical techniques. In [2], an application of probabilistic graphs to describe the behavior of POP3 users is presented.

3.3 IMAP4 Workload

The description of the IMAP4 workload has to reflect all the advanced features provided by the protocol. IMAP4 users are described in terms of number of sessions they open towards the mail server, number of different mailboxes, size and number of messages per mailbox. As for the POP3 workload, a session is described by its duration, length and time between two consecutive sessions. Figure 6 shows the distribution of the session duration as measured over a 24 hours time period for a population of about 80 users. The diagram plots a zoom

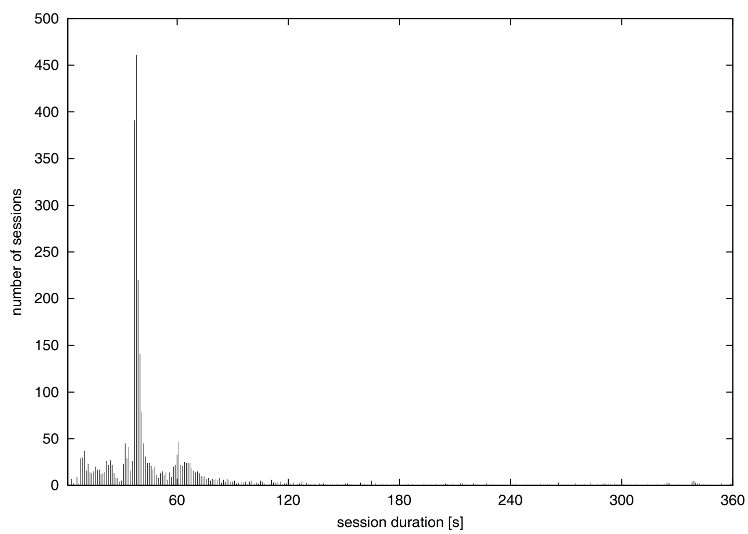


Fig. 6. Distribution of the session duration.

of the distribution for durations in the range 0-360 seconds. Most of the sessions are short, even though there are sessions spanning the whole 24 hours. The duration of approximately 10% of the sessions is between 32 and 37 seconds.

About 92% of the sessions are shorter than 2 minutes and 94% end within 6 minutes.

Figure 7 plots the cumulative distribution of the session length. The distribution refers to lengths in the range 0-400 commands. The diagram shows the

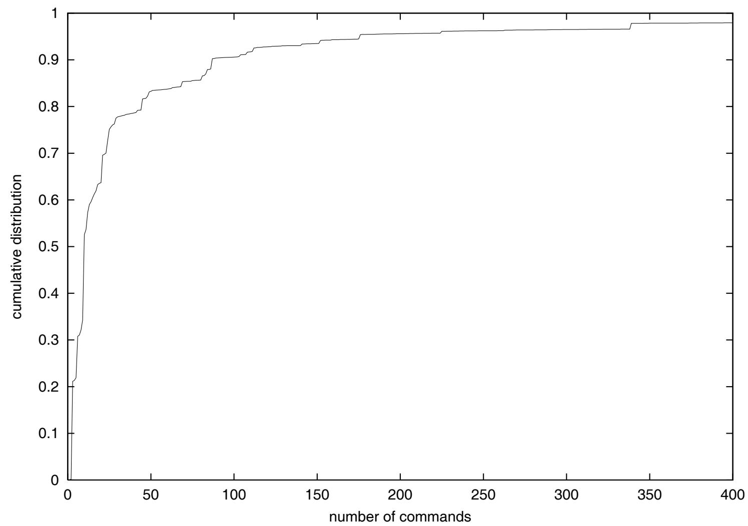


Fig. 7. Cumulative distribution function of the session length.

tendency of the users to issue few commands per session, even though we have observed a session with as many as 39000 commands. The 75-th percentile of the distribution corresponds to sessions with 25 commands at most. The 98-th percentile of the distribution corresponds to sessions with less than 400 commands. We notice that about 20% of the sessions consists of 3 commands only. These sessions correspond to the automatic checks for new messages performed by mail readers and ending up with no new messages in the mailbox.

The IMAP4 protocol supports a large variety of commands. The set of commands implemented varies as a function of the mail reader. Because of the heterogeneity of the commands, their characterizing parameters are specific of each type. For example, a `SELECT` command is described by its time stamp and by the identifier of the mailbox being selected. A `FETCH` command is described by its time stamp, the identifier of the message being fetched, the flags associated with the message and the message size. The quantitative characterization of the commands has to take into account these parameters as well as the description of the traffic generated on the network. Indeed, each command generates multiple packets on the network, whose number, type and size depend on the command type and the data to the transferred.

The functional characterization of the IMAP4 workload takes into account the sequences of commands issued by the users within a session. The models of these sequences are based on probabilistic graphs. The nodes of these graphs correspond to the different command types. The arcs, with their associated probabilities, represent the transitions between commands. Figure 8 shows an example of a graph that describes the behavior of an IMAP4 user. As can be seen, the

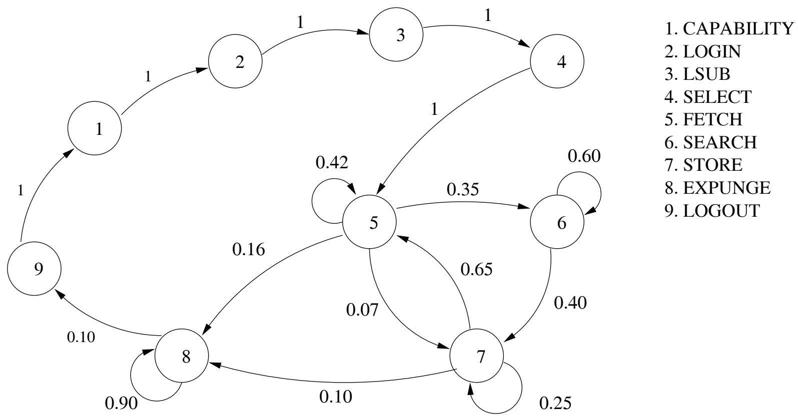


Fig. 8. Probabilistic graph of an IMAP4 session.

user issues nine different command types. The first command of the session is a `CAPABILITY` command. The last command of the session is a `LOGOUT` command. Once the mailbox has been selected, the user fetches the messages from the mailbox, then issues various commands to search, store or expunge messages. Note that the graph contains an extra arc, with probability 1, between the `LOGOUT` and `CAPABILITY` commands. This arc represents the "transition" between sessions, that is, at the end of a session a new session eventually starts.

The behavior of each user is then described by one graph. To obtain more compact and manageable models, clustering techniques are applied to the graphs described by their transition probability matrices. As a result, classes of graphs that identify users characterized by similar behavior are obtained.

4 Benchmarking

Benchmarking is a technique extensively used to evaluate and compare the performance of different systems [8]. A benchmark of a mail system has to exercise its client/server architecture and reproduce the static and dynamic characteristics of the workload. All the components of the system under test (SUT), i.e., user clients, mail server and network, are involved in a benchmark experiment.

Clients play a key role in that they have to load the SUT and emulate the behavior of the users. The accuracy of a benchmark experiment is heavily influenced by the representativeness of the workload being executed.

Figure 9 shows the typical phases of a benchmark experiment. As can be seen, the characterization of the workload is a preliminary step towards the construction of an experiment. The workload is specified in terms of several parameters described by their corresponding models. The number and type of users to be emulated, the user profile, that is, the number and type of commands each user will issue towards the mail server and the user think time, that is, the time between two consecutive commands, are examples of parameters describing the workload of a mail system. As a result of the workload description, the timed sequences of commands that will be executed by the clients are generated. The setup of the SUT deals with the initialization of the working environment, that is, the installation and configuration on the SUT of the software modules required by a benchmark experiment and the distribution of the workload to the individual clients. Moreover, to avoid a “cold start” of the experiment, the mailstore has to be initialized by populating the user mailboxes according to their initial message distribution. After the setup phase, the actual execution of the experiment takes place. The emulated users send their commands to the mail server that processes them and responds with its replies. Note that the execution of a benchmark experiment requires the full availability of the SUT

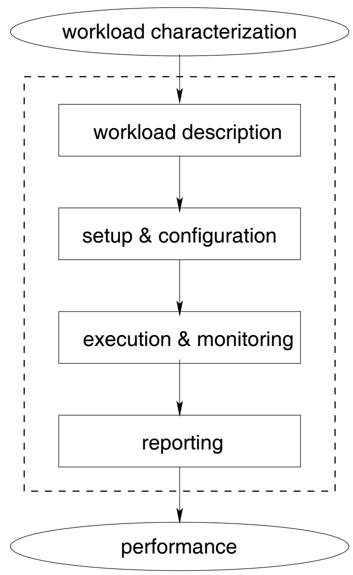


Fig. 9. Phases of a benchmark experiment.

in a dedicated mode. During the execution of the experiment, measurements of the workload and traffic of the network are collected. The reporting phase deals with the analysis of the measurements. As a result of this phase, indices that describe the performance achieved by the SUT are obtained.

Benchmarking tools have to implement all the phases outlined in Fig. 9 and guarantee the repeatability and the reproducibility of the benchmark experiments. SPECmail2001 [21], a standard benchmark that focuses on SMTP and POP3 workloads, describes the user profile in terms of parameters, such as, number of messages sent per user, number of recipients per message, percentage of messages to local destination, message size, number of mail checks per user. Running and reporting rules are defined as part of the benchmark specifications to ensure that the generated results are meaningful, comparable to other generated results and repeatable.
The research prototype presented in [3] focuses on SMTP, POP3 and IMAP4 workloads. Several types of statistical distributions, e.g., uniform, exponential, are provided to describe the workload parameters. Moreover, models based on probabilistic graphs can be used to specify the behavior of the IMAP4 users. To guarantee the repeatability of the experiments, the tool is provided with an archive that acts as a repository of the workload specifications, log files and reports of the benchmark experiments.

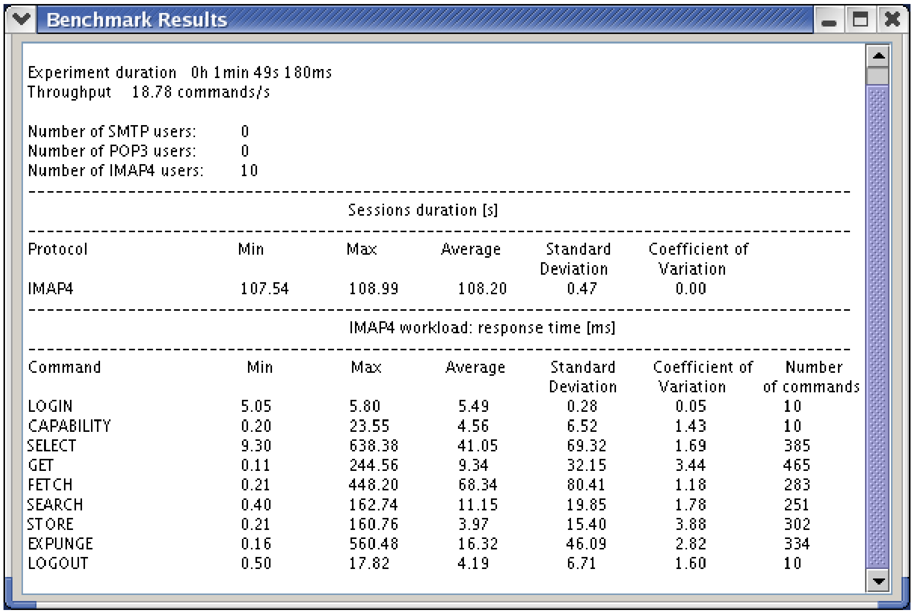
The metrics used to compare the performance of different mail systems typically refer to the throughput of the mail server, i.e., number of commands processed by the mail server per time unit, and response time, that is, time elapsed between the start and the end of a command. A composite metrics, known as "messages-per-minute", is used by SPECmail2001 to indicate the load the server can sustain with a reasonable quality of service.

Figure 10 shows an example of an output of a benchmark experiment. The experiment refers to a run with 10 IMAP4 users, where each user opens one session. The report summarizes the performance of the mail server in terms of duration of the experiment and throughput of the mail server, that is, commands processed by the mail server per minute. Statistics of session duration and response time are also provided. Response times are reported on a per command basis.

5 Conclusions

Performance and reliability of mail services are important requirements for the modern society to work properly. Even excluding spam messages, the number of messages being exchanged world–wide is rapidly increasing. Similarly, the number of new users approaching mail services is increasing. Mail servers have to meet user demands and guarantee a good QoS. In this framework, performance evaluation plays a key role.

This paper focuses on performance evaluation of mail systems that rely on the Internet standard protocols SMTP, POP3 and IMAP4. The analysis of the workload of these systems shows the properties of each type of workload and



Benchmark Results

Experiment duration 0h 1min 49s 180ms
Throughput 18.78 commands/s

Number of SMTP users: 0
Number of POP3 users: 0
Number of IMAP4 users: 10

Sessions duration [s]

Protocol	Min	Max	Average	Standard Deviation	Coefficient of Variation
IMAP4	107.54	108.99	108.20	0.47	0.00

IMAP4 workload: response time [ms]

Command	Min	Max	Average	Standard Deviation	Coefficient of Variation	Number of commands
LOGIN	5.05	5.80	5.49	0.28	0.05	10
CAPABILITY	0.20	23.55	4.56	6.52	1.43	10
SELECT	9.30	638.38	41.05	69.32	1.69	385
GET	0.11	244.56	9.34	32.15	3.44	465
FETCH	0.21	448.20	68.34	80.41	1.18	283
SEARCH	0.40	162.74	11.15	19.85	1.78	251
STORE	0.21	160.76	3.97	15.40	3.88	302
EXPUNGE	0.16	560.48	16.32	46.09	2.82	334
LOGOUT	0.50	17.82	4.19	6.71	1.60	10

Fig. 10. Output of a benchmark experiment.

the behavior of the users. Benchmarking is a powerful technique to assess and compare the performance of different mail systems. An accurate benchmark experiment has to exercise all the components of the systems under a realistic workload.

Open issues in the characterization of workload of mail systems deal with the analysis of the behavior of "spammers" and the identification of atypical usage patterns that might indicate the presence of a security attack against mail servers. Often security threats, e.g., virus propagation, denial of service attacks, come via email and are source of severe performance degradation as they lead to unexpected peaks of very heavy load on the mail servers. It is then important to detect these patterns and design mechanisms for the mail servers to quickly react and adapt to sudden changes in their workload. Moreover, benchmarking tools have to be designed as to accurately reproduce the behavior of the users and the new features of messaging services.

References

1. F.M. Avolio and P.A. Vixie. *Sendmail: Theory and Practice*. Digital Press, 1995.
2. L. Bertolotti and M. Calzarossa. Models of Mail Server Workloads. *Performance Evaluation - An International Journal*, 46(2/3):65–76, 2001.

3. M. Calzarossa. A Tool for Mail Servers Benchmarking. In G. Kotsis, editor, *Performance Evaluation - Stories and Perspectives*, OCG Schriftenreihe, pages 231–240. Austrian Computer Society, 2003.
4. M. Calzarossa, L. Massari, and D. Tessera. Workload Characterization - Issues and Methodologies. In G. Haring, C. Lindemann, and M. Reiser, editors, *Performance Evaluation – Origins and Directions*, volume 1769 of *Lecture Notes in Computer Science*, pages 459–484. Springer, 2000.
5. B. Costales and E. Allman. *Sendmail - 3rd Edition*. O'Reilly, 2002.
6. M. Crispin. Internet Message Access Protocol - Version 4 rev1. RFC 3501, March 2003. http://www.ietf.org/rfc/rfc3501.txt.
7. D.H. Crocker. Standard for ARPA Internet Text Messages. RFC 822, August 1982. http://www.ietf.org/rfc/rfc822.txt.
8. R. Eigenmann, editor. *Performance Evaluation and Benchmarking with Realistic Applications*. MIT Press, 2001.
9. N. Freed and N. Borenstein. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. RFC 2045, November 1996. http://www.ietf.org/rfc/rfc2045.txt.
10. N. Freed and N. Borenstein. Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. RFC 2046, November 1996. http://www.ietf.org/rfc/rfc2046.txt.
11. P. Hoffman. Putting it together: Designs on Internet mail. *netWorker*, 2(1):19–23, 1998.
12. The IMAP Connection. http://www.imap.org.
13. W. Jackson, R. Dawson, and D. Wilson. Understanding Email Interactions Increases Organizational Productivity. *Comm. of the ACM*, 46(8):80–84, 2003.
14. J. Klensin. Simple Mail Transfer Protocol. RFC 2821, April 2001. http://www.ietf.org/rfc/rfc2821.txt.
15. P. Loshin. *Essential Email Standards: RFCs and Protocols Made Practical*. Wiley, 2000.
16. W. Miles. A High–Availability High–Performance E–Mail Cluster. In *Proc. of the ACM SIGUCC Conf.*, pages 84–88, 2002.
17. D. Mullet and K. Mullet. *Managing IMAP*. O'Reilly, 2000.
18. J. Myers and M. Rose. Post Office Protocol - Version 3. RFC 1939, May 1996. http://www.ietf.org/rfc/1939.txt.
19. B. Pope. Characterizing Lotus Notes Email Clients. In *Proc. IEEE Third Int. Workshop on Systems Management*, pages 128–132, 1998.
20. M.T. Rose and D. Strom. *Internet Messaging: From the Desktop to the Enterprise*. Prentice Hall, 1998.
21. SPECmail2001. Standard Performance Evaluation Corporation. http://www.spec.org/mail2001.
22. E.V. Wilson. Email Winners and Losers. *Comm. of the ACM*, 45(10):121–126, 2002.