NIST Special Publication 500-182

Computer
Systems
Technology

U.S. DEPARTMENT OF
COMMERCE
National Institute of
Standards and
Technology

Guidelines for the Evaluation of Message Handling Systems Implementations

Steve Trus
Curtis Royster
Paul Markovitz

NIST Special Publication 500-182

Guidelines for the Evaluation of Message Handling Systems Implementations

Steve Trus
Curtis Royster
Paul Markovitz

National Computer Systems Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899

Sponsored By:
U.S. Army Information Systems Command
U.S. Air Force Communications Command
Internal Revenue Service

August 1990

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
John W. Lyons, Director

Reports on Computer Systems Technology

The National Institute of Standards and Technology (NIST) (formerly the National Bureau of Standards) has a unique responsibility for computer systems technology within the Federal government. NIST's National Computer Systems Laboratory (NCSL) develops standards and guidelines, provides technical assistance, and conducts research for computers and related telecommunications systems to achieve more effective utilization of Federal information technology resources. NCSL's responsibilities include development of technical, management, physical, and administrative standards and guidelines for the cost-effective security and privacy of sensitive unclassified information processed in Federal computers. NCSL assists agencies in developing security plans and in improving computer security awareness training. This Special Publication 500 series reports NCSL research and guidelines to Federal agencies as well as to organizations in industry, government, and academia.

National Institute of Standards and Technology Special Publication 500-182
Natl. Inst. Stand. Technol. Spec. Publ. 500-182, 128 pages (Aug. 1990)
CODEN: NSPUE2

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1990

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402

Contents

List of Figures

List of Tables

1. Introduction

In August 1990, the Government Open Systems Interconnection Profile (GOSIP, FIPS 146) will mandate that Federal agencies procure Message Handling Systems (MHS) products to provide the electronic mail capabilities required by those agencies. The GOSIP Users' Guide (NIST Special Publication 500-163) advances the goals of the GOSIP by providing government technical, managerial, and procurement personnel with the information they need to acquire and use GOSIP compliant products. This document, the Guidelines for the Evaluation of Message Handling Systems Implementations, also advances the goals of the GOSIP by providing guidelines for evaluating MHS implementations. These guidelines can assist a user in the determination of which implementation, among several candidates, will best meet the functional and performance requirements of that user.

The philosophy of the MHS Evaluation Guidelines is explained in an analogy. When people buy new cars, if they make their selection based on a "gut" feeling such as how the car looks or how much fun it is to drive, rather than on concrete facts, they may later find that they did not purchase the "best" car for their needs, and they may be disappointed with their purchase. Likewise if people who are selecting an MHS implementation base their selection on a "gut" feeling such as the implementation's initial appearance, rather than on concrete facts, they may later find that they did not purchase the "best" MHS implementation for their needs, and they may be disappointed with their procurement.

A more logical approach to the problem of purchasing a car which best meets the users' needs is to: (1) Determine the type of car to be purchased, e.g., sedan, sports, van, etc. The type of car can be determined by examining the purposes for which the car will be used, e.g., to drive one person to work, to drive an entire family to various places, to carry packages home from the store, etc. (2) Make a list of cars which are candidates to be purchased. Initially, the list may contain all cars which the user would consider purchasing. After the user has determined any restrictive factors, such as price range, specific manufacturers which the user favors, etc., the list will be narrowed to include only cars which meet the user's restrictive factors. (3) Create one list of functional characteristics of cars, and another list of performance measurements of cars. The user may obtain the information for the lists from product information provided by the manufacturer, magazines which evaluate cars, etc. The functional list would include concrete features such as the number of passengers that can ride in the car, how many cylinders the engine has, the capacity of the gas tank, etc. The performance list would include measurable features such as how fast the car accelerates from 0 to 60 miles per hour, how many miles per gallon of gas the car gets, etc. (4) Create a list containing any functional characteristics and performance measurements which are required by the user. Cars which do not meet these requirements should not be further evaluated. For example, the user may require the car to get at least 25 miles per gallon of gas. Then cars that do not meet this requirement are unacceptable to the user and will no longer be considered. (5) Assign weights to each of the functional and performance features to indicate their importance. For example, the user may consider the feature of how fast the car accelerates from 0 to 60 miles per hour to be of little importance, and therefore assign it a small weight. On the other hand the user may consider the feature of how many miles per gallon of gas the car gets to be very important, and therefore assigned it a large weight. (6) Score each of the cars by summing the availability of each functional feature times its weight and the measurement of each performance feature times its weight, for all of the features. The score for each car reflects how well it meets the requirements of the user. The car with the highest score is likely to be the "best" car for that user. Note that these ratings are not absolute ratings; another user might rate the same set of cars differently based on a different set of needs.

This document provides guidelines for evaluating MHS implementations, using the approach defined for evaluating cars. Details are provided to guide the user through each step of the MHS implementation evaluation process.

1.1 Methodology

This document contains: (1) guidelines for evaluating the functional specifications of MHS implementations, (2) guidelines for measuring the performance of MHS implementations, and (3) guidelines for matching the functional and performance specifications of an MHS implementation to the functional and performance requirements of the user.

The evaluation guidelines are composed of the following components:

(1) An MHS Configuration. The evaluation document provides guidelines for assisting the user in determining the most appropriate MHS configuration.

(2) A list of candidate MHS implementations. The user creates a list of the MHS implementations which are candidates for procurement. The evaluation document provides guidelines for creating this list.

(3) A set of functions. The guidelines provide a set of the functions which may be available in an MHS implementation. The user should become familiar with these functions, noting which ones are important to that user.

(4) A set of performance measurements. The guidelines provide a set of performance measurements which may be derived for an MHS implementation. The user should become familiar with these performance measurements, noting which ones are important to that user.

(5) A set of user requirements. The user determines the user's set of functional and performance requirements. The evaluation document provides guidelines for determining this set.

(6) A rating formula. The guidelines provide formulas to calculate a functional, performance, and overall rating of each of the implementations being considered. The user should become familiar with these formulas.

1.2 Scope

These evaluation guidelines apply to implementations which have been produced according to the 1984 CCITT X.400 series of Recommendations, Version 3 of the NIST Stable Implementors Agreements for Open Systems Interconnection Protocols December 1989, and the COS Stack Specification. The guidelines are intended for users who are installing a private MHS implementation that will communicate with other users' private MHS implementations and/or public MHS implementations.

1.3 Overview

The contents of this document are organized as follows. Section 1 contains an introduction to the document. Section 2 presents an MHS tutorial. Section 3 specifies the procedure for evaluating MHS implementations. This consists of sections providing guidelines for determining the MHS configuration, creating a list of candidate MHS implementations, functional evaluation guidelines, performance evaluation guidelines, guidelines for rating MHS implementations based on their functional and performance rating, an example rating, and miscellaneous guidelines not fitting into the above two categories. Section 4 summarizes the conclusions derived from the project. Appendix A reviews the MHS laboratory used in this project. Appendix B contains a listing of the MHS functions described in these guidelines, presented in a tabular form. Appendix C contains a listing of the performance experiments described in these guidelines, presented in a tabular form. Appendix D defines the abbreviations used in this document, and Appendix E provides a glossary of MHS terms. Following the Appendices is a list of References.

1.4 Acknowledgments

NIST wishes to acknowledge and thank the four vendors who provided implementations to assist this project (Digital, Retix, Xerox, and Hewlett Packard). These implementations facilitated the development of this document. A diagram of these implementations, as configured in our MHS Laboratory, is presented in Appendix A.[1]

[1] Certain commercial products are identified in this report. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment identified is necessarily the best available for the purpose.

2. MHS Tutorial

The Electronic Mail application specified in GOSIP Version 1.0 is based on the CCITT 1984 X.400 series of Recommendations. This section gives a general description of MHS. For more information, see the References portion of this Guide.

2.1 Functional Model

Figure 1 is a drawing of the functional model of the CCITT 1984 X.400 series of Recommendations. There are two major MHS components - the Message Transfer System (MTS) and the cooperating User Agents (UAs). The MTS is composed of a series of Message Transfer Agents (MTAs) that are responsible for relaying the message from the originator's UA to the recipient's UA. The MTA serving the recipient need not be active when the message leaves the originator's MTA; the message can be stored at an intermediate MTA until the recipient's MTA becomes operational. Intermediate MTAs can also perform Application-Layer routing based on address information contained in the message.

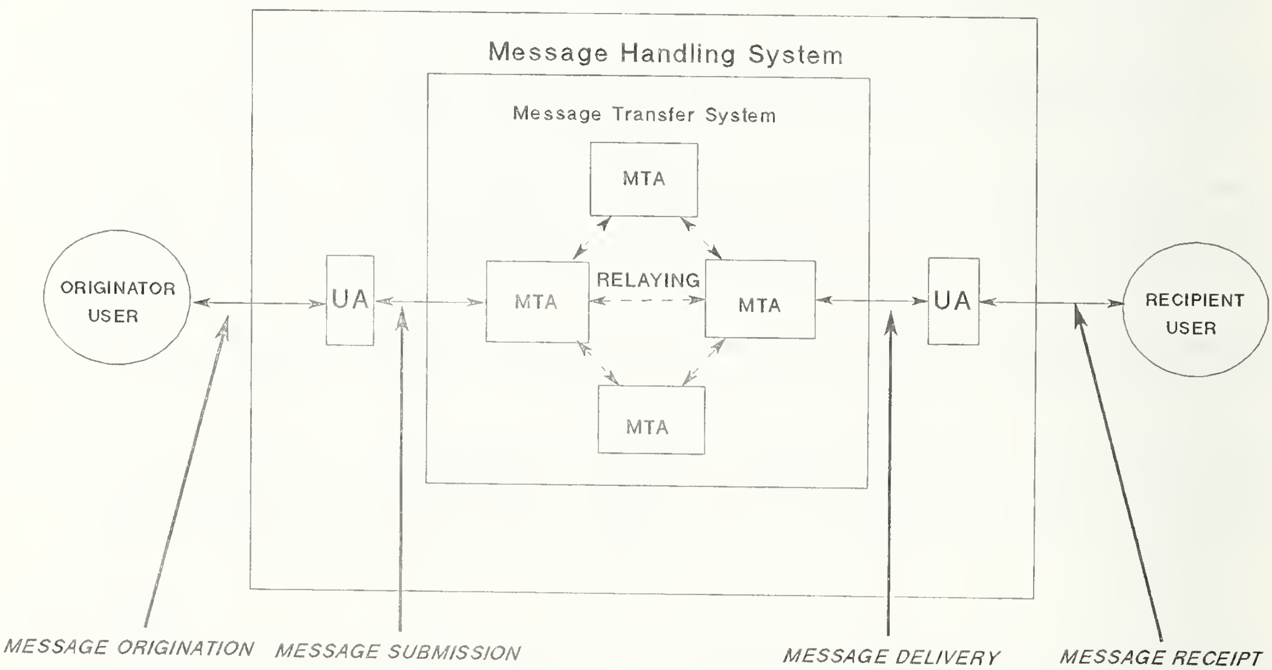


Figure 1. MHS Functional Model.

The MTAs can be managed by different organizations or administrations. An administration is either the central Postal Telephone and Telegraph (PTT) service in a country or, in the United States, a common carrier recognized by the CCITT. The collection of MTAs and UAs owned and operated by an Administration is called an Administration Management Domain (ADMD). The collection of MTAs and UAs owned and operated by a private organization is called a Private Management Domain (PRMD). Figure 2 shows how PRMDs can cooperate with ADMDs and with each other to provide the message transfer service. All ADMDs must comply with the CCITT Recommendations. PRMDs that wish to use a message transfer system provided by an ADMD must comply with the CCITT Recommendations at the point of interconnection.

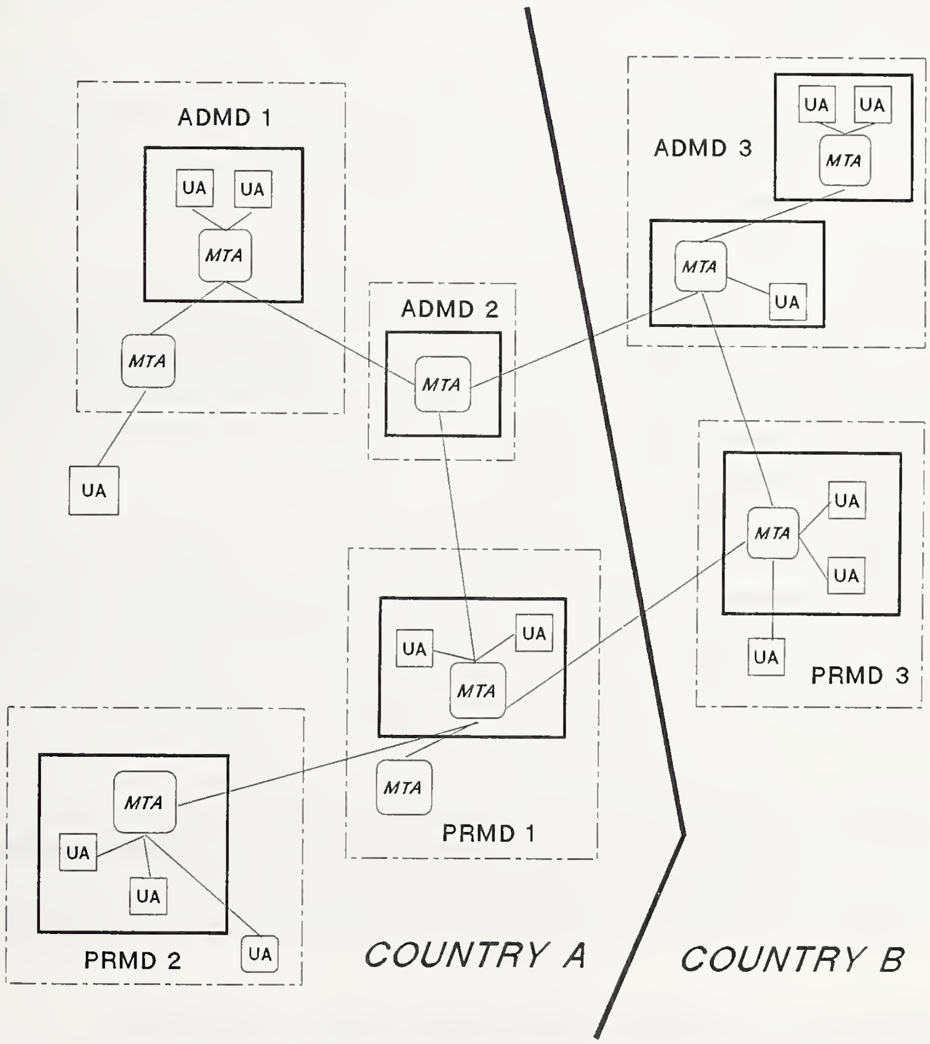


Figure 2. Cooperation of ADMDs and PRMDs.

CCITT has mandated that Transport Class 0 and the Connection Oriented Network Service (CONS) be used in message systems provided by ADMDs. The NIST Workshop Agreements allow PRMDs to use either Transport Class 0 and CONS or Transport Class 4 and either CONS or the Connectionless Network Layer Service (CLNS) at layers 3 and 4. Transport Class 4 and the CLNS are the alternatives most widely implemented in the United States. If a PRMD that does not use Transport Class 0 and CONS wishes to interoperate with an ADMD, a relay MTA containing both Transport and Network Layer implementations must be provided by either the PRMD or the ADMD.

User Agents are the other major component of a Message Handling System. User Agents have many functions that are outside the realm of standardization. The originator's User Agent assists in the creation and editing of a message; the recipient's User Agent stores the message until the recipient chooses to read it and can use certain message fields to determine the display order. However, the message submission and delivery interaction with the MTA must be standardized.

The originator's User Agent must supply to the MTS the message content, the address(es) of the message recipients, and the MTS services that are being requested. The message content is the information that the message originator wants transferred to the message recipient. The address and service request data are placed on the message envelope and used by the MTS to deliver the message.

User Agents can be implemented either in the same system as the MTA or remotely located from the MTA. A remote or stand-alone UA can be under the control of an ADMD, a PRMD vendor, or an organization that provides no message transfer services. Since the UA-MTA message submission and delivery interactions involve a transfer of responsibility for delivering a message, there must be a protocol between the remote UA and MTA to ensure that the transfer of responsibility occurs.

There can be many different types of User Agents. The Message Transfer System can be used to transfer data unrelated to a personal message. It could be a binary bit stream of process control information. As long as the recipient's User Agent can interpret the data sent by the originator's User Agent, meaningful communication can occur. The Message Transfer System does not examine the message content unless the User Agent requests that the content be converted from one format to another before delivery. Although there are many potential User Agents that can use the message transfer services, the most common use of the Message Transfer System is to send a personal message from an originator to one or more recipients. The User Agent that provides this service is called an Interpersonal User Agent and that functionality is standardized in the 1984 Recommendations. Although other types of User Agents have not been standardized they can also use the services of the Message Transfer System as long as they comply with the rules of interaction when submitting or accepting delivery of a message.

2.2 Message Structure

Figure 3 shows the MHS message format. The message structure consists of the message envelope and the message content. As with a postal message, the message envelope contains the information required by the Message Transfer System to deliver the message and the message content contains the information that the originator wants to convey to the message recipient(s). There is a unique message content type (P2) to identify all interpersonal messages. CCITT has thus far standardized only the interpersonal message content type but other content types (e.g., Electronic Data Interchange) will be standardized in the future. Interpersonal messages can be further subdivided into user messages or service messages. If the message is a service message, the message content will contain information about the status of a previously sent message.

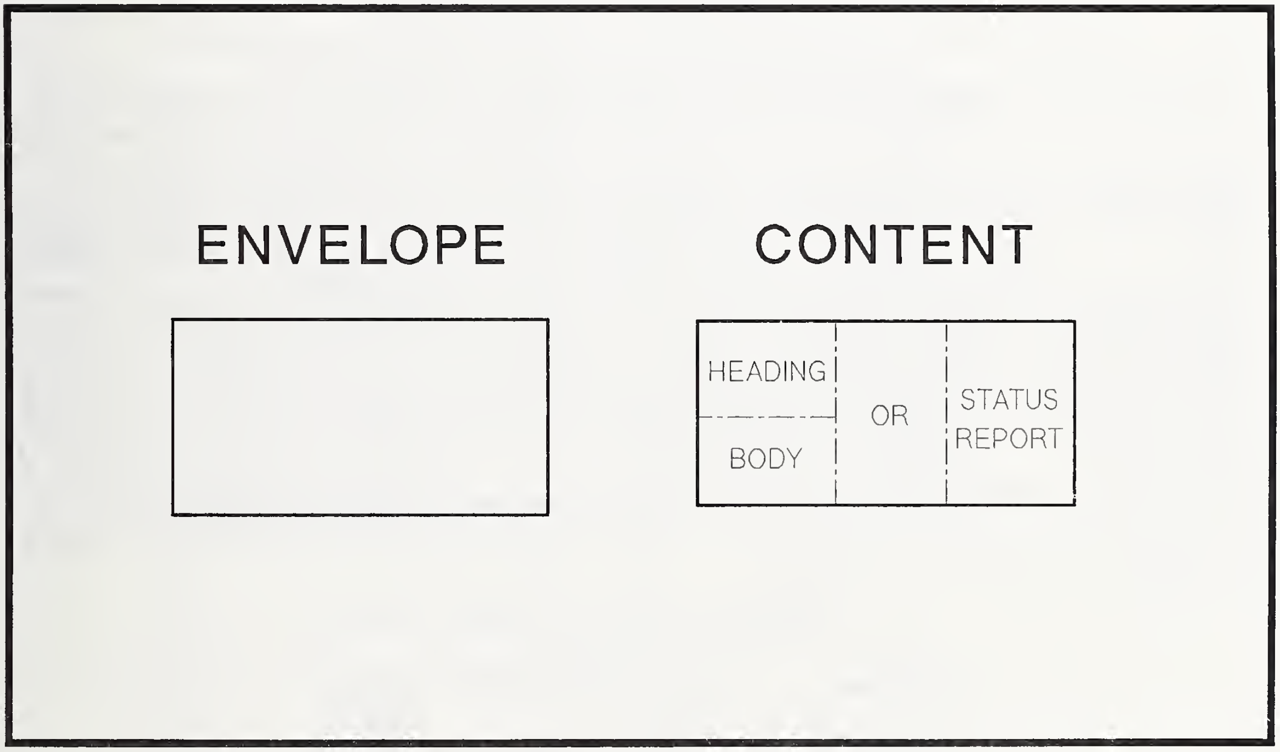


Figure 3. MHS Message Format.

The message content of an interpersonal message is more highly structured than a postal message in order to facilitate processing of the message by the recipient User Agent. For example, if there is a standardized way of cross referencing the incoming message to a previously sent message, both messages could be displayed to the recipient by the User Agent. For that reason, the message content is subdivided into the message header and one or more message body parts. The message header contains a structured representation of information about the message (e.g., the message originator, primary and copy recipients, subject, expiration date, importance, message cross reference, etc.). The message body can be partitioned into several body parts of different types such as IA5 or ASCII text, G3Fax, and Forwarded Interpersonal Message. When a message is forwarded, the header and body of the original message become the body of the forwarded message. See figure 4 for an example of the content of an interpersonal message.

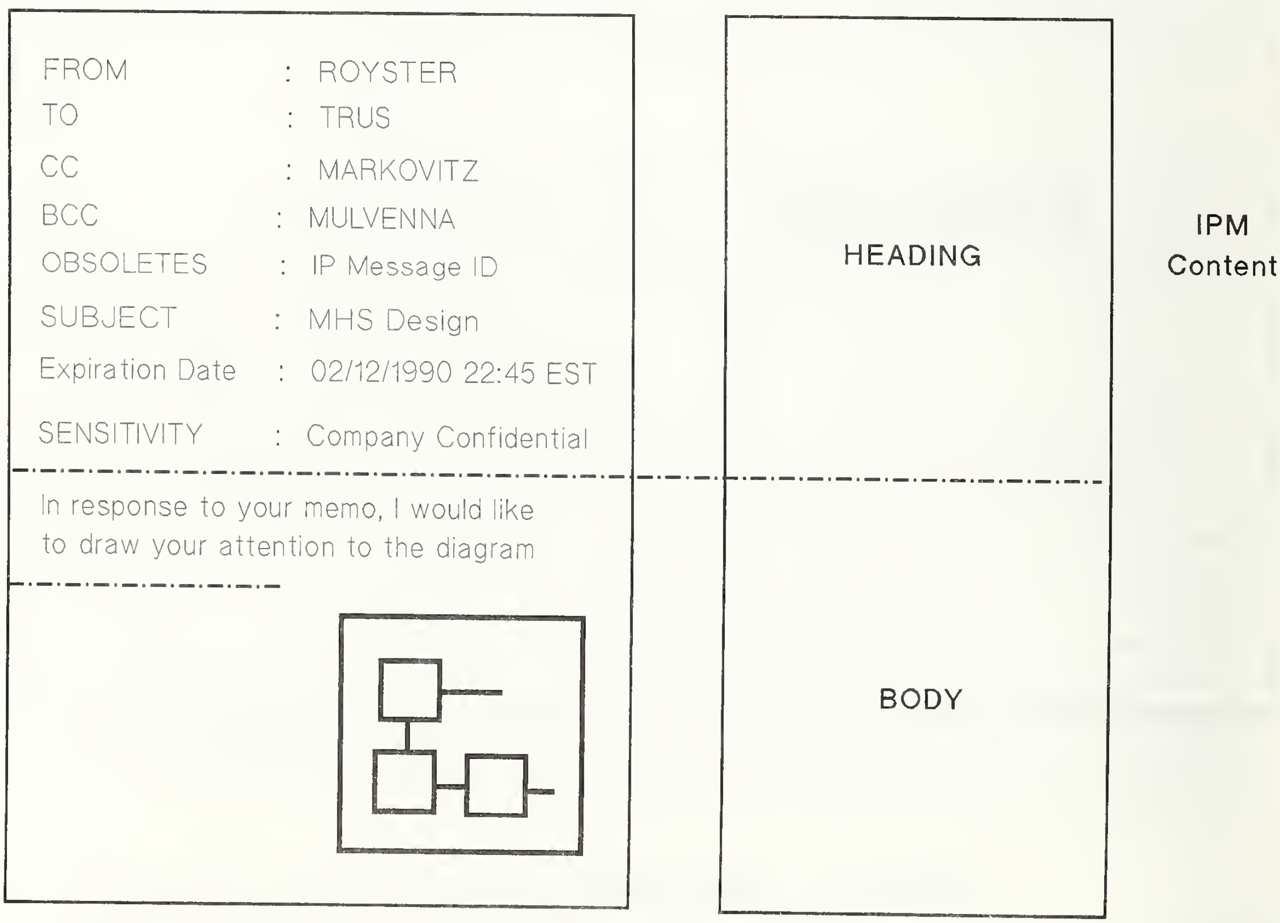


Figure 4. Example Interpersonal Message Content.

2.3 Naming and Addressing

In the context of electronic mail, a name is the term by which originators and recipients of messages identify each other. An address identifies an entity by specifying where it is, rather than what it is. An address has characteristics that help the MTS locate the recipient UA's point of attachment.

A name is formed by specifying a set of attributes and the associated values of those attributes. Table 1 gives an attribute list that can uniquely identify a user of the Message Handling System:

Table 1. MHS Attribute List

Country = United States

Organization Name = ABC Corporation

Personal Name = John Taylor

The address of the message recipient consists of the set of attributes required to deliver the message to the recipient's User Agent.

The CCITT and ISO have developed a standard for a directory service to perform the name-to-address mapping. However, directory service products are not widely available now. In the interim, a method of performing the name-to-address mapping is needed.

The solution is to think of an address as a name that contains attributes that are used to locate the message recipient. Name attributes normally consist of information that the originator knows about the potential recipient of a message. Address attributes describe the architecture of the MTS and may be harder for users to remember but they can be used to route the message to the correct MTA.

Table 2 gives an example of how architectural attributes can be applied to the attributes in table 1 to assist in the message routing.

Table 2. MHS Architectural Attributes

Country = United States

Administration Name = Public Mail System X

Private Domain Name = Private Mail System Y

Organization Name = ABC Corporation

Personal Name = John Taylor

3. Evaluation Guidelines

This section details the evaluation guidelines for MHS implementations, and contains the following sections: Section 3.1 assists users in determining their MHS configuration. Section 3.2 provides suggestions for selecting the MHS implementations which are procurement candidates. Section 3.3 recommends a procedure to functionally evaluate the candidate MHS implementations. Section 3.4 recommends a performance evaluation procedure for the candidate MHS implementations. Section 3.5 recommends a procedure for rating the candidate MHS implementations based on the functional and performance evaluations. Section 3.6 provides an example rating using the previously described guidelines, and section 3.7 describes other factors to consider in evaluating the candidate MHS implementations.

3.1 MHS Configurations

This section assists users in determining their MHS configuration. This configuration is useful in evaluating the functionality of MHS implementations; it can provide input to the user's functional requirements for this evaluation. This configuration is important in evaluating the performance of MHS implementations; the performance of the MHS implementations should be measured on an MHS configuration which matches the user's configuration.

An MHS configuration consists of two parts; a UA-MTA configuration and an MTA-MTA configuration. The UA-MTA configuration details the connections between the UA(s) and MTA(s) within the MHS implementation. The MTA-MTA configuration details the connections between the MTA(s) within the MHS implementation and other MTA(s). If there are multiple MTAs within the MHS implementation, this configuration also details the connections between each of these MTAs.

Most MHS implementations provide one or more of the three UA-MTA configurations presented in this section. If the MHS implementation provides a single UA-MTA configuration, then the vendor has determined the user's UA-MTA configuration. If the MHS implementation provides multiple UA-MTA configurations, then the user may select the UA-MTA configuration, provided by the vendor, which best meets the user's requirements. Users should examine the UA-MTA configurations presented in this section to determine the most appropriate UA-MTA configuration. The following is a description of the UA-MTA configurations.

(1) One or more UA(s) and an MTA running on one system. (See fig. 5.) This configuration is usually found in mini or main frame computers, which have the processing power to run both the UA(s) and MTA on one system.

(2) One or more UA(s) running on one system connected to an MTA running on another system. (See fig. 6.) This configuration is usually found in workstation computers, which do not have the processing power to run both the UA(s) and MTA on one system.

(3) One or more UA(s) running on separate systems, connected to an MTA running on another system. (See fig. 7.) This configuration is usually found in personal computers, which are single user systems.

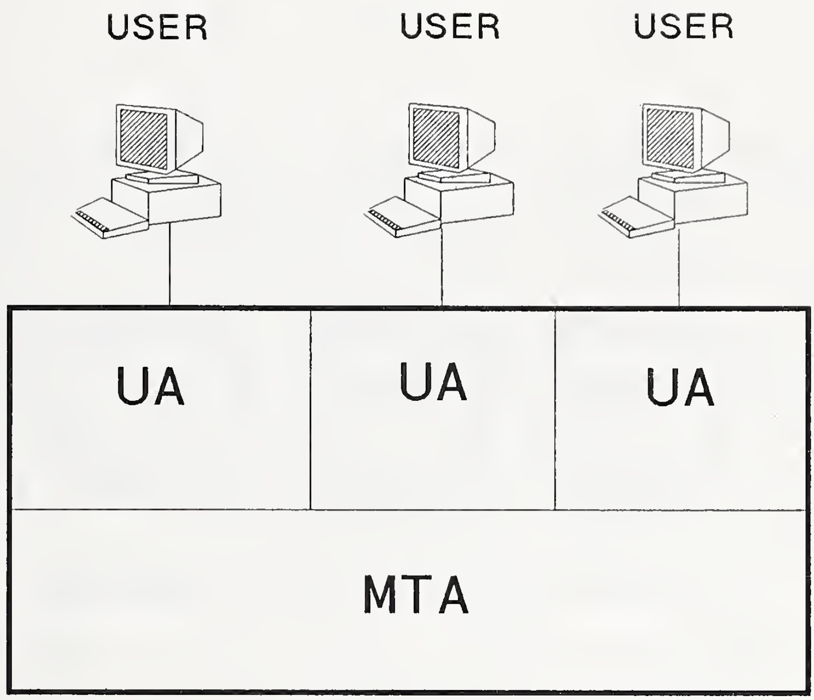


Figure 5. UA-MTA Configuration 1.

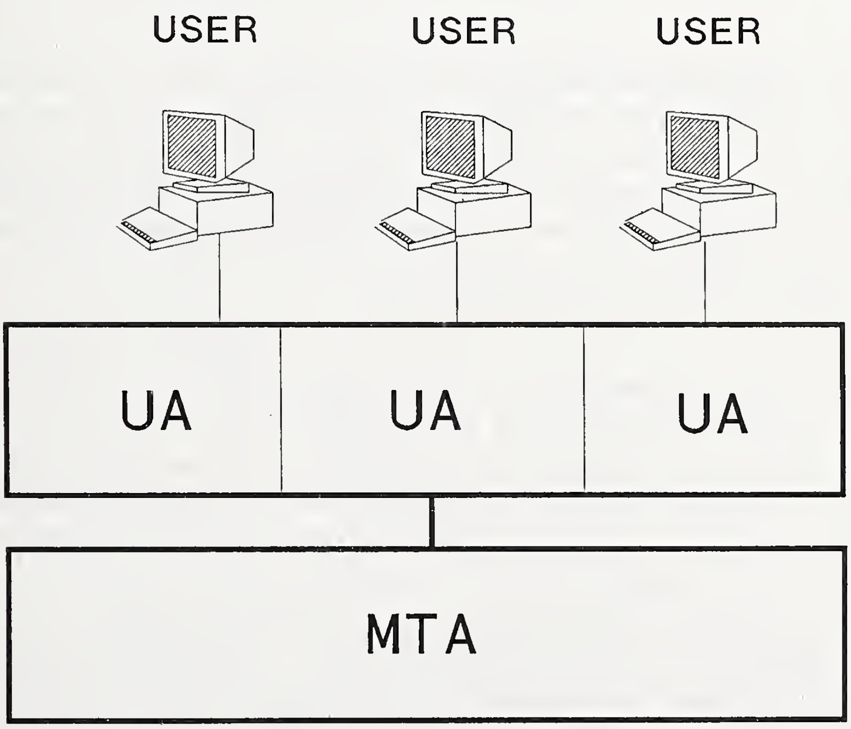


Figure 6. UA-MTA Configuration 2.

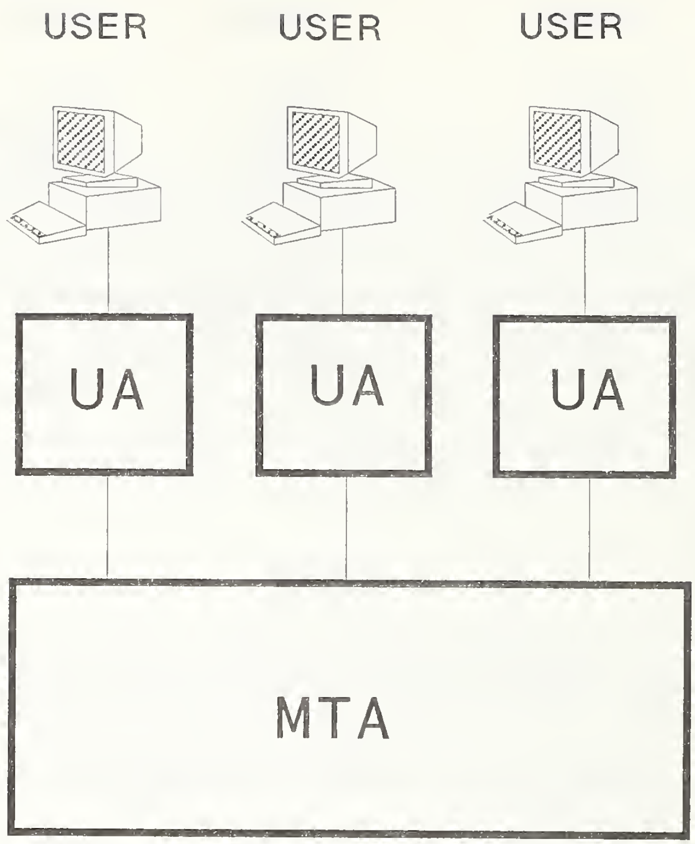


Figure 7. UA–MTA Configuration 3.

Most MHS implementations allow the user to select one of a variety of MTA-MTA configurations as presented in this section. The user's requirements determine which MTA-MTA configuration is to be selected. The user's MTA-MTA configuration will fall into one of three categories: (1) It may match one of these configurations, (2) it may consist of a variation of one of these configurations, or (3) it may consist of a combination of two or more of these configurations. Users should examine the MTA-MTA configurations presented in this section to determine their MTA-MTA configuration.

A user's MTA may be connected to a Wide Area Network, a Local Area Network, or both. If the user's MTA is connected to a Wide Area Network in the form of a Public Data Network, several options for sending mail are possible. The user may send mail to users of another private mail system connected to the same Public Data Network. Additionally, if the Public Data Network provides a public mail system, then the user may send mail to users of the public mail system, or to users who may have mail from the public mail system relayed to them. The following is a description of the MTA-MTA configurations.

(1) Two or more MTAs interconnected by a Wide Area Network. (See fig. 8.) This configuration allows users of a private mail system to send mail to and receive mail from users of another private mail system, or a public mail system. The mail systems are interconnected by a Wide Area Network. An example of this configuration follows. A small company with about 100 employees all located in a single building may have a mail system connected to a Wide Area Network. Users of the mail system may send

mail to and receive mail from users of the same mail system. Additionally, they may send mail to and receive mail from users of another private mail system or a public mail system, which is connected to the same Wide Area Network.

(2) Two or more MTAs interconnected by a Local Area Network. (See fig. 9.) This configuration allows users of a private mail system to send mail to and receive mail from users of another private mail system. The mail systems are interconnected by a Local Area Network. An example of this configuration follows. A medium sized company with about 1000 employees all located in a single building may contain several organizations. Each organization may have its own mail system, which is connected to a Local Area Network. Users of the mail system within one organization may send mail to and receive mail from users of the same mail system. Additionally, they may send mail to and receive mail from users of another private mail system, which is connected to the same Local Area Network.

(3) An MTA connected to a relay MTA by a Local Area Network, and the relay MTA is connected to one or more MTAs by a Wide Area Network. (See fig. 10.) This configuration allows users of a private mail system to send mail as described in MTA-MTA configurations (1) and (2), as well as to send mail to and receive mail from a private mail system which will relay messages to and from users of another private mail system, or a public mail system. The mail system is connected to the relay mail system by a Local Area Network, and the relay mail system is connected to other private mail systems and public mail systems by a Wide Area Network. An example of this configuration follows. A large company with about 10,000 employees has offices located in several cities. Each office contains several organizations, and each organization may have its own mail system, which is connected to a Local Area Network. One of the mail systems within each office is a relay mail system, which is connected to both the Local Area Network, and to a Wide Area Network. Users of the mail system within one organization may send mail to and receive mail from users of the same mail system. Additionally, they may send mail to and receive mail from users of another private mail system, which is connected to the same Local Area Network. Finally, they may send mail to and receive mail from the private mail system which will relay messages to and from users of another private mail system or a public mail system, which is connected to the same Wide Area Network.

(4) Two or more MTAs each connected to different Wide Area Networks, which are interconnected. (See fig. 11.) This configuration allows users of a private mail system to send mail to and receive mail from users of another private mail system. The private mail systems are connected to different Wide Area Networks, and the Wide Area Networks are interconnected. An example of this configuration follows. A large company with about 100,000 employees has a main office in the U.S. and branch offices located in several other countries. The main office and each branch office has a mail system connected to a Wide Area Network. The Wide Area Networks from all of the countries are interconnected. Users of the mail system may send mail to and receive mail from users of the same mail system. Additionally, they may send mail to and receive mail from users of another private mail system, which is connected to one of the Wide Area Networks.

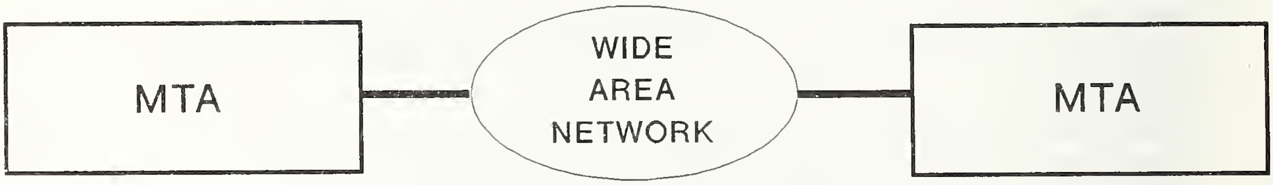


Figure 8. MTA-MTA Configuration 1.

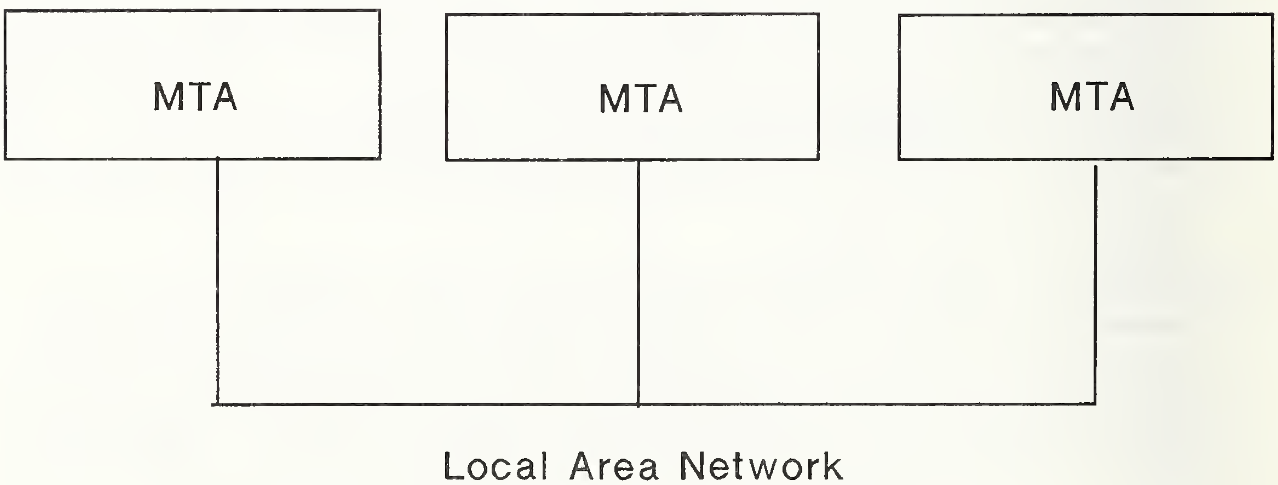


Figure 9. MTA-MTA Configuration 2.

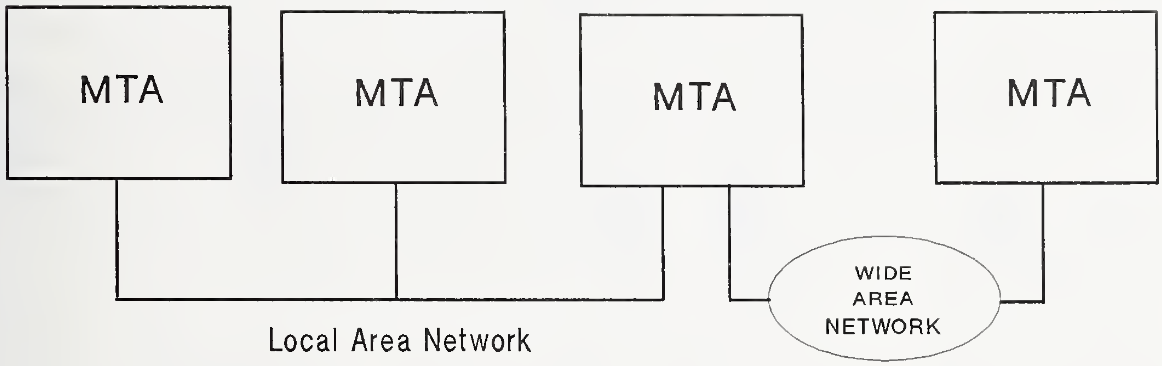


Figure 10. MTA–MTA Configuration 3.

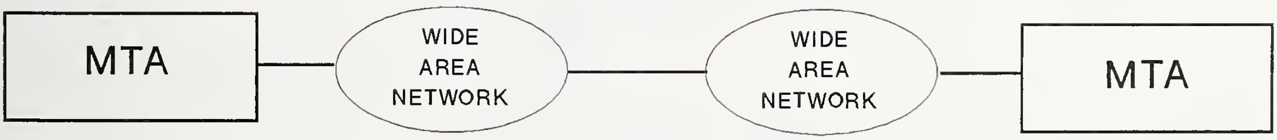


Figure 11. MTA–MTA Configuration 4.

The user's MHS configuration is the combination of their UA-MTA and MTA-MTA configurations. Figure 12 provides an example MHS configuration using the MTA-MTA configuration described in figure 10, the UA-MTA configuration described in figure 5 for one MTA, and the UA-MTA configuration described in figure 7 for the other MTA.

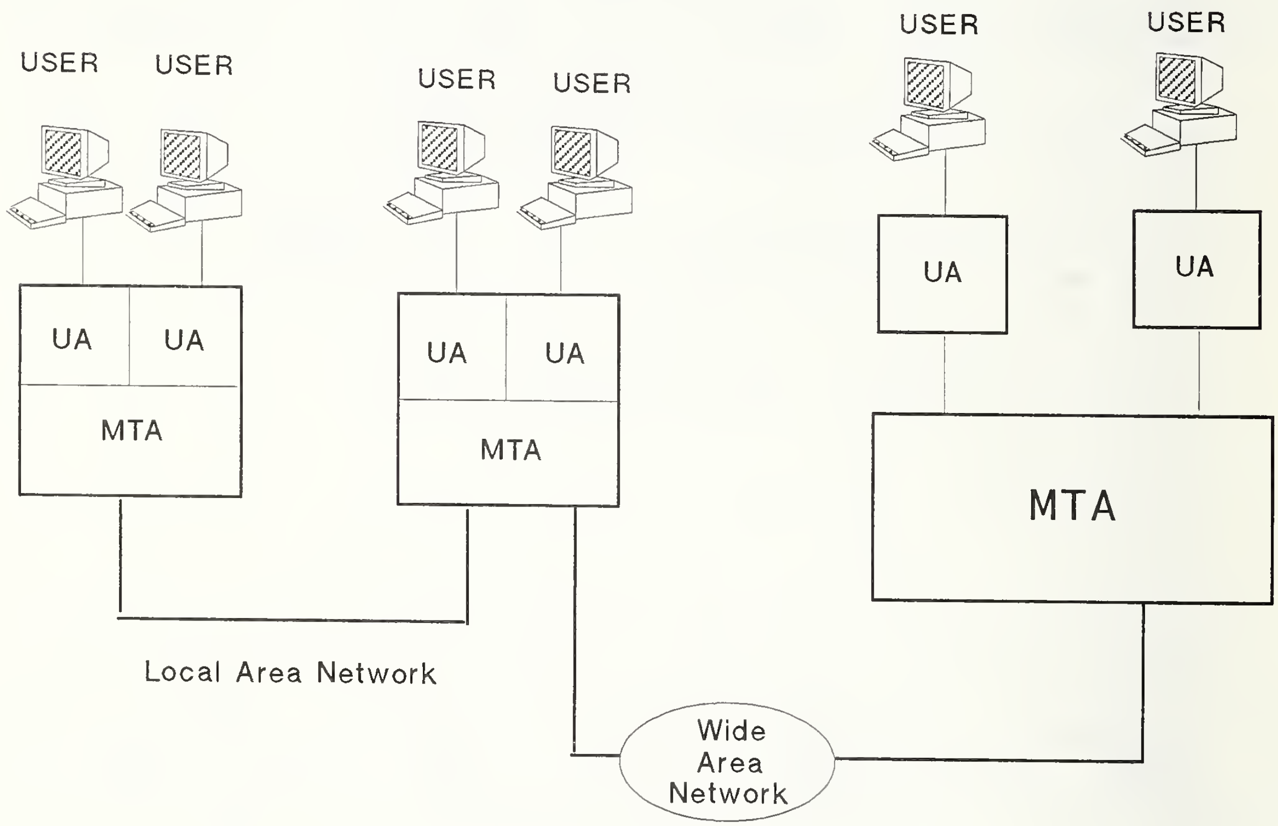


Figure 12. Example MHS Configuration.

3.2 Candidate Implementations

This section recommends a two step procedure for creating a list of MHS implementations, which are procurement candidates. First, the user creates a list of available MHS implementations. The user may find available MHS implementations by contacting prominent computer and communications vendors, checking Commerce Business Daily for product offerings, perusing computer and communications periodicals for product advertisements and product announcements, and by attending computer and communications trade shows. Second, the user determines any user restrictions which apply to the candidate MHS implementations. These restrictions may include specifying the hardware the candidate MHS implementations must run on, specifying the operating system the candidate MHS implementations must run over, and specifying a price range for the candidate MHS implementations. (For example, the user may have a requirement that the candidate MHS implementations run on an IBM PC system. Then, only MHS implementations which run on an IBM PC system would be on the list of candidate MHS implementations.) After the user has determined the restrictions, the list of candidate implementations may be created by placing the available MHS implementations, which comply with the restrictions, on the list. Once the list of candidate MHS implementations is created, product literature, users manuals, technical specifications, and other available information should be requested from each of the vendors for the candidate MHS implementations. This information will provide input to the evaluation procedure.

3.3 Functional Evaluation Guidelines

This section recommends a procedure to evaluate the functionality of candidate MHS implementations. It is divided into three sections. Section 3.3.1 describes functionality potentially available in MHS implementations. Section 3.3.2 recommends a procedure for eliminating candidate MHS implementations which do not meet mandatory functional requirements of the user. Section 3.3.3 recommends a procedure for determining which remaining candidate MHS implementation best meets the functional requirements of that user.

3.3.1 Functional Categories

This section describes functionality potentially available in an MHS implementation. The section is organized by grouping MHS functions into categories of related functionality. This collection of categories provides a representative sampling of the functionality currently available in MHS implementations. The functions comprising the categories were derived from the following sources: (1) the 1984 CCITT X.400 series of Recommendations, (2) Version 3 of the NIST Stable Implementors Agreements for Open Systems Interconnection Protocols, December 1989, (3) the COS Stack Specification, and (4) by working with, and reviewing documentation from, MHS implementations (See app. A) in the NIST Network Applications Laboratory. The user should carefully study each category described to become familiar with the functionality potentially available in the candidate MHS implementations. Although this list is extensive, it is not possible to include every function that is important to every user. The user may insert, in any category, any appropriate functions which are not included in that category, but are important to that user.

Certain MHS functionality must be present in all MHS implementations that conform to the 1984 X.400 Recommendations. These mandatory functions should not be rated; they are included in this document for informative purposes only. Mandatory UA functions are described in section 3.3.1.1, and mandatory MTA functions are described in section 3.3.1.2.

The majority of MHS functions are optional. For this reason, functional evaluation and rating is very important. Optional functionality consists of UA and MTA functions which are categorized as optional in the MHS standard, as well as non-standard functions. Optional UA functions are described in section 3.3.1.3. Optional MTA functions can be divided into three categories. Functions pertaining to the MTS are described in section 3.3.1.4. Types of O/R names that may be supported by an MTA are described in section 3.3.1.5, and message body parts are described in section 3.3.1.6.

The MHS standard leaves many MHS implementation decisions to the vendor. For this reason, the area in which most MHS implementations will vary is the provision of non-standard MHS functionality. A brief introduction to these non-standard functions is presented here, beginning with functionality relating to outbound messages. Sending MHS messages may be viewed as a two stage procedure. During the first stage the user composes the message to be sent. Message composition functions are described in section 3.3.1.7. During the second stage, the user sends the message composed during the first stage. Functions pertaining to sending messages are described in section 3.3.1.8. When sending MHS messages, the user must specify the recipient(s) of the message. Recipients are typically addressed by means of a fully specified O/R name, which may be very cumbersome. Aliases and distribution lists provide alternatives to O/R names. Functions relating to aliases are described in section 3.3.1.9, and functions relating to distribution lists are described in section 3.3.1.10.

Non-standard MHS functionality also pertains to inbound messages. Inbound MHS message functions can be divided into three categories. The first category applies to message reception, and is described in section 3.3.1.11. The second category applies to listing a summary of messages, and is described in section 3.3.1.12. The third category applies to reading messages, and is described in section 3.3.1.13.

Once received, an MHS user may save, delete, or print MHS messages. Functions relating to saving messages are described in section 3.3.1.14. Functions relating to deleting messages are described in section 3.3.1.15, and functions relating to printing messages are described in section 3.3.1.16. In addition, an MHS implementation may provide a user with a default MHS configuration profile. This profile may contain default values for MHS message options, as well as default values for the user's working environment. Functions pertaining to this default configuration profile, as well as examples of profile entries, are described in section 3.3.1.17. Section 3.3.1.18 describes on-line help facility functions. An MHS implementation may provide the user with several interfaces. The UA interface, which can be window-driven or command-driven, determines how easy the implementation is to use. Additionally, a user may be provided with programmable interfaces to the MTA and UA. Functions relating to the UA interface and programmable interfaces are described in section 3.3.1.19. Also, an implementation may allow a user to interface with the operating system on which the implementation resides. Specifically, a user may access operating system commands and the local file system. System interface functionality is described in section 3.3.1.20.

The functions described thus far relate almost exclusively to the UA. They affect any user interested in sending or receiving MHS messages. Additional MHS functionality may be contained within the MTA, which is typically managed by a system administrator. MTA functionality directly affects the system administrator. Its effect upon other users, however, may be less consequential.

Administrative functions relating to the MTA can be divided into four categories: administration, remote connections, access control, and debugging. MTA administration functions relate to the management and maintenance of the MTA, and are described in section 3.3.1.21. Remote MTA connection functions describe connections an MTA establishes with remote MTAs, and connection options available to the system administrator. These functions are detailed in section 3.3.1.22. Access control functions enable the system administrator to limit MTA access, and are described in section 3.3.1.23. Debugging functions aid a system administrator in resolving problems, and are described in section 3.3.1.24.

Functions pertaining to the UA and MTA describe OSI application layer functionality. Additional functionality may be present in other OSI layers. These functions are described in section 3.3.1.25.

The remaining categories in this document include certification, hardware requirements, software requirements, documentation and pragmatic constraints. Certification relates to conformance and interoperability testing, and is described in section 3.3.1.26. An MHS implementation may require certain hardware and software for operation. Hardware requirements are described in section 3.3.1.27, and software requirements are described in section 3.3.1.28. An MHS vendor may provide a user with documentation detailing the use and administration of the implementation. Documentation is described in section 3.3.1.29. Finally, an MHS vendor may place pragmatic constraints on the UA or MTA. These pragmatic constraints are described in section 3.3.1.30.

3.3.1.1 Mandatory UA Functions

Certain functionality must be present in an MHS implementation to conform to the MHS standard and/or the NIST Workshop Agreements. This section describes all functions that must be available in a UA. None of the functions in this section should be rated by the user. The first two functions, IP-Message Identification and message originator, must appear in all interpersonal messages. The other functions will only appear in an interpersonal message if requested by the user.

The IP-Message Identification function enables a UA to provide a unique identifier for each message sent or received by that UA. This identifier is used to refer to a previously sent or received UA message (for example, in receipt notifications).

The Originator function enables a UA to convey the identity, or to accept the identity of the originator of the message. This information is provided to the message recipient.

The Primary and Copy Recipients function enables an originating UA to allow the originator to provide the names of one or more users, and enables the recipient UA to accept the names of one or more users, who are the intended primary recipients of the message, and the names of the zero or more users who are the intended copy recipients of the message. It is intended to enable a recipient to determine the category in which each of the specified recipients (including the recipient himself) was placed. The exact distinction between these two categories of recipients is unspecified. However, the primary recipients, for example, might be expected to act upon the message, while the copy recipients might be sent the message for information only.

The Replying Message function enables the originating UA to indicate to the recipient(s) that this message is being sent in reply to another message. It enables the recipient UA to accept a reply indication. If, by means of the Reply Request Indication service element, the originator of the original message specified the intended recipients of the reply, the originator of the reply should address it to those users. Otherwise the reply should be sent only to the originator. The originator of the reply may also specify the O/R names of additional users who are to receive copies of the reply for information. The recipients of the reply receive it as a regular message, together with an indication of which message it is a reply to.

The Subject function enables an originating UA to indicate to the recipient(s) the subject of the message being sent, and allows a recipient UA to accept a subject indication. The subject information is made available to the recipient.

The Typed Body function enables an originating UA to convey, and a recipient UA to receive, the type of the message body being sent along with the message. Examples of a typed body include: unstructured IA5 text, teletex, and G3Fax.

The UA functions, which the originating UA is not required to generate but which must be processed appropriately upon receipt, are listed in section 3.3.1.2.

3.3.1.2 Mandatory MTA Functions

Certain functionality must be present in an MHS implementation to conform to the MHS standard and/or the NIST Workshop Agreements. This section describes mandatory functions which pertain to an MTA. The functions in this section are not to be rated.

The Alternate Recipient Allowed function enables an MTA to accept a request from a UA that the message being submitted may be delivered to an alternate recipient.

The Content Type function enables an MTA to accept a content type from the originating UA for each submitted message. If an interpersonal message is submitted, the content type is P2.

The Conversion Prohibition function enables an MTA to accept instructions from the originating UA which indicate that encoded information type conversion(s) should not be performed for a particular submitted message.

The Converted function enables an MTA to indicate to a recipient UA that the MTS performed an encoded information type conversion on a delivered message. The recipient UA is informed of the resulting types.

The Delivery Notification function enables an MTA to accept a request from the originating UA that an explicit notification be returned to the originating UA when a submitted message has been successfully delivered to a recipient UA. The notification includes the date and time of delivery. In the case of a multi-destination message, a delivery notification may refer to any or all of the recipient UAs to which the message was delivered.

The Delivery Time Stamp function enables an MTA to indicate to a recipient UA the date and time at which the MTS delivered a message.

The Disclosure of Other Recipients function enables an MTA to accept instructions from the UA, when submitting a multi-recipient message, to disclose the O/R names of all other recipient(s) to each recipient UA upon delivery of the message. The O/R names disclosed are as supplied by the originating UA. Receipt of this function by an MTA is mandatory; generation of this function by an MTA is optional.

The Grade of Delivery function enables an MTA to accept a request from the UA that transfer through the MTS be urgent or non-urgent, rather than normal. The time periods defined for non-urgent and urgent transfer are longer and shorter, respectively, than that defined for normal transfer. In the absence of standardized quality of service parameters, the following delivery time targets are defined in the NIST Stable Agreements:

Delivery Class	95% Delivered Before
Urgent	3/4 hour
Normal	4 hours
Non-Urgent	24 hours

The Message Identification function enables an MTA to provide a UA with a unique identifier for each message submitted to or delivered by the MTS. This identifier is used to refer to a previously submitted message in connection with the Delivery and Non-delivery Notification services.

The Multi-destination Delivery function enables an MTA to accept instructions from a UA that specify that a message being submitted is to be delivered to more than one recipient UA.

The Non-delivery Notification function enables an MTA to notify the originating UA if a submitted message was not delivered to the specified recipient UA(s). The reason the message was not delivered is included in the notification. The following forced non-delivery times are defined in the NIST Stable Agreements:

Delivery Class	95% Delivered Before
Urgent	4 hour
Normal	24 hours
Non-Urgent	36 hours

The Original Encoded Information Types function enables an MTA to accept from the originating UA the encoded information types of a message being submitted. When a message is delivered, it also indicates to the recipient UA the encoded information types of the message specified by the originating UA. Examples of original encoded information types include: unstructured IA5 text, teletex, and G3Fax.

The Submission Time Stamp function enables an MTA to indicate to an originating UA and the recipient UA the date and time at which a message was submitted to the MTS.

The O/R name is encoded as a set of attributes describing the originator and recipient of a message. These attributes are the country name, administration name, private domain name, organization name, organizational unit and personal name. A message is routed to the correct private management domain using the first three of these attributes.

MTAs are classified by their ability to discriminate between O/R names when making routing decisions within a PRMD. Three classes of MTAs are described in the MHS standard, however, GOSIP mandates the use of either Class 2 or Class 3 MTAs. To explain the differences between MTA classes, two examples are provided.

Class 1 MTAs base their intra-domain routing decisions on the Organization Name part of the O/R name. This means that if an MTA is a Class 1 MTA, then that MTA cannot make routine decisions based on organizational unit or personal name. The message routing capability of Class 1 MTAs is illustrated in figure 13. Since MTA A is a Class 1 MTA serving the Organization XYZ, both UA A and UA B must be assigned to MTA A.

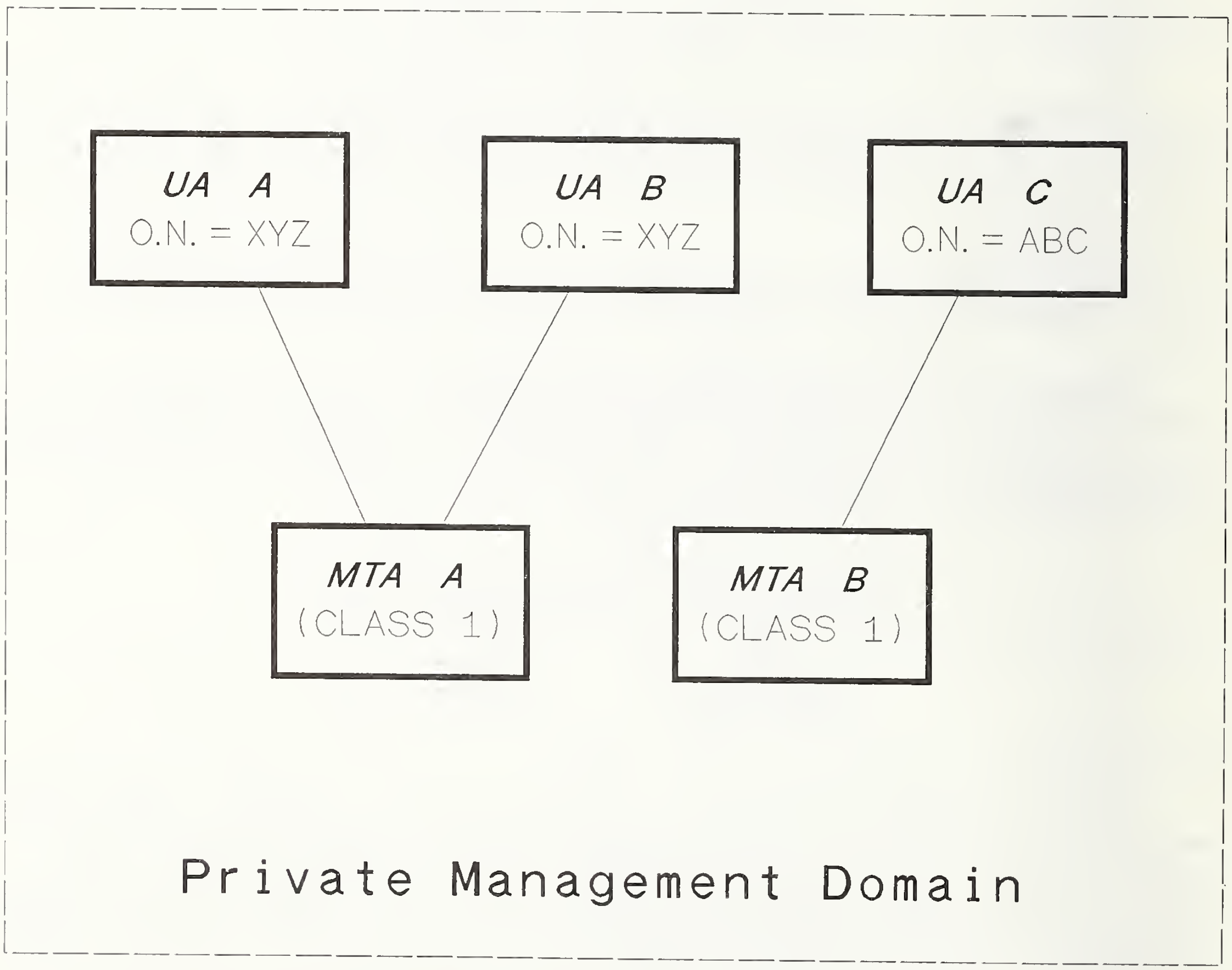


Figure 13. Class 1 MTA.

Class 2 MTAs base their intra-domain routing decisions on the Organization Name and Organization Units parts of the O/R name. This means that if an MTA is a Class 2 MTA, that MTA cannot make routing decisions on personal names. This message routing capability of class 2 MTAs is illustrated in figure 14. MTA A is a Class 2 MTA serving Organization Name XYZ, Organization Unit 001. In this diagram, even though UA A, UA B, and UA C have the same Organization Name, UA C can belong to a different MTA since it has a different Organization Unit. UA A and UA B have the same Organization Unit of 001, and therefore, must both be assigned to MTA A.

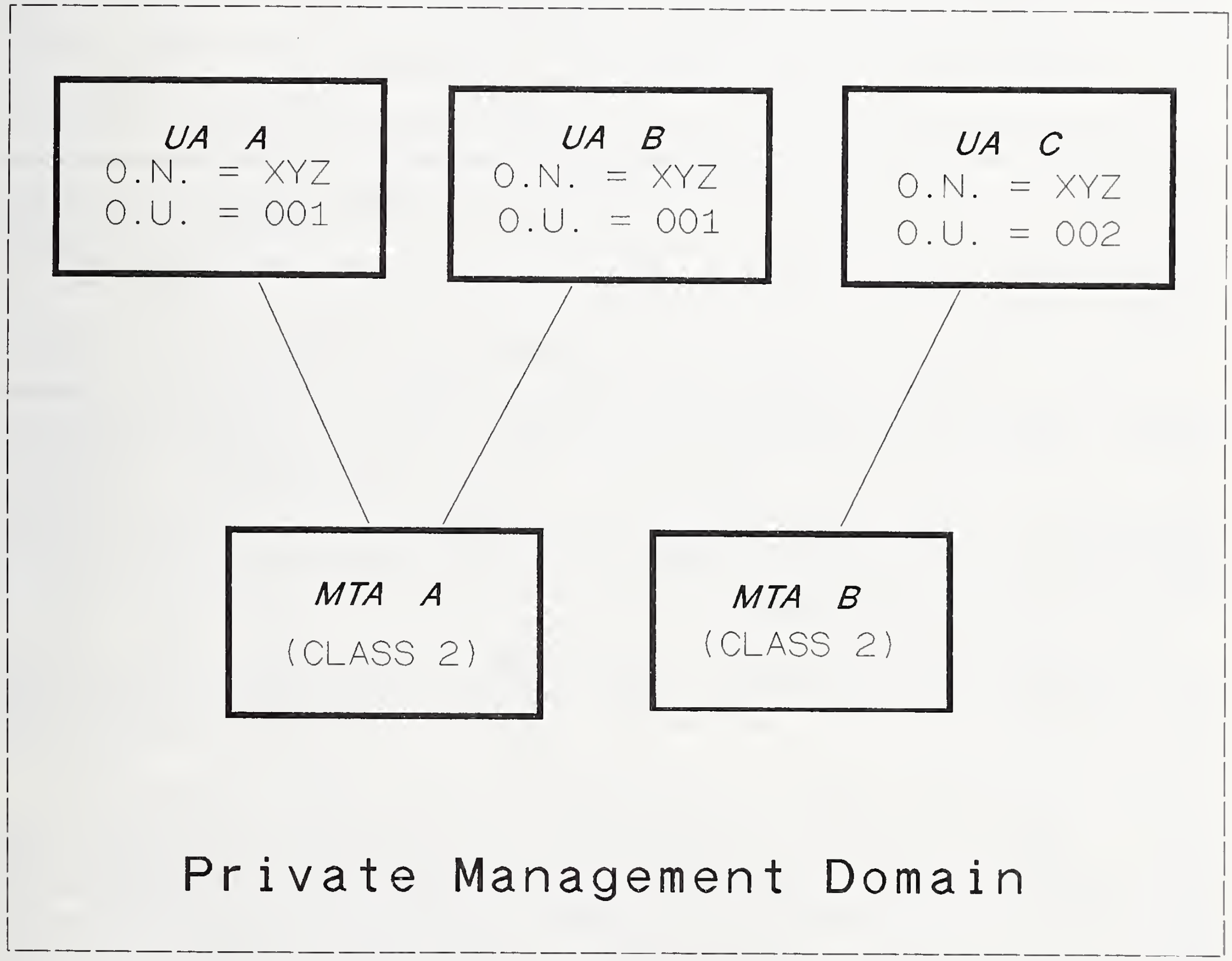


Figure 14. Class 2 MTA.

Class 3 MTAs, can base their intra-domain routing decisions on the Organization Name, Organization Units, and Personal Name parts of the O/R name. All classes of MTAs must be able to perform message delivery based on the six attributes listed previously; the designation of classes of MTA pertains to message routing capability only.

3.3.1.3 Optional UA Functions

The MHS standard contains certain functionality which the standard and the NIST Workshop Agreements do not require an MHS implementation to support. This functionality pertains to both the UA and MTA. In this section optional UA functions are described. The functions are presented in two lists. Each list is preceded by a description of the type of functions contained in the list. The functions in this section, as well as all functions described in the remaining sections of 3.3.1 are optional, and their provision by an MHS implementation should be taken into account when rating that implementation.

The first list contains functions that the originating UA is not required to generate. However, each UA is required to process these functions appropriately upon receipt.

The Authorizing Users function enables the originating UA to indicate to the recipient the names of one or more persons who authorized its sending. For example, an individual may authorize a particular action which is subsequently communicated to those concerned by another person such as a secretary. The former person is said to authorize its sending while the latter person is the one who sent the message (originator). This does not imply signature-level authorization.

The Auto-forwarded function enables the originating UA to request that a message be forwarded automatically by a recipient UA. Thus, the recipient can distinguish an incoming message containing a forwarded message in the body. As with a forwarded message, an auto-forwarded message may be accompanied by information (e.g., time stamps, indication of conversion) associated with its original delivery. When a UA auto-forwards a message, it designates it as auto-forwarded. If receipt notification has been requested for the message being auto-forwarded, the UA generates a non-receipt notification informing the originator of the auto-forwarding of the message.

The Blind Copy function enables the originating UA to provide the names of one or more additional users who are intended recipients of the message being sent. These names are not disclosed to either the primary or copy recipients. These names may or may not be disclosed to each other.

The Body Part Encryption function enables the originating UA to indicate to the recipient that any body part of the message being sent has been encrypted. Encryption can be used to prevent unauthorized inspection or modification of the body part. This service element can be used by the recipient to determine that some body part(s) of the message must be decrypted. However, this service element does not itself encrypt or decrypt any body part. This facility requires the use of bilateral agreements.

The Cross Referencing function enables the originating UA to associate with the message being sent the identifiers of one or more other messages. This allows the recipient's UA, for example, to retrieve from storage a copy of the referenced messages.

The Expiry Date function enables the originating UA to indicate to the recipient the date and time after which the originator considers the message to be invalid. The intent of this service element is to state the originator's assessment of the current applicability of the message. The particular action on behalf of a recipient by the UA or by the recipient himself is unspecified. Possible actions might be to file or delete the message after the expiry date has passed.

The Forwarded Message function enables the originating UA to send a forwarded message or a forwarded message plus its "delivery information" as the body (or as one of the body parts) of a message. An indication that the body part is forwarded is conveyed along with the body part. In a multi-part body, forwarded body parts can be included along with body parts of other types. "Delivery information" is information which is conveyed from the MTS when a message is delivered (for example, time stamps and indication of conversion). However, inclusion of this delivery information along with a forwarded message in no way guarantees that this delivery information is validated by the MTS. The Receipt and Non-receipt Notification service elements are not affected by the forwarding of a message.

The Importance function enables the originating UA to provide an indication to the recipients his assessment of the importance of the message being sent. Three levels of importance are defined: low, normal and high. The NIST Workshop Agreements state that appropriate action is for further study.

The Multi-Part Body function enables the originating UA to send to a recipient or recipients a message with a body that is partitioned into several parts. The nature and attributes, or type, of each body part is conveyed along with the body part.

The Obsolete function enables the originating UA to provide an indication that one or more messages he sent previously are obsolete. The message that carries this indication supersedes the obsolete message.

The Reply Request function enables the originating UA to request that a recipient send a message in reply to the message that carries the request. The originator can also specify the date by which the reply should be sent, and the O/R names of the one or more users to whom the reply should be sent. The recipient is informed of the date and names, but it is up to the recipient to decide whether or not to reply. A blind copy recipient should consider carefully to whom he sends a reply, in order that the meaning of the blind copy designation service element is preserved.

The Sensitivity function enables the originating UA to specify guidelines for the relative security of the message upon its receipt. It is the intent that the sensitivity indication should control such items as: (1) Whether the recipient should have to prove his identity to receive the message, (2) Whether the message should be allowed to be printed on a shared printer, (3) Whether a UA should allow the recipient to forward the received message, or (4) Whether the message should be allowed to be auto-forwarded. The sensitivity levels are (1) none, which implies no restriction on the recipient's further disposition of the message, (2) personal, which implies the message is sent to the recipient as an individual, rather than to him in his role, (3) private, which implies the message contains information that should be seen or heard only by the recipient and not by any one else, and (4) company-confidential, which implies that the message contains information that should be handled according to company-specific procedures.

The following list contains functions that the originating UA is not required to generate. Moreover, no specific processing action is mandated by the recipient UA. The user should be warned that the originator may have the capability to request one of these functions, but the recipient UA may not support the function, thus ignoring the request.

The Non-receipt Notification function enables an originating UA to request that it be notified that a message was not received by the intended recipient. It also enables a recipient UA to accept a notification that a message was not received by the intended recipient. For multi-recipient messages, this service element can be specified on a per-recipient basis.

The Receipt Notification function enables an originating UA to request that it be notified of the receipt, by a recipient, of a message being sent. It also enables a recipient UA to submit a receipt notification for a message. Requesting this function implicitly requests notification of message non-receipt.

3.3.1.4 Optional MTA Functions

Certain MTA functionality described in the MHS standard, although not required, may be provided by an MHS implementation. This section details these optional MTA functions.

The Access Management function enables an MTA to establish access with the UA, and manage information associated with access management. This function permits the UA and MTA to identify and validate the identity of each other. It provides a capability for the UA to specify its O/R name and to maintain access security. When access security is achieved through passwords, these passwords can be periodically updated. If the UA and MTA are co-resident, this function does not apply. Otherwise this function is optional in an MHS application.

The Alternate Recipient Assignment function enables an MTA to accept a request from the UA to have certain messages delivered to the UA when there is not an exact match between recipient attributes specified and the descriptive name(s) of the UA. Such a UA is specified in terms of one or more attributes for which an exact match is required, and one or more attributes for which any value is acceptable. For example, an organization may establish a UA to receive all messages for which country name, Administration management domain name and organization unit (e.g., company name) are an exact match but the personal name of the recipient does not correspond to an individual known by the MHS in that organization. This permits the organization to manually handle the messages to these individuals. In order for a message to be assigned to an alternative recipient, the originator must have requested the Alternative Recipient Allowed service element.

The Deferred Delivery function enables an MTA to accept instructions from the originating UA which indicate that a message being submitted should be delivered no sooner than a specified date and time. Delivery will take place as close to the date and time specified as possible, but not before.

The Deferred Delivery Cancellation function enables an MTA to accept instructions from the originating UA which indicate that a previously successfully submitted deferred delivery message should be canceled. The cancellation attempt may not always succeed. Possible reasons for failure are: deferred delivery time has passed, or the message has already been forwarded within the MTS.

The Disclosure of Other Recipients function enables an MTA to accept instructions from the UA, when submitting a multi-recipient message, to disclose the O/R names of all other recipient(s) to each recipient UA upon delivery of the message. The O/R names disclosed are as supplied by the originating UA. Receipt of this function by an MTA is mandatory; generation of this function by an MTA is optional.

The Explicit Conversion function enables an MTA to accept a request from the originating UA to perform a specific conversion, such as the conversion required when interworking between different telematic services. When a message is delivered after conversion has been performed, the recipient UA is informed of the original encoded information types as well as the current encoded information types in the message.

The Hold For Delivery Function enables an MTA to accept a request from the recipient UA that the MTS hold its messages for delivery until a later time. The UA can indicate to the MTS when it is unavailable to take delivery of messages and notifications, and also, when it is again ready to accept delivery of messages and notifications from the MTS. The MTS may indicate to the UA that messages are waiting due to the criteria the UA established for holding messages. Responsibility for the management of this service element lies with the recipient MTA. Criteria for requesting a message to be held for delivery are: encoded information type, maximum content length, and priority. The message will be held until the maximum delivery time for that message expires.

The Implicit Conversion function enables an MTA to accept a request from the UA to have the MTS perform for a period of time any necessary conversion on submitted messages. Neither the originating nor recipient UA explicitly requests this service element. If the encoded information type capabilities of the recipient UA are such that more than one type of conversion can be performed, the most appropriate conversion is performed. When a message is delivered after conversion has been performed, the recipient UA is informed of the original encoded information types as well as the current encoded information types in the message.

The Prevention of Non-delivery Notification function enables an MTA to accept instructions from the originating UA not to return a non-delivery notification to the originating UA in the event that the message being submitted is judged undeliverable.

The Probe function enables an MTA to provide support for a UA function to establish before submission whether a particular message could be delivered. The MTS provides the submission information and generates delivery and/or non-delivery notifications indicating whether a message with the same submission information could be delivered to the specified recipient UAs. The Probe function includes the capability of checking whether the message size, content type, and/or encoded information types would render it undeliverable. The significance of the result of a Probe depends upon the recipient UA(s) having registered with the MTA the encoded information types, content type and maximum message size that it can accept. This function is subject to the same delivery time targets as for the urgent class. The NIST Workshop Agreements state that PRMDs need not generate probes, and on reception, a PRMD must respond with a delivery report.

The Return of Contents function enables an MTA to accept a request from the originating UA that the content of a submitted message be returned with any non-delivery notification. This will not be done, however, if any encoded information type conversion has been performed on the message's content.

3.3.1.5 O/R Name Types

An O/R name is a descriptive name for a user that originates and receives MHS messages. Two forms of O/R names are described in the MHS standard. The first form contains three variants. An MHS implementation must support Form 1 Variant 1. The NIST Workshop Agreements do not support Variant 2, Variant 3 and Form 2.

The first form of MHS O/R name specifies the originator or recipient of a message based on the country, ADMD, and a subset of other attributes. Three variant representations are defined. Variant 1 consists of a country, ADMD, and a collection of one or more attributes chosen from PRMD Name, Organization Name, Organization Unit Names, Personal Name, and Domain-defined Attributes. Variant 2 allows O/R names to be entered from terminals equipped only with numeric keypads. It consists of a country, ADMD, UA Unique Numeric Identifier, and optionally Domain-defined Attributes. Variant 3 allows Telex terminals to be identified in the context of store-and-forward Telex, by use of the escape digit defined for Telex. It consists of a country, ADMD, X121 Address, and optionally Domain-defined Attributes.

The second form of O/R name specifies the originator or recipient of a message by identifying his telematic terminal. This form comprises the Telematic Address, and optionally, a Telematic Terminal Identifier. The Telematic Terminal Identifier might be, for instance, a TELEX AnswerBack string or a Teletex Terminal Identifier.

A user may find that an MHS implementation deviates from the MHS standard when assigning O/R names. For example, with Form 1 Variant 1, an implementation may mandate that all O/R names for users of that implementation contain an Organization Unit, even though the MHS standard identifies the Organization Unit as an optional O/R name attribute. Although this does not hinder interoperability, the user must decide if the deviation is significant based on his needs.

3.3.1.6 Message Body Parts

The NIST Workshop Agreements mandate that MHS implementations support the transfer of IA5 text body parts. Implementations may, however, generate and/or receive additional standardized body part types. These other body part types, which include TLX, Voice, G3Fax, TIF0, TTX, Videotex, Nationally Defined, Encrypted, ForwardedIPMessage, SFD, and TIF1, are listed in the NIST Workshop Agreements. In addition, the NIST Workshop Agreements provide recommended practices for the transfer by MHS systems of certain body part types that are not included in the X.400 standard, including binary data and Office Document Architecture (ODA) documents.

Certain user communities, such as the Computer-Aided Acquisition and Logistics Support (CALS) community, may want to use the MHS to transfer body part types that have not been recognized internationally. These body part types can be assigned a Private Message Body Part ID by a registration authority, which is currently the NIST. They can then be included in an interpersonal message if the MHS implementation is capable of transferring Private Message Body Parts.

3.3.1.7 Composing Messages

This category begins the description of non-standard MHS functionality which may be provided by an implementation. The provision of functions in this section, as well as functions in the remaining sections of 3.3.1, is where most MHS implementations will vary. The first non-standard functional category to be described is message composition.

Text editors are used for entering the content of a message. An implementation may provide an internal text editor for this purpose. Internal editors exist solely within the MHS application. Additionally, a user may be given access to external editors. External editors reside on the system supporting the MHS implementation. Examples of external editors include: WordPerfect within a MS-DOS environment, VP Document Editor within a Xerox environment, EDT within a VMS environment, and Vi within a UNIX environment. Both internal and external editors may take the form of line editors, full screen editors, or word processors. A user may be allowed to select which available editor to use when composing a message. One limitation an editor may place on a user is maximum message size. For example, an editor may limit the user to only composing messages of size 5000 characters or less.

In addition to entering message text, users may add attachments to a message being composed. These attachments may be files on the local file system, or messages the user has saved. For example, if a user wants to send an updated document to some recipient, the user could compose a message stating that an update to the original document existed, then attach a copy of the update to his message. Once added, the user may be allowed to modify the attachment.

When composing messages, a user enters values for certain MHS message options, such as message priority, subject, etc. To facilitate this, the user may be provided with a "fill-in form." This form contains all MHS options for which the user can enter a value. Using a fill-in form, the user need not be burdened with remembering which options are available. In addition, the MHS implementation may provide tables listing possible values for the message options. For example, a user may be allowed to create a table containing the recipients to which the user frequently sends mail. The user can scroll through the table and select the recipients, as opposed to entering their O/R names in the fill-in form. Also, a user may be allowed to specify whom the message is from. This function could be useful, for example, when a secretary sends a message for another person.

When message composition is completed, the user may have the option of sending the message immediately, or saving the message for possible future revision.

3.3.1.8 Sending Messages

MHS implementations may provide users with various functions pertaining to sending MHS messages. The user may save a copy of all outgoing messages in a mail storage unit. The user may send one or more messages saved in a storage unit with one command. When replying to a message, a user may return the contents of the original message with the reply, or reply to all O/R names referenced in the message, as opposed to just the sender. When forwarding a message a user may prepend an explanatory note to the forwarded message, or modify the message before forwarding it. A user may specify instructions for outgoing messages with invalid recipients. Possible instructions would be to send the message to all valid recipients, or not to send the message unless all recipients are valid. Also, a user may send a file from the local file system as a message or as a reply to a message.

3.3.1.9 Aliases

An alias can be defined as a symbolic representation of an O/R name. For example, the alias "John_S" can be created to represent the following:

> Country = US
> Administration Management Domain = AT&T
> Private Management Domain = Private_Mail_System_X
> Organization Name = Company_Y
> Surname = Smith
> Givenname = John

The originator of an MHS message can enter "John_S" as the recipient, as opposed to entering the entire O/R name shown above.

Two types of aliases may be available to an MHS user: personal and system. Personal aliases are created by a user, and can be accessed only by that user. System aliases are created by a system administrator, and can be employed by all users on that system. Functions relating to personal aliases include: creating, displaying, modifying, and deleting aliases, as well as creating multiple aliases for a single recipient, expanding aliases on the screen so as to view the O/R name represented by the alias, and creating aliases from the O/R names of incoming MHS messages. Functions relating to system aliases are similar to those of personal aliases, except that only a system administrator can create, modify, or delete system aliases. Both personal and system aliases may reference recipients that reside on the same MTA as the originating user, as well as recipients residing on different MTAs.

Personal and system aliases provide limited proprietary Directory Services type functions to MHS implementations. The CCITT X.500 Recommendation standardizes a set of Directory Services functions, which may be accessed by MHS implementations. These functions include a white pages type service to obtain a recipient's O/R name. Currently, only a limited number of commercial OSI Directory Service Implementations are available to an MHS user.

3.3.1.10 Distribution Lists

A distribution list can be defined as a group of recipients referenced by a name. For example, the distribution list "Managers" could contain the O/R names of all the managers in company XYZ. If the president of this company needed to send a message to his managers, he could enter the name "Managers" as the recipient of the message, as opposed to listing their O/R names individually.

In the above example, "Managers" consisted of a list of fully specified O/R names. In addition to O/R names, distribution lists may contain aliases or other distribution lists. The term "nested distribution list" is used to describe a distribution list containing an entry which is another distribution list.

Two types of distribution lists may be available to an MHS user: personal and system. As with personal aliases, personal distribution lists are created by a user, and can be accessed only by that user. System distribution lists are created by a system administrator, and can be employed by all users on that system. Functions relating to personal distribution lists include: creating, displaying, modifying, deleting, and nesting distribution lists. In addition, if a recipient is referenced more than once in a distribution list, the MHS implementation may recognize the duplication, and send only one copy of the message to the recipient. Functions relating to system distribution lists are similar to those of personal distribution lists, except that only a system administrator can create, modify, or delete system distribution lists. Both personal and system distribution lists may reference recipients that reside on the same MTA as the originating user, as well as recipients residing on different MTAs. As with aliases, the use of personal and system distribution lists is additional to any Directory Services functionality provided by the MHS implementation.

3.3.1.11 Receiving Messages

An MHS implementation may notify a user that an MHS message has arrived. Notification can occur immediately upon arrival, or after a specified time interval has elapsed. An example of the second scenario would be the UA querying the MTA for new messages every 5 minutes. If an MHS message arrives for a user, the user will not receive notification until the 5 minute interval has elapsed. In an implementation where the MTA is polled for new messages, a user may set the polling frequency, or manually query the MTA at any given time. Notification of new mail can take the form of a sound, such as a beep, or a message displayed to the screen. A user may select the method of notification.

Once received, a user may be allowed to specify a set of criteria used to discriminate messages, such as by sender, or by character strings occurring in the message body. Message discrimination can be used to separate junk mail from other incoming mail. For example, if a user receives messages from the company "Bogus Computers" about products, most of which do not interest him, the user can have all messages from "Bogus Computers" placed into a certain mail storage unit, which he can read at his leisure. Finally, an MHS implementation may allow a user to receive body parts not recognized or supported by the implementation. This is useful if the user has appropriate software to convert the body parts into an understandable form.

Occasionally a user may be unable to read his messages. One such situation could be if the user is on vacation. To deal with this circumstance, an MHS implementation may provide an automatic reply mechanism. This mechanism will reply automatically, with a message precomposed by the user, to any message sent to that user. Using the automatic reply, a person could notify anyone sending him mail that he is out of the office, and will read his mail when he returns on a specific date. In addition to an automatic reply function, an MHS implementation may allow a user to redirect messages. By redirecting messages, a user can have his messages forwarded automatically to another address. For example, if a user is on vacation, he can have his messages redirected to a colleague. Also, an MHS implementation may allow a user to specify a secondary address, in addition to the address used primarily by the user. If, for any reason, the user's primary address is inaccessible, the implementation will attempt to deliver incoming messages to the secondary address.

3.3.1.12 Listing Summary of Messages

When listing a summary of MHS messages, only portions of the message header, such as the subject, sender, and delivery time are shown; the message contents are not displayed.

Typically, one line of information is displayed per message. Part of that information may include indicators for new mail, forwarded mail, multiple body parts, converted body parts, etc. In addition, statistical information about the message, such as whether it has been read previously, whether a reply is requested, and whether it is a user message or status report message may be displayed.

A user may be allowed to list messages according to various criteria. Examples of these criteria are listing all messages in a specific mail storage unit, messages received before a certain time, messages not yet read, messages of a specific importance, messages containing certain text in the body part, or messages to which replies are requested. A user may obtain listings of mail sorted alphabetically by message subject. In addition, one or more messages may be selected from the message summary on which to perform certain operations such as reading, deleting, matching delivery reports to messages, etc.

3.3.1.13 Reading Messages

An MHS implementation may allow a user to read messages in any order. With a single command, a user may read multiple messages conforming to certain criteria, such as all messages not yet read, messages received before a certain time, messages of a specific importance, messages to which replies are requested, or messages saved in a certain mail storage unit. While reading a message, a user may modify the message, or redraw the screen in case noise, nonprintable characters, or a notification, such as a new mail announcement, appears on the screen making the current message unreadable. A user may also be allowed to specify a mail storage unit into which all new mail, once read, will be transferred.

3.3.1.14 Saving Messages

An MHS implementation may provide a facility for saving MHS messages. This facility can be structured in different ways. It may take the form of a single storage unit in which all messages must be saved. The facility may be partitioned into multiple storage units in which a user can save mail based on various criteria. These multiple storage units are often referred to as "drawers." Finally, the facility may be partitioned into hierarchical storage units. A graphic example of a hierarchical facility for saving mail is a file cabinet. Messages can be placed into "folders," which comprise one level of storage, and folders can be placed into drawers. Drawers and/or folders are useful for organizing mail that a user wishes to retain. For example, a user working on three projects concurrently, may receive MHS messages relating to any one of the projects. Depending on the MHS implementation, this user could create three drawers, one for each project, and save messages received in the appropriate drawer. If Project 1 involved releasing experimental software and related documentation, the user could create two folders inside the Project 1 drawer, one for messages concerning the software, and one for messages containing comments on the documentation. If a message received contained comments on both the software and documentation, a copy of this message may be saved in both folders.

An MHS implementation may supply a user with default mail storage units. Default mail storage units are created by the MHS implementation, and will exist when the user first enters the mail application. Default mail storage units generally include a drawer or folder for receiving messages, and a drawer or folder used when sending messages. In addition, an MHS implementation may allow a user to create, display, rename, make copies of, purge the contents of, and delete his own mail storage units. Note that if a mail storage unit is copied, all the messages saved inside the storage unit are copied. Also, when displaying a storage unit, an MHS implementation may provide information such as its size, the number of saved messages, etc.

A user may be provided with functions that relate to the messages stored in a mail storage unit. Messages saved inside a drawer or folder may be listed, sorted, edited, or saved in a file on the local file system. Also, messages may be moved or copied from one mail storage unit to another.

3.3.1.15 Deleting Messages

The deletion of MHS messages can be managed in different ways. One method of deleting mail is to discard messages immediately upon receipt of the delete command. Mail deleted in this fashion cannot be recovered. An alternate method employs a storage area for deleted mail, often called the "wastebasket." Upon receipt of the delete command, a message is transferred to the wastebasket. The contents of the wastebasket are not discarded until the user issues a second command to empty the wastebasket, or until the user exits the mail application. Messages placed in the wastebasket may be recovered, that is transferred from the wastebasket to a permanent storage location, any time before the wastebasket is emptied.

3.3.1.16 Printing Messages

An MHS implementation may allow a user to print, or obtain paper copies of messages. In addition, a user may print other information, including listings of local users, aliases, and distribution lists. Certain options, such as the printer to which the output is directed, the date and time the printing is to begin, whether the output should be single or double spaced, and the number of copies to be printed, may also be specified with the print command.

3.3.1.17 Default Configuration Profile

An MHS implementation may provide a user with a default configuration profile containing MHS message options for outgoing messages. For example, one entry in the profile may be "sensitivity," and its default value "company confidential." When composing a message, if a user does not enter a value for message sensitivity, the default value of company confidential will be set for that outgoing message. Examples of MHS message options which may be contained in the default configuration profile include: specific O/R name attributes such as country, ADMD, and PRMD, grade of delivery, message importance, message sensitivity, original encoded information type, content type, delivery report type, alternate recipient allowed, return of contents, conversion prohibited and reply request. In addition, when sending a message, an implementation may automatically set the user's O/R name to be the originator of the message.

In addition to MHS message options, an implementation may provide default settings for a user's working environment. These settings could include: the editor used when composing a message, whether a copy of all outgoing messages is saved in a mail storage unit, the mail storage unit in which new mail is placed, and the mail storage unit in which deleted mail is placed. There are many other functions that may be included in this profile.

The original default values for entries in the MHS configuration profile are set by the MHS implementation. The implementation may, however, allow the user to display and/or modify these settings.

3.3.1.18 On-Line Help Facilities

An MHS implementation may provide a user with an on-line help facility within the mail application. Using the help facility, a user can obtain information about the mail application, such as available commands, how the commands are used, and the parameters and options supported by the commands.

3.3.1.19 UA Interface

The UA interface, provided by the MHS implementation, may be window-driven or command-driven. With a command-driven interface, menus may be employed to prompt the user for information. With either interface, the user may be allowed to select the level of sophistication, such as beginner, intermediate or advanced. For example, the beginner level may display only the basic options such as send, read, and save. The intermediate level may add more sophisticated options such as print, delete and recover. The advanced level may display all the options. All options should be available to the user, regardless of the level, even if they are not displayed. The interface to the UA is an important function, because it greatly influences the implementation's ease of use.

In addition to the UA interface provided by the MHS implementation, a user may be given programmatic interfaces to the MTA and UA. Using the MTA programmatic interface, the user can develop an application that employs the message transfer service. Using the UA programmatic interface, the user can develop an application that employs UA functionality, such as submitting and accepting the delivery of messages. If provided, the MTA and UA programmatic interfaces may be POSIX conformant. POSIX is a Portable Operating System Interface for Computer Environments standard sponsored by the Technical Committee on Operating Systems of the IEEE Computer Society. An application running in a POSIX environment can be ported to other POSIX systems with minimal changes. POSIX MHS interfaces to the UA and MTA are expected to be standardized by the end of 1990, and MHS implementations using these interfaces should appear shortly thereafter.

3.3.1.20 System Interface

From within an MHS implementation, a user may be allowed to execute operating system commands. MHS messages, when appropriate, may be used as input to these commands. For example, if a user is mailed a document, the user could execute a program on the operating system that checks for spelling errors, using the document received as input to the program. A user may also employ the receipt of MHS messages as a triggering mechanism. In other words, the user can have a program on the operating system begin execution when an MHS message is received.

In addition, a user may be given access to the local file system. A user may be allowed to save MHS messages in files on the file system, create files on the file system to use as body parts for outbound messages, or send existing files as outbound messages or as replies to messages. Files mailed by the user may contain binary data, as well as text.

3.3.1.21 Administrative Functions

Sections 3.3.1.21 through 3.3.1.24 detail functionality relating to the MTA. The functions in these sections primarily effect a system administrator, who is responsible for managing and maintaining the MTA.

MTA administrative tasks begin with the installation of the MHS implementation. An implementation may provide an on-line training program for installing and configuring the MTA. To ensure the installation was performed correctly, an installation verification facility may be provided. The MTA may run in a local mode, in which all submitted messages can only be delivered to UAs connected to the MTA. This function is useful when the implementation is first installed and configured.

Once installation is complete, certain information must be registered with the MTA before MHS message transfer can commence. Local information generally registered with the MTA includes: local users who are to send or receive MHS messages, and MTA authentication information such as the MTA name and MTA password. In addition, if the MTA is to connect with remote MTAs, routing information pertaining to the remote MTAs must also be registered.

An MHS implementation may provide the system administrator with certain MTA statistics such as the number of messages sent and received from remote MTAs, the throughput of these messages, the time delay between the arrival of messages at the MTA and the retrieval of these messages by the UA, etc. If the statistics show the MTA to be overloaded, the system administrator may modify specific MTA parameters to control message traffic and optimize the MTA. These parameters include the frequency of retransmissions, the maximum transfer time for which a message can be delivered before a non-delivery notification must be returned, and the maximum size of an MHS message. A system administrator may also specify the time of day during which messages of a specific priority will be sent. For example, the system administrator may have the MTA send all low priority mail after usual working hours. This can ease the work load of the MTA during the time when the bulk of MHS messages are transferred.

MTA maintenance includes purging old messages from queues, purging old log files, checking for the existence of certain files, etc. An MHS implementation may perform maintenance functions automatically. In addition, the system administrator may be allowed to manually perform selective maintenance tasks. Whether automatic or manual, an implementation may log information on all maintenance performed.

To carry out administrative functions, such as the ones described above, an MHS implementation may provide a utility program. With this utility program, the system administrator may view and update MTA and user information, modify MTA parameters, and perform manual maintenance. An implementation may provide an on-line help facility, so that the system administrator may view information about the MTA such as available commands, how they are used, and the options and parameters supported by the commands.

An MHS implementation may also provide other administrative functions. A system administrator may be able to backup the MHS implementation, or restore the implementation from a backup. The backup may be restored to a different machine if the original machine is encountering hardware problems. Restoring an implementation is different from installation in that information registered with the MTA need not be reentered. Also, an implementation may allow a copy of every inbound or outbound message to be saved on the MTA.

3.3.1.22 Remote MTA Connections

Certain MTA functionality relates to MTA connection establishment. For example, an MTA may only possess the capability to connect to one remote MTA. In this scenario, the one remote MTA would most likely belong to an ADMD, through which all MHS messages could be routed. In contrast, an MTA may connect to multiple remote MTAs. In this new scenario, the MTA could establish connections with MTAs belonging to PRMDs, as well as ADMDs. The MTA may also support multiple simultaneous connections. The number of simultaneous connections may be modified by the system administrator to control message traffic.

An MTA may establish a connection with a remote MTA for the purpose of relaying MHS messages. If, for example, an MTA receives a message destined for a different MTA, the MTA may route the message to the destination MTA. To accomplish this, the relaying MTA must possess routing information for the destination MTA.

When establishing MTA connections, certain MTA options may be set on a per MTA basis. One such option is transport class. For example, an MTA may connect to one remote MTA using transport class 4, and connect to a different remote MTA using transport class 0. Other options that may be set on a per MTA basis include allowing/disallowing inbound or outbound connections, allowing/disallowing MTA relaying, and MTA authentication parameters such as MTA name and MTA password. In addition, once a connection is established, an MTA may keep the connection open for a certain length of time in case further messages, such as status report messages, are to be transferred. The duration of time that the connection is kept open may be modified by a system administrator.

3.3.1.23 Access Control

An MHS implementation may provide functions which limit access to the MTA. Typically, users must be registered with the MTA to send and receive MHS messages. A system administrator may further limit MTA access to a subset of the registered users, or may limit any specific user to only sending or receiving messages. Also, a system administrator may give himself exclusive access to the MTA. Such a scenario could occur if the system administrator needed to prevent use of the MTA while performing an MTA backup.

3.3.1.24 Debug Capabilities

An MHS implementation may provide functions which aid a system administrator in resolving problems encountered. The problems referenced here deal primarily with message routing (i.e., why an MHS message is not delivered to the recipient specified). To aid a system administrator, an implementation may provide a log file containing information on message traffic, as well as any errors encountered. A system administrator can monitor the status of the mail system, or view various message statistics, such as the date and time of the last successful and unsuccessful message transfer, the range of message sizes, the number of forwarded messages, the number of rejected messages, etc. An implementation may also provide a facility that will verify whether an O/R name can be routed to another MTA.

If an outbound message fails to be delivered, an MTA may queue the message for retransmission. The system administrator may be allowed to access the queued message. The system administrator may force a retransmission of the message, disable retransmissions of the message, or delete the message from the MTA queue. Also, the system administrator may analyze the message by translating it into English or ASN.1 format.

In addition to routing errors, other problems, such as those involving system resources may occur. Notification of these problems may be written to the log file, displayed on the console device, or detailed in a message which is sent by the MHS implementation to the system administrator. In addition to notification of problems, an MHS implementation may provide an MTA test function. This function can be used to ensure that all required software is running, and that the MTA is working satisfactorily.

3.3.1.25 Underlying OSI Layers

In addition to UA and MTA functions, an MHS implementation may provide additional functionality in the underlying OSI layers. Session Layer functionality determines the Session Layer implementations with which the MHS implementation can interoperate. The MHS standard specifies that the following Session services be used: Session Connection, Normal Data Transfer, Orderly Release, U-Abort, P-Abort, User Exception Reporting, Provider Exception Reporting, Activity Start, Activity Resume, Activity End, Activity Interrupt, Activity Discard, Please Tokens, Give Tokens, and Give Control.

Transport Layer functionality determines the Transport Layer implementations with which the MHS implementation can interoperate. Specifically, this function specifies if the MHS implementation can interoperate with other MHS implementations using Class 0 Transport or Class 4 Transport.

Network layer functionality determines the Network Layer implementations with which the MHS implementation can interoperate. Specifically, this function specifies if the MHS implementation can interoperate with other MHS implementations using: the Connection Oriented Network Service over X.25, the Connectionless Network Service over X.25, or the Connectionless Network Service over a local area network such as the 802 family.

If the MHS implementation is to be used over an X.25 subnetwork, the X.25 software provided may conform to the 1980 or 1984 version of the CCITT X.25 protocol. Additionally, an X.25 prototype may be provided to assist the system administrator with establishing X.25 parameter values. For example, if the MHS implementation will transfer messages over ACCUNET, the Wide Area Network (WAN) provided by AT&T, the system administrator could choose the ACCUNET prototype. Included in this prototype are all X.25 parameter values needed to establish X.25 ACCUNET connections. The provision of these values will decrease installation time and optimize X.25 performance. Once established, the system administrator may modify X.25 parameters, to better suit any individual requirements.

Whether MHS message transfer occurs over a WAN or a LAN, certain statistical information, such as the number of bytes sent and received, the number of packets sent and received, and various error counters, may be available. In addition to all the functionality described above, facilities to optimize performance or troubleshoot interworking problems within the underlying OSI Layers may be provided.

3.3.1.26 Certification

Certification can be divided into two sections: conformance and interoperability. Conformance testing, which is required by GOSIP, verifies that the implementation conforms to the standard. Interoperability testing verifies that the implementation interoperates with other implementations. Interoperability testing with the NIST Reference implementation is required by GOSIP, if the NIST Reference implementation is available. Interoperation testing with other implementations is optional, but strongly recommended. Information such as which tests were performed, with whom, when, as well as the actual test results should be made available to the user. For more information on certification, the user should read the "GOSIP Conformance and Interoperation Testing and Registration" document, which is currently a draft Federal Information Processing Standard.

3.3.1.27 Hardware Requirements

Different MHS implementations may require specific hardware configurations for operation. For example, a UA and MTA may reside on the same machine, or may require separate machines which are interconnected. The same is true for the MTA and RTS. In addition, requirements pertaining to the CPU, disk space, memory, and external devices (e.g., X.25 interface cards) may need to be met.

3.3.1.28 Software Requirements

Different MHS implementations may require specific software configurations for operation. For example, an MTA may consist of multiple software components which need to be installed separately. OSI software, not residing in the Application Layer, such as Session software and lower layer software may require installation. In addition, an MHS implementation may need a specific operating system and version. If the MHS mail system is implemented as a gateway between a proprietary mail system and the MHS messaging network, as opposed to a pure MHS application, the proprietary mail system may need to be installed.

3.3.1.29 Documentation

An MHS implementation may provide a user with a variety of manuals explaining the interworkings of the mail application. Although each implementation may organize its documentation differently, the following information, regardless of format, may prove useful to the user: installation guide, user's guide, administration guide, troubleshooting guide, MHS reference guide, and quick reference guide. An installation guide provides information for installing and configuring the MHS implementation. It may contain sample installations and lists of files which are created on the local file system. A user's guide describes how the implementation is used, and the administration guide, which is written for the system administrator, details management and maintenance of the MHS implementation. The troubleshooting guide describes possible errors, and how they are corrected. The MHS reference guide lists MHS functions supported by the implementation, and the quick reference guide, which is useful once the user is familiar with the implementation, provides a quick reference for mail system operations.

3.3.1.30 Pragmatic Constraints

The NIST Workshop Agreements list UA pragmatic constraints, such as maximum message subject size, maximum number of recipients per message, etc. These constraints are large enough to accommodate any typical application. An MHS implementation may place additional pragmatic constraints on the MTA. These constraints, such as the maximum number of UAs supported by the MTA, the maximum number of users registered with the MTA, the number of ADMDs to which the MHS implementation can be connected, the number of messages and duration of time for which messages can be stored at the MTA, are not addressed in the NIST Workshop Agreements.

3.3.2 Mandatory Functional Requirements

This section recommends a procedure for eliminating the candidate MHS implementations which do not meet the mandatory functional requirements of the user. The user should create a list of any functions which must be included in a candidate MHS implementation, for that implementation to be acceptable. The user should be careful not to list as mandatory, functions which are instead highly desirable because this can unnecessarily restrict the list of candidate MHS implementations. This list may be created by reviewing the functions in each of the functional categories, noting which functions are mandatory for the user. Once the list is created, the user should verify that each of the candidate MHS implementations contains the mandatory functions. A candidate MHS implementation which does not contain all of the mandatory functions should be removed from the list of implementations at this point. As an example, if after reviewing the functional categories, the user decides that the candidate MHS implementations must support connections to both an X.25 Wide Area Network and an 802.3 Local Area Network, then MHS implementations which do not support connections to both X.25 and 802.3 are unacceptable to the user and should no longer be evaluated.

3.3.3 Functional Evaluation

This section recommends a procedure for determining which of the candidate MHS implementations, possessing all required functions, best meets the functional capabilities desired by the user. First, the user must assign weights to each category of functions and to each function within that category based on how important the category of functions and each function within that category is to the user. This procedure is defined in section 3.3.3.1. Second, the user must rate each of the candidate MHS implementations, based on the weights assigned by the user and what functionality is available in the candidate MHS implementations. This procedure is defined in section 3.3.3.2.

3.3.3.1 Weighing Functions

This section provides guidelines for assigning weights to each category of functions and to each function within a category based on how important the category of functions and each function within the category is to the user. First, the user should select one of the three suggested options for weighing the functions within each category. The options are:

(1) Determine a weight for each individual function in a category based on how important that function is to the user. As an example, let us assume that a category contains 20 functions and that the user has decided that functions 1-5 are very important, functions

6-10 are moderately important, functions 11-15 are slightly important, and functions 16-20 are not of any importance. Then the user may assign a weight of 3 for functions 1-5, a weight of 2 for functions 6-10, a weight of 1 for functions 11-15, and a weight of 0 for functions 16-20.

(2) Assign each function in a category, which is of interest to the user, a weight of 1, and all other functions in the category a weight of 0. This option assumes that the user is either interested or not interested in a function, and that each of the functions of interest to the user are of equal importance. As an example, let us assume that a category contains 10 functions and that the user is interested only in functions 1-5. Then the user would assign a weight of 1 for functions 1-5, and a weight of 0 for functions 6-10.

(3) Do not assign any weight to the functions in the category.

The user must decide which option to use for evaluating each category, based on the tradeoffs of the three options. An evaluation of a category based on option 1 is the most precise and the most time consuming of the three options. An evaluation of a category based on option 3 is the least precise and the least time consuming of the three options. The user should note that since the functions within one category are rated independently of the functions within the other categories, different rating options may be selected for different categories, based on the user's preferences.

Second, the user should balance each of the categories, and determine how important each category is to the user. This step results in the assignment of a weight to each category.

The categories must be balanced because the maximum score which may be received by one category has no relation to the maximum score which may be received by the other categories. For example, one category may be scored as the sum of the number of functions available in the implementation. If there are 10 functions in the category then the category can receive a maximum score of 10. Another category may be subjectively scored as 1 if the implementation acceptably performs the functions in this category; otherwise it will receive a score of 0. If these two example categories are of equal importance to the user, then the weight of the second category must be 10 times as large as the weight of the first category, in order to balance the categories.

The user must determine relative levels of importance of the categories. The user first assigns a maximum score to each category to reflect its importance to the user. For example, the user may decide that a category which is extremely important to the user can receive a maximum score of 400 points, a category which is very important to the user can receive a maximum score of 300 points, a category which is important to the user can receive a maximum score of 200 points, a category which is less important to the user can receive a maximum score of 100 points, and a category which is not of any importance to the user can only receive a score of 0 points.

The user must then compute a weight for each category. The weight for each category is calculated by dividing the maximum score the user has determined that the category can receive by the maximum score of the functions within the category. As an example the user may consider the categories of sending messages and receiving messages to be extremely important (400 points), the category of saving messages to be fairly important (300 points), the category of aliases to be important (200 points), the category of system interfaces to be less important (100 points), the category of default configuration profile to be of no importance (0 points). If the maximum score of the functions in the sending messages category is 10, then it is assigned a weight of 400/10 which equals 40. If the maximum score of the functions in the receiving messages category is 20, then it is assigned a weight of 400/20 which equals 20. If the maximum score of the functions in the saving messages category is 10, then it is assigned a weight of 300/10 which equals 30. If the maximum score of the functions in the aliases category is 5, then it is assigned a weight of 200/5 which equals 40. If the maximum score of the functions in the systems interfaces category is 1, then it is assigned a weight of 100/1 which equals 100. If the maximum score of the functions in the default configuration profile category is 10, then it is assigned a weight of 0/10 which equals 0.

3.3.3.2 Functional Evaluation Rating

This section recommends a procedure to functionally rate the candidate MHS implementations, based on the weights assigned by the user and what functionality is available in the candidate MHS implementations. In order to determine which of the functions in a category are available in a candidate MHS implementation, information such as product literature, users manuals, technical specifications, etc. should be obtained from the vendors. This functional rating procedure must be performed on all of the categories described in this document, and the procedure must be repeated for each candidate MHS implementation. The procedure for determining a functional rating for each candidate MHS implementation follows.

First, the user must score each category of a candidate MHS implementation. The procedure for scoring a category varies, depending on which of the three previously described weighing options is chosen.

If option 1 is chosen, then the score of the category is the sum of the weight of each function which is present in the implementation. For example, using the weights defined for the 20 functions given in the example of option 1 in section 3.3.3.1, if functions 1, 2, 3, 6, 7, 15 are present in an implementation, then that implementation would receive a score of 3 + 3 + 3 + 2 + 2 + 1 which equals 14.

If option 2 is chosen, then the score of the category is the sum of the number of functions, which are important to that user, and are present, in the implementation. For example, using the weights defined for the 10 functions given in the example of option 2 in section 3.3.3.1, if functions 1, 2, 3, 6, 7 are present in an implementation, then that implementation would receive a score of 1 + 1 + 1 + 0 + 0 which equals 3.

If option 3 is chosen, then the user selects one of two recommended subjective scoring methods. First, the user can subjectively score the category based on the user's overall impression of how well that category is represented in the candidate MHS implementation. As an example, the user may decide to rate the functionality of a category, contained in a candidate MHS implementation, on a scale of 0-10 where a score of 10 indicates that the candidate MHS implementation contains all of the functionality in the category that the user requires, a score of 5 indicates that the candidate MHS implementation contains an average amount of functionality in the category that the user requires, and a score of 0 indicates that the candidate MHS implementation does not contain any of the functionality in the category that the user requires. Second, the user can subjectively score the category based on the users overall impression of whether or not the candidate MHS implementation acceptably performs the functions in this category. If the candidate MHS implementation acceptably performs the functions in this category it will receive a passing score (or 1); otherwise it will receive a failing score (or 0).

The user must repeat this procedure for scoring categories, on each category. The equations for rating categories, using each of the four options, are specified in table 3.

Table 3. Equations for Rating Categories of Functionality

Option 1: $$C = \sum_{i=1}^{n} (F_i * WF_i)$$

Option 2: $$C = \sum_{i=1}^{n} (F_i)$$

Option 3: $$C = S$$

$C = Score\ of\ Category$

$F_i = 1$ if $Function\ i\ is\ Present\ in\ Implementation$, 0 $Otherwise$

$WF_i = Weight\ of\ Function\ i$

$S = Subjective\ Score\ of\ Category$

$n = Number\ of\ Functions\ in\ Category$

Second, the user determines the total functional rating, for a candidate MHS implementation, by summing the weight of each category times the score for that category, for each of the categories. For example if there are categories X, Y, and Z and category X has a weight of 5, category Y has a weight of 3, and category Z has a weight of 1, and an implementation has a score of 25 for category X, a score of 50 for category Y, and a score of 75 for category Z, then the functional rating for the implementation is ((5*25) + (3*50) + (1*75)) which equals 350. The equation for determining the total functional rating is specified in table 4. The candidate MHS implementation with the highest score is likely to be the "best" implementation, functionally, for that user. Note that these ratings are not absolute ratings; another user might rate the same candidate MHS implementations differently based on a different set of requirements.

Table 4. Equation for Functionally Rating MHS Implementations

$$RF = \sum_{i=1}^{m} (C_i * WC_i)$$

$RF = Total\ Functional\ Rating$

$C_i = Rating\ of\ Category\ i$

$WC_i = Weight\ of\ Category\ i$

$m = Number\ of\ Categories$

3.4 Performance Evaluation Guidelines

This section recommends a procedure to evaluate the performance of candidate MHS implementations. It is divided into three sections. Section 3.4.1 contains experiments which measure the performance of MHS implementations. Section 3.4.2 recommends a procedure for eliminating candidate MHS implementations which do not meet mandatory user performance requirements. Section 3.4.3 recommends a procedure for determining which remaining candidate MHS implementation best meets the performance requirements of a user.

Performance may not be an issue for all users. Since there are costs and time involved in evaluating the performance of candidate MHS implementations, it is recommended that performance only be evaluated if it is an issue to the user. The following may assist a user in determining whether performance is an issue.

Some users are not concerned with the time required to transfer a message to the recipient. For example, a clerk at a mail order house may continuously send out junk mail. The user in this example is not concerned whether a message takes 10 or 30 minutes to be transferred to the recipient. Other users may be more concerned with the time required to transfer a message to the recipient. For example, a manager of a field service organization may send mail to offsite employees containing urgent service requests. The user in this example would like the messages to be delivered as quickly as possible. Thus, the urgency of the user's mail may determine whether performance is an issue.

Overall mail system usage can also determine whether performance is an issue. An example of a mail system which is minimally used is one consisting of 50 users, and each user sends and receives an average of one 1024 byte message per day. Performance may not be an issue for this mail system, which sends or receives an average of only one hundred 1024 byte messages per day, because any MHS implementation should be capable of transferring this minimal amount of mail without significant delays. An example of a mail system which is extensively used is one consisting of 500 users, and each user sends and receives an average of ten, 2048 byte messages per day. Performance may be an issue for this mail system, which sends or receives an average of ten thousand, 2048 byte messages per day, because not all MHS implementations may be able to transfer this extensive amount of mail without significant delays.

3.4.1 Performance Measurements

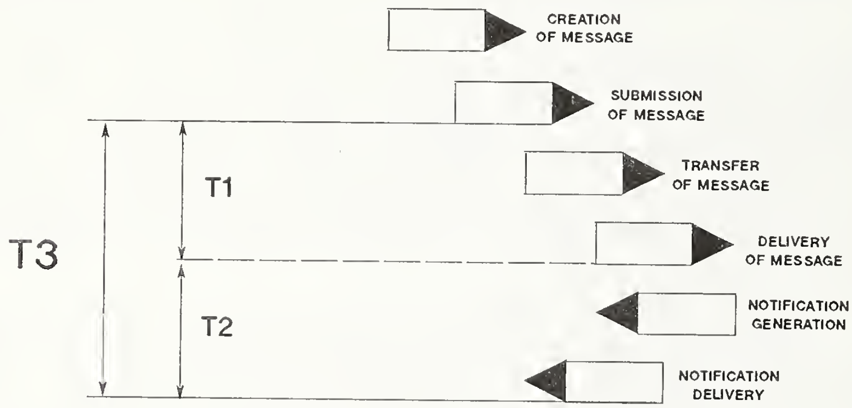
This section contains a variety of experiments which measure the performance of MHS implementations. These experiments provide a representative sampling of MHS performance measurements. The experiments were created with the assistance of users and vendors. MHS performance measurements that may be important to a user were determined, then the experiments necessary to perform these measurements were created. The user should carefully study each experiment to become familiar with the performance measurement obtained. Since there is cost associated with performing experiments, the user should only select the performance measurements which are essential. Although a variety of experiments are described in this section, it is not possible to include every performance measurement that is important to every user. The user may add any performance experiments which are not included in these guidelines, but are important to that user.

The performance experiments defined in these guidelines can be performed by either the user or the vendor. It is more practical for the vendor to perform these experiments for four reasons. The reasons are: (1) The vendor should have the hardware and software required to run their MHS implementation in house. Thus, the user avoids procuring or leasing hardware and software to perform the experiments. (2) The vendor should have the expertise available to install the MHS implementation on the test system, or it may already be installed. Thus, the user avoids the installation procedure for the MHS implementation. (3) The vendor should have access to the source code for the MHS implementation to make any modifications needed to perform the experiments. For example, the vendor may need to add a print statement to the MHS implementation to display the submission and delivery times for a message. Thus, the user avoids the problems associated with acquiring and modifying the source code for the MHS implementation. (4) The vendor should have the expertise to optimize the performance of their implementation, providing the best results possible for the implementation. It is important when comparing performance of different MHS implementations, to compare the best performance of each implementation so the comparison is fair. Thus, the user avoids the time required to learn how to optimize the MHS implementation being measured. Because of these reasons, it is recommended that the user procuring an MHS implementation request the vendor to perform the required experiments.

The person performing these experiments must follow some general guidelines. Each experiment is performed by sending mail from an MHS implementation of a specific vendor to an MHS implementation of the same vendor (e.g., when evaluating Vendor A's MHS implementation, mail is sent from one Vendor A MHS implementation to another Vendor A MHS implementation.) Furthermore, if a message is relayed by an intermediate MHS implementation, that MHS implementation must be from Vendor A also. One exception to this rule is when it is necessary to interoperate with public mail systems. The experiments are performed in this homogeneous environment to ensure that only one variable is measured (i.e., only one MHS implementation.) The user should note, however, that experimental measurements may vary from measurements performed in a production environment. This is because a user may transfer messages from an MHS implementation of one vendor to an MHS implementation of a different vendor.

The results of experiments which relay messages through a public mail system (ADMD) may be less accurate than when all MHS implementations are provided by a single vendor. This is because there are many variables in a public mail system outside the control of the person conducting the experiments. For example, the person conducting the experiments cannot control the number of nodes through which a public mail system relays a message. Also, the person conducting the experiments cannot control the amount of traffic on a public mail system at the time of an experiment.

A basic measurement made in most of the experiments is the message transfer time interval. This interval, referred to as T1, is defined as the message delivery time minus the message submission time. Message delivery time is the time the MTA transfers the contents of a message plus the delivery envelope to a recipient UA. Message submission time is the time the originating UA transfers the contents of a message plus the submission envelope to an MTA. A second basic measurement is the message notification transfer time interval. This interval, referred to as T2, is defined as the notification delivery time minus the notification generation time. Notification delivery time is the time the MTA transfers the notification for a previously submitted message to the UA which originated the message. Notification generation time is the time the notification is generated by the MTA which received the message. Finally, the total confirmed message transfer time interval, referred to as T3, is defined as the notification delivery time minus the message submission time. The user should note that T3 = T1 + T2. These time intervals are pictured in figure 15. Although the experiments in this section only reference the measurement of T1, any experiment may be modified to measure T2 or T3, as required by the user.



T1 = Message Transfer Time Interval

Note 1 – Starting time of T1 corresponds to the Message Submission Time.

Note 2 – Ending time of T1 corresponds to the Message Delivery Time.

T2 = Notification Transfer Time Interval

Note 1 – Starting time of T2 corresponds to the Notification Generation Time.

Note 2 – Ending time of T2 corresponds to Notification Delivery Time.

T3 = Confirmed Message Transfer Time Interval

T3 is defined as the Notification Delivery Time minus the Message Submission Time.

Figure 15. T1, T2, and T3 Model.

To obtain accurate experiment results, measurements must be taken repeatedly and averaged. This is because there are several factors which may vary the result, each time the measurement is performed. These factors include utilization of the CPU and disk by other processes, utilization of the LAN or WAN by other users of the network, retransmission of packets due to transmission errors, and others. The person conducting the experiments must determine the number of repetitions required to stabilize the results of the experiment. This number may vary from experiment to experiment. In order to determine the number of repetitions for a measurement, the measurement should be taken some reasonable number of times (e.g., 20 times) and the results averaged. The measurement should then be taken an additional number of times (e.g., 10 additional times), and averaged over all times (30 total times), to determine whether the additional measurements affected the results. This process should be repeated, increasing the number of additional measurements, until the results are stable to the satisfaction of the person conducting the experiment.

Since these experiments entail one mail system sending messages to another mail system, the two mail systems will most likely be running on separate hardware. The person performing the experiments must synchronize the clocks on the hardware, in order for the experiment results to be meaningful. It is recommended that the clocks be synchronized to within 1 second. It is beyond the scope of these guidelines to recommend procedures for synchronizing clocks.

Several general inputs must be provided by the user in order to perform the experiments. Inputs 1-3 are constant for all experiments; inputs 4-5 may vary from experiment to experiment. A description of these inputs follows:

(1) The user's MHS configuration. As described in section 3.2, the user should determine the MHS configuration to be used. The performance of candidate MHS implementations should be measured on a configuration which matches the user's MHS configuration. If the user's configuration contains more MTAs than referenced in an experiment, the user must specify which MTAs are to be used in that experiment.

(2) The model(s) of the computer(s) to be used in the experiments. Many computer systems are available in a variety of models, where the low end models have a slower CPU and disk access time than the high end models. If the user already has the hardware for the MHS implementation, then the model number should be specified to the person conducting the experiment. If the user is procuring hardware with the MHS implementation, then the user may provide the vendor with input on the user's performance requirements. The vendor may then recommend which models of computers are likely to meet the performance measurements required by the user. The user should be careful when specifying performance requirements to the vendor, not to specify a level of performance which greatly exceeds the user's requirements. This may increase hardware costs unnecessarily. The person conducting the experiments should run the experiments on the model of hardware that will be used by the user.

(3) The network type. The user should provide the vendor with the type of Local Area Network or Wide Area Network, and if relevant, the network speed and network configuration (i.e., public vs. private WAN). If possible, the person conducting the experiments should perform the experiments on a network of the same type, speed, and configuration as the user's network. Otherwise, the user should note that the results of

the experiments may differ from the user's actual performance. For example, if the person conducting the experiments performs them on a public WAN while the user's configuration contains a private WAN, which may carry significantly less network traffic, the user's actual performance may be better than the performance measured in the experiments.

(4) The experimental environment. Two types of user environments are defined for performing the experiments. The first environment, referred to as the ideal environment, is one in which there are no application processes competing with the MHS implementation for the resources of the CPU, disk, and network. This environment is useful for measuring the maximum performance of an MHS implementation. The second environment, referred to as the real environment, is one in which there are application processes competing for the resources of the CPU, disk, and network with the MHS implementation. The percent of CPU, disk, and network resources utilized by the competing application processes should reflect the utilization of these resources by application processes normally run on the user's system. This environment is useful for measuring the typical performance of an MHS implementation. On a single user system the only resource for which there may be competition is the network, since multiple processes cannot run in this environment. The experiments may be performed, as requested by the user, in either an ideal or real environment, or both. It is beyond the scope of these guidelines to recommend procedures for performing the experiment in a real environment.

(5) The message size used in the experiment. This message size refers to the length of the message body; it does not include content header information. Only one message size is input to an experiment, and it should represent the average size of a message to be transferred by the user's MHS implementation. If the user has an existing mail system, the average message size can be calculated by dividing the number of bytes sent in messages in a certain time period, such as a day, by the number of messages sent in that time period. Otherwise, 2000 bytes (i.e., one screen of text) may be considered as an average message size by the user.

The performance experiments in this section are presented with the following format. Each experiment has five parts: introduction, methodology, user input, example, and summary of results. The introduction contains the purpose of the experiment. It describes what the experiment measures and its applicability to a user. The methodology section contains a diagram followed by instructions for the person conducting the experiment. The diagram shows the message transfer path(s) between the users, UAs, and MTAs involved in the experiment, and the instructions specify how the experiment is to be performed. The user input section lists the information the user must provide to the person conducting the experiment. Certain user inputs remain constant for each experiment, e.g., the MHS configuration, the model(s) of the computer(s) to be used, and the network type. Thus, these requirements are not repeated in the user input section of each experiment. There is an example for each experiment. Both user input values and experiment results are presented in the example. The input values and results in the examples are simulated. The final section, summary of results, discusses the results presented in the example section. Formulas for calculating results as well as an interpretation of the results are provided.

Seven performance experiments are presented in this section. The first five experiments measure T1. Experiment #1 provides the base measurement; T1 is calculated for one MHS message. Experiments #2, #3, #4 and #5 each introduce one additional factor that may affect T1. Experiment #2 measures the impact of estimated mail system usage. Experiment #3 measures the impact of an MTA relay. Experiment #4 measures the impact of an additional recipient for the MHS message, and Experiment #5 measures the impact of varying the message size. Comparing the measurements of T1 in Experiments #2, #3, #4, and #5 to the measurement of T1 in Experiment #1, the user can determine the effect of the factor measured in the experiment. As reflected in the examples for the message transfer experiments, the user must input the same values (e.g., experiment environment) for each experiment for the comparison to be valid.

Experiments #6 and #7 measure mail system capacity and system utilization respectively. Experiment #6 determines the amount of time required by an MHS implementation to transfer a specific number of messages. Experiment #7 determines the amount of CPU and I/O utilized by an MHS implementation.

3.4.1.1 Experiment 1

(1) Introduction

Experiment #1 measures T1 for a single MHS message. This measurement represents a baseline message transfer time interval; T1 measurements in other experiments can be compared to the T1 measurement in this experiment to determine the impact of other factors on T1. Since this experiment measures no other factors, the measurement of T1 in this experiment also represents the optimum transfer time interval for a message.

(2) Methodology

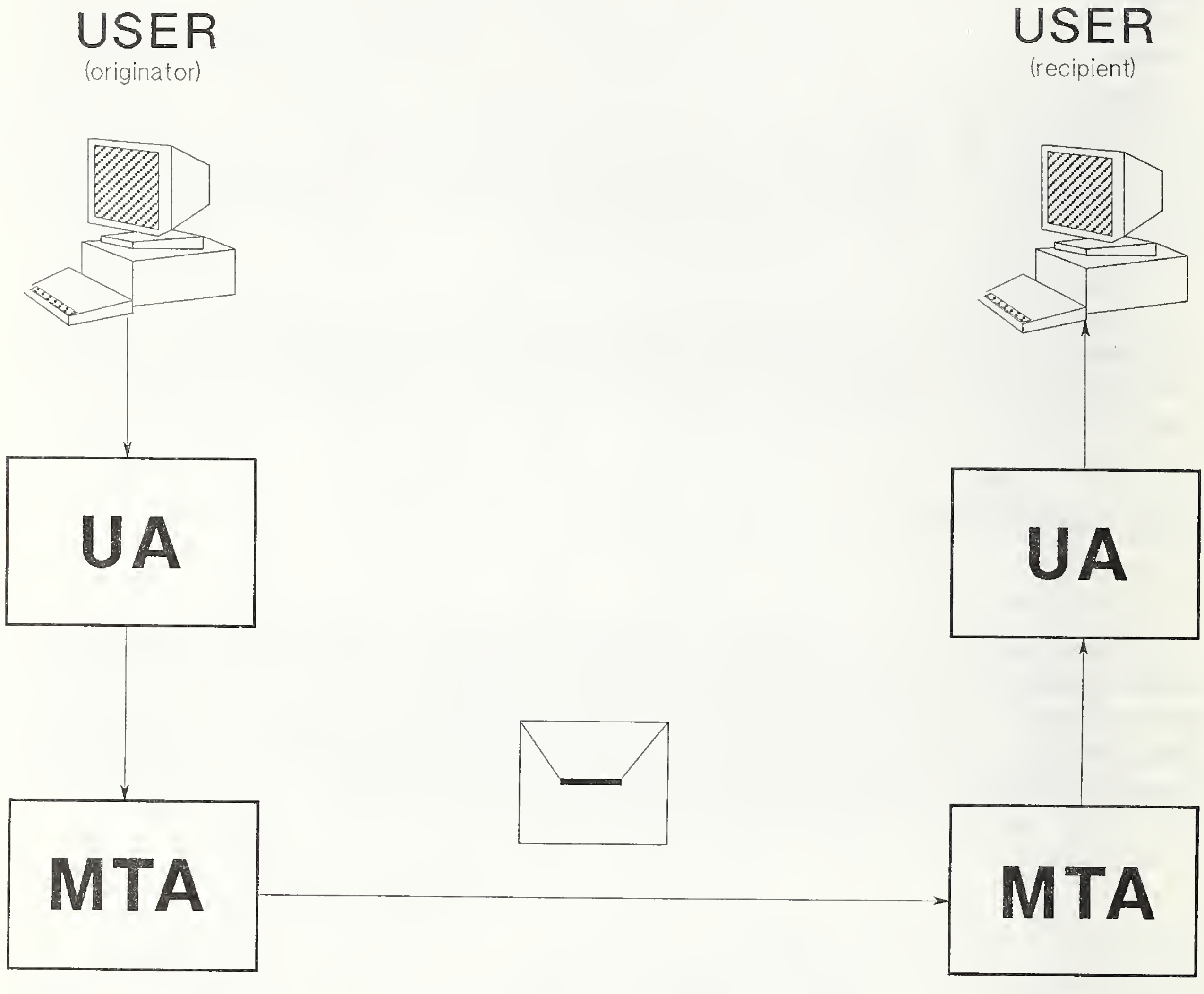


Figure 16. Experiment 1 Message Transfer Path.

This experiment involves one message originator and one message recipient, each residing on different MTAs. MHS messages are transferred serially; the originator should not submit a message until the previous message has been delivered. This may be accomplished by submitting messages according to a predefined time interval. The interval must be long enough to allow the previous message to be delivered and the MTA connection to close. All messages submitted must be of uniform size. For each message transferred, the person conducting the experiment should log the submission and delivery times.

(3) User Input

The user must provide the person conducting the experiment with the following information:

- the experiment environment (i.e., ideal or real)

- the size of the messages to be transferred

(4) Example

For this example, a government agency is interested in procuring an MHS implementation. The agency has three locations: (1) Overton, TX, (2) Ellicott City, MD, and (3) Washington, DC The Overton branch receives approximately 500 letters per day requesting information. These requests are sorted, then depending on the request, sent to the Ellicott City location, the Washington, DC location, or to both locations. The Ellicott City and Washington, DC locations mail the appropriate information to the people who requested it, then return billing information to the Overton branch.

The information requests and billing information are currently transferred between Overton and Ellicott City and Overton and Washington, DC via Simple Mail Transfer Protocol (SMTP) implementations. These SMTP implementations are to be replaced with MHS implementations. The government agency, in cooperation with a vendor of choice, has determined that the Overton location will use two MTAs to support its 250 users. The Ellicott City location will use one MTA to support its 75 users, and the Washington, DC location will use one MTA to support its 50 users. One of the two Overton MTAs, and the Ellicott City and Washington, DC MTAs will be connected by a WAN. The two Overton MTAs will be connected by a LAN. See figure 17 for this MHS configuration. The agency expects heavy mail system usage between Overton and Ellicott City and Overton and Washington, DC, and thus, wants to test performance across the WAN.

For Experiment #1, the agency requests that the experiment be performed in an ideal environment using a message size of 2000 bytes (i.e., one screen of text). The person conducting the experiment provides the agency with the following results.

Table 5. Example Results For Experiment #1

Message Number	Submission Time	Delivery Time	T1
1	12:00.00 PM	12:03.52 PM	3 min 52 sec
2	12:10.00 PM	12.13:59 PM	3 min 59 sec
3	12:20.00 PM	12:24.04 PM	4 min 04 sec
4	12:30.00 PM	12:34.16 PM	4 min 16 sec
5	12:40.00 PM	12:43.44 PM	3 min 44 sec
6	12:50.00 PM	01.54.09 PM	4 min 09 sec
7	01:00.00 PM	01:04.01 PM	4 min 01 sec
8	01:10.00 PM	01:13.47 PM	3 min 47 sec
9	01:20.00 PM	01.24.00 PM	4 min 00 sec
10	01:30.00 PM	01.33.51 PM	3 min 51 sec

The minimum T1 measurement is 3 minutes 44 seconds.
The maximum T1 measurement is 4 minutes 16 seconds.
The average T1 measurement is 3 minutes 58 seconds.

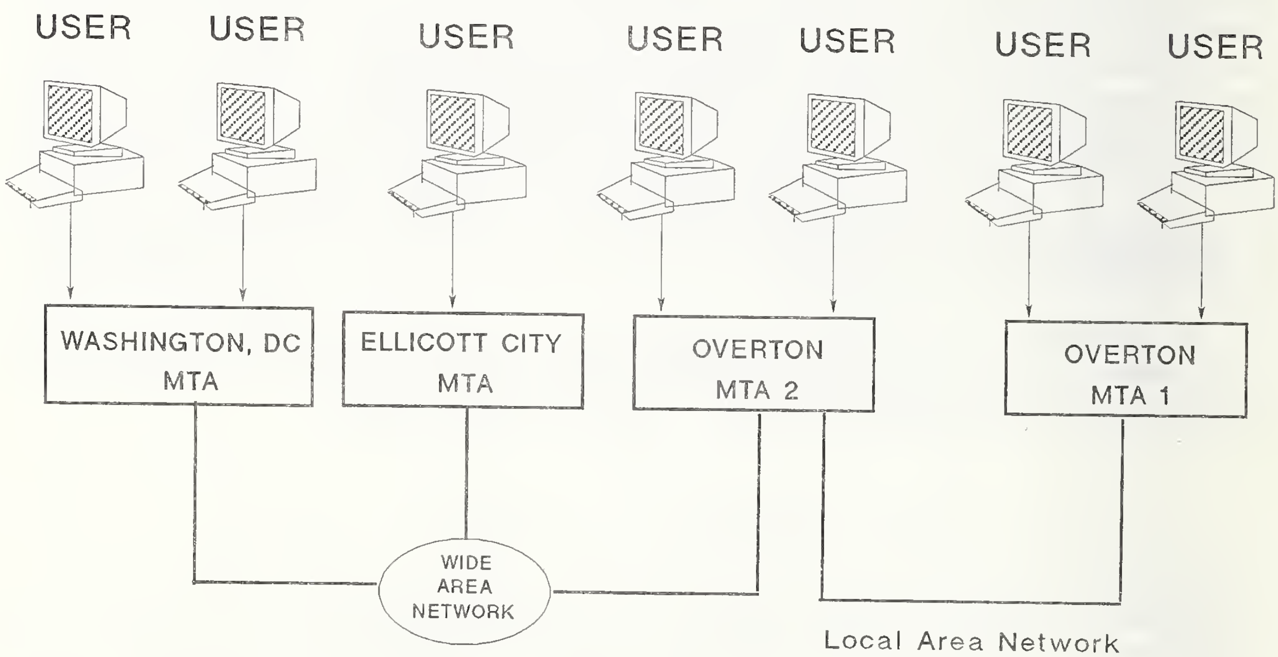


Figure 17. MHS Configuration used in Experiment Examples.

(5) Summary of Results

The person conducting the experiment determined that the experiment results were stable with the transfer of 10 messages. For each message transferred the following information is provided: the message number, the submission time, the delivery time, and T1. T1, which is the only calculated value, is the delivery time minus the submission time.

The minimum, maximum, and average T1 measurements are also provided. The minimum and maximum T1 measurements are the fastest and slowest message transfers respectively. The average T1 measurement is the sum of all T1 measurements divided by the number of messages transferred. See table 6 for the formula. The average T1 measurement for this experiment represents the optimum transfer time interval for the message size requested.

The results recorded in this experiment do not represent actual data collected. They are listed for example purposes only. While the results simulate data obtained from one vendor's MHS implementation, the experiment must be repeated for each vendor's MHS implementation which is of interest to the user.

Table 6. Equation for Average T1

$$AT1 = \left(\sum_{i=1}^{m} (T1_i) \right) / m$$

$AT1 = Average\ T1$

$T1_i = T1$ for $Message\ i$

$m = Number\ of\ Messages\ Transferred$

3.4.1.2 Experiment 2

(1) Introduction

In a typical MHS mail system, messages are submitted randomly by users. A user can create a table of sample message submission times which estimate usage of the user's mail system. This experiment measures T1 based on estimated mail system usage.

(2) Methodology

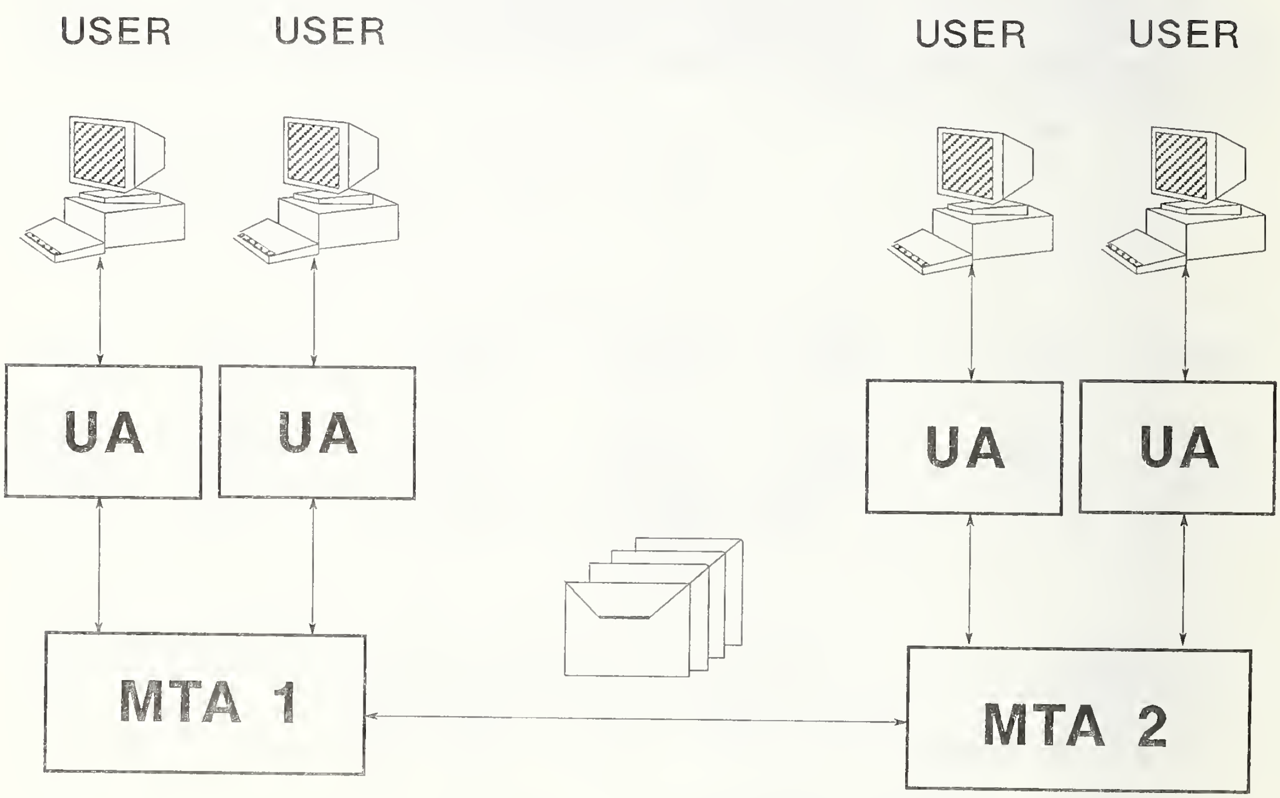


Figure 18. Experiment 2 Message Transfer Path.

This experiment involves two MTAs, each supporting multiple UAs and users. Messages are submitted to MTA 1 based on information in a user defined message submission time table. Similarly, messages are submitted to MTA 2 based on information in a second user defined message submission time table. Each table should include first, last, and various intermediate message submission time entries along with a number of messages to be transferred for each entry. If multiple messages are to be transferred for a single entry, the messages must be submitted simultaneously. All messages must address only one recipient and be of uniform size. For each message transferred, the person conducting the experiment should log the submission and delivery times.

(3) User Input

The user must provide the person conducting the experiment with the following information:

- the experiment environment (i.e., ideal or real)
- the size of the messages to be transferred
- two message submission time tables

(4) Example

The government agency described in Experiment #1 had previously analyzed the number of information requests and billing notices transferred between the Overton and Ellicott City locations. Based on this information the agency has requested this experiment to be performed over the course of 1 hour using a message size of 2000 bytes. Messages are to be submitted according to the following message submission time tables.

Table 7. Example Message Submission Time Table for Overton MTA

Time	Number Of Messages
12:00	2
12:02	1
12:05	1
12:15	3
12:20	2
12:30	2
12:33	1
12:37	1
12:45	5
12:59	3

Table 8. Example Message Submission Time Table for Ellicott City MTA

Time	Number Of Messages
12:00	2
12:04	1
12:12	1
12:15	2
12:22	2
12:28	1
12:35	1
12:43	4
12:50	2
12:53	1
12:56	1
12:59	3

The person conducting the experiment provides the agency with the following results.

Table 9. Example Results for Overton MTA

Message Number	Submission Time	Delivery Time	T1
1	12:00.00 PM	12:03.54 PM	3 min 54 sec
2	12:00.00 PM	12.05.18 PM	5 min 18 sec
3	12:02.00 PM	12:05.57 PM	3 min 57 sec
4	12:05.00 PM	12:08.28 PM	4 min 28 sec
5	12:15.00 PM	12:18.58 PM	3 min 58 sec
6	12:15.00 PM	12.20.15 PM	5 min 15 sec
7	12:15.00 PM	12:21.29 PM	6 min 29 sec
8	12:20.00 PM	12:23.52 PM	3 min 52 sec
9	12:20.00 PM	12.25.34 PM	5 min 34 sec
10	12:30.00 PM	12.34.01 PM	4 min 01 sec
11	12:30.00 PM	12:35.30 PM	5 min 30 sec
12	12:33.00 PM	12.37.15 PM	4 min 15 sec
13	12:37.00 PM	12:41.00 PM	4 min 00 sec
14	12:45.00 PM	12:48.56 PM	3 min 56 sec
15	12:45.00 PM	12:50.11 PM	5 min 11 sec
16	12:45.00 PM	12.51.20 PM	6 min 20 sec
17	12:45.00 PM	12:52.37 PM	7 min 37 sec
18	12:45.00 PM	12:53.51 PM	8 min 51 sec
19	12:59.00 PM	01.02.51 PM	3 min 51 sec
20	12:59.00 PM	01.04.37 PM	5 min 37 sec
21	12:59.00 PM	01.05.55 PM	6 min 55 sec

Table 10. Example Results for Ellicott City MTA

Message Number	Submission Time	Delivery Time	T1
1	12:00.00 PM	12:04.04 PM	4 min 04 sec
2	12:00.00 PM	12.05.22 PM	5 min 22 sec
3	12:04.00 PM	12:08.00 PM	4 min 00 sec
4	12:12.00 PM	12:16.17 PM	4 min 17 sec
5	12:15.00 PM	12:18.56 PM	3 min 56 sec
6	12:15.00 PM	12.20.16 PM	5 min 16 sec
7	12:22.00 PM	12.26.01 PM	4 min 01 sec
8	12:22.00 PM	12:27.20 PM	5 min 20 sec
9	12:28.00 PM	12.32.15 PM	4 min 15 sec
10	12:35.00 PM	12:39.08 PM	4 min 08 sec
11	12:43.00 PM	12:47.03 PM	4 min 03 sec
12	12:43.00 PM	12:48.18 PM	5 min 18 sec
13	12:43.00 PM	12.49.25 PM	6 min 25 sec
14	12:43.00 PM	12:50.42 PM	7 min 42 sec
15	12:50.00 PM	12:54.04 PM	4 min 04 sec
16	12:50.00 PM	12.55.32 PM	5 min 32 sec
17	12:53.00 PM	12.57.07 PM	4 min 07 sec
18	12:56.00 PM	01.00.05 PM	4 min 05 sec
19	12:59.00 PM	01.03.15 PM	4 min 15 sec
20	12:59.00 PM	01.04.39 PM	5 min 39 sec
21	12:59.00 PM	01.05.55 PM	6 min 55 sec

The minimum T1 measurement is 3 minutes 51 seconds.
The maximum T1 measurement is 8 minutes 51 seconds.
The average T1 measurement is 5 minutes 02 seconds.

The difference between the average T1 measurement of Experiment #1 (3 min 58 s) and the average T1 measurement of Experiment #2 is 1 minute 04 seconds.

(5) Summary of Results

For each message transferred the following information is provided: the message number, the submission time, the delivery time, and T1. T1, which is the only calculated value, is the delivery time minus the submission time.

The minimum, maximum, and average T1 measurements are also provided. The minimum and maximum T1 measurements are the fastest and slowest message transfers respectively. The average T1 measurement is the sum of all T1 measurements divided by the number of messages transferred. See table 6 for the formula. The final calculated value is the difference between the average T1 measurement for this experiment and the average T1 measurement for Experiment #1.

The results of this experiment reflect that transferring multiple MHS messages across one instance of an MTA connection will require less time than transferring the messages across multiple instances of the MTA connection. The explanation is as follows. When MHS messages are transferred, a connection is established between the two MTAs involved in the transfer. Once established many messages can be transferred across the connection. Thus, until the MTA connection is closed, every message transferred after the first message will not include MTA connection establishment time.

This experiment is different from Experiment #1 in that MHS messages are submitted based on estimated mail system usage. The average T1 measurement in this experiment can be compared to that in Experiment #1 to determine the effect of this difference. The user can expect that additional time may be required to transfer a message. This is because the MHS implementation must process messages submitted simultaneously by different users. The amount of additional time will be affected by the frequency of message transfers.

The results recorded in this experiment do not represent actual data collected. They are listed for example purposes only. While the results simulate data obtained from one vendor's MHS implementation, the experiment must be repeated for each vendor's MHS implementation which is of interest to the user.

3.4.1.3 Experiment 3

(1) Introduction

This experiment measures T1 for a single MHS message relayed by an intermediate MTA. Comparing the results of this experiment with those of Experiment #1, a user can determine the amount of time that an intermediate MTA relay adds to the message transfer.

(2) Methodology

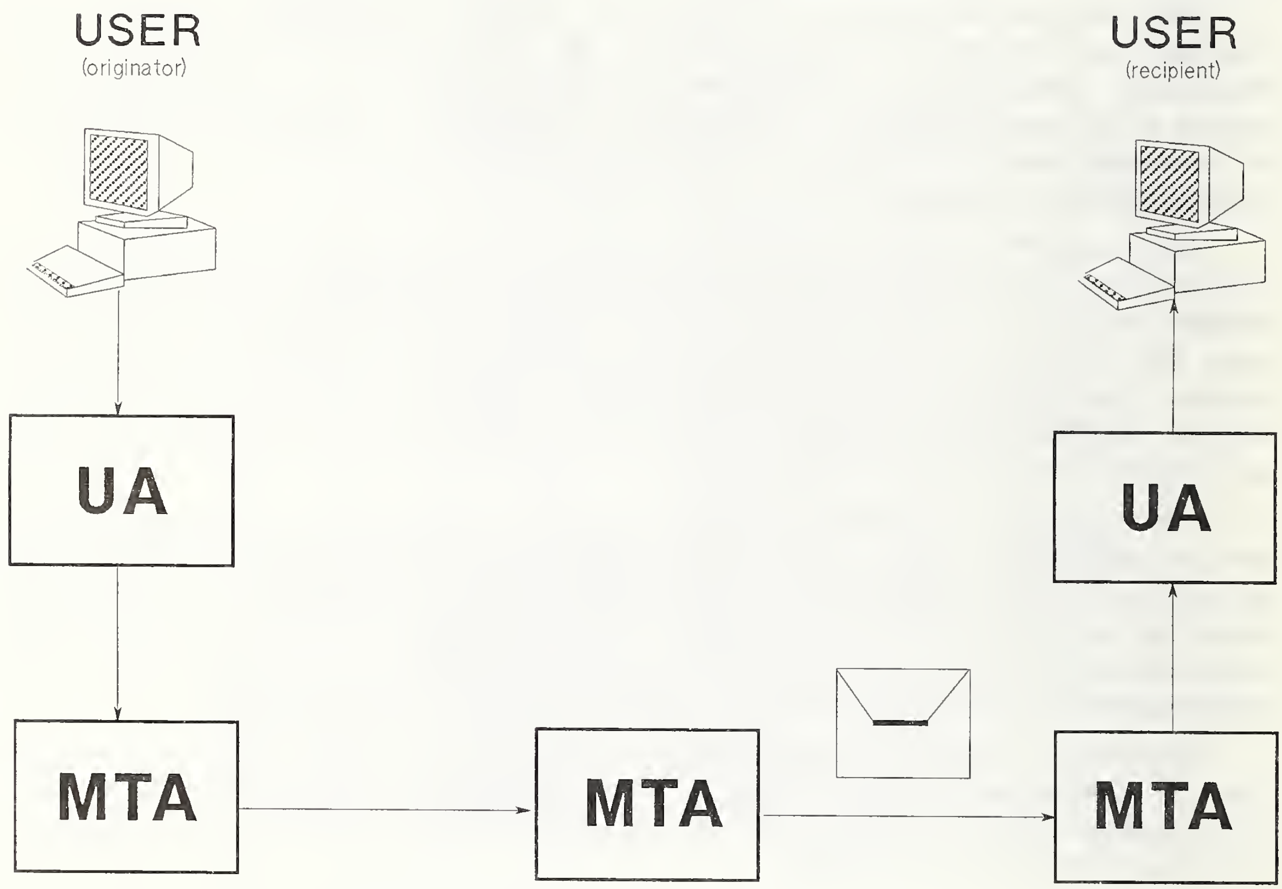


Figure 19. Experiment 3 Message Transfer Path.

This experiment involves one message originator and one message recipient, each residing on different MTAs. MHS messages are relayed from the originator's MTA to the recipient's MTA by an intermediate MTA. Messages are transferred serially; the originator should not submit a message until the previous message has been delivered. This may be accomplished by submitting messages according to a predefined time interval. The interval must be long enough to allow the previous message to be delivered and the MTA connection to close. All messages submitted must be of uniform size. For each message transferred, the person conducting the experiment should log the submission and delivery times.

(3) User Input

The user must provide the person conducting the experiment with the following information:

- the experiment environment (i.e., ideal or real)

- the size of the messages to be transferred

- the network type between the originating and intermediate MTAs and the intermediate and recipient MTAs (i.e., LAN or WAN)

(4) Example

The government agency described in Experiment #1 has two MTAs at the Overton location. It is of interest to the agency to determine the amount of additional time needed to relay a message from the Overton MTA not connected to the WAN, through the Overton MTA connected to the WAN, to the Ellicott City location. For this experiment, the agency requests that the experiment be performed in an ideal environment using a message size of 2000 bytes. A LAN connects the originating and intermediate MTAs, and a WAN connects the intermediate and recipient MTAs. The person conducting the experiment provides the agency with the following results.

Table 11. Example Results For Experiment #3

Message Number	Submission Time	Delivery Time	T1
1	12:00.00 PM	12:05.59 PM	5 min 59 sec
2	12:10.00 PM	12.16:14 PM	6 min 14 sec
3	12:20.00 PM	12:26.14 PM	6 min 14 sec
4	12:30.00 PM	12:36.26 PM	6 min 26 sec
5	12:40.00 PM	12:46.20 PM	6 min 20 sec
6	12:50.00 PM	01.56.09 PM	6 min 09 sec
7	01:00.00 PM	01:06.01 PM	6 min 01 sec
8	01:10.00 PM	01:16.17 PM	6 min 17 sec
9	01:20.00 PM	01.26.00 PM	6 min 00 sec
10	01:30.00 PM	01.35.59 PM	5 min 59 sec

The minimum T1 measurement is 5 minutes 59 seconds.
The maximum T1 measurement is 6 minutes 26 seconds.
The average T1 measurement is 6 minutes 10 seconds.

The difference between the average T1 measurement of Experiment #1 (3 min 58 s) and the average T1 measurement of Experiment #3 is 2 minutes 12 seconds.

(5) Summary of Results

The person conducting the experiment determined that the experiment results were stable with the transfer of 10 messages. For each message transferred the following information is provided: the message number, the submission time, the delivery time, and T1. T1, which is the only calculated value, is the delivery time minus the submission time.

The minimum, maximum, and average T1 measurements are also provided. The minimum and maximum T1 measurements are the fastest and slowest message transfers respectively. The average T1 measurement is the sum of all T1 measurements divided by the number of messages transferred. See table 6 for the formula. The final calculated value is the difference between the average T1 measurement for this experiment and the average T1 measurement for Experiment #1.

This experiment is identical to Experiment #1 except that an intermediate MTA is used to relay messages between the originator and the recipient MTAs. The average T1 measurement in this experiment can be compared to that in Experiment #1 to determine the effect of the MTA relay. The user should expect the relay to add a certain amount of time to the message transfer. The exact amount will be affected by the type and speed (if relevant) of the network. Once the additional amount of time for one MTA relay is calculated, the user can generalize that each additional relay will add approximately the same amount of time to the message transfer.

The results recorded in this experiment do not represent actual data collected. They are listed for example purposes only. While the results simulate data obtained from one vendor's MHS implementation, the experiment must be repeated for each vendor's MHS implementation which is of interest to the user.

3.4.1.4 Experiment 4

(1) Introduction

This experiment measures T1 for a single MHS message addressed to two recipients on two different MTAs. Comparing the results of this experiment with those of Experiment #1, a user can determine the additional time required to transfer a message to an additional recipient residing on a separate MTA.

(2) Methodology

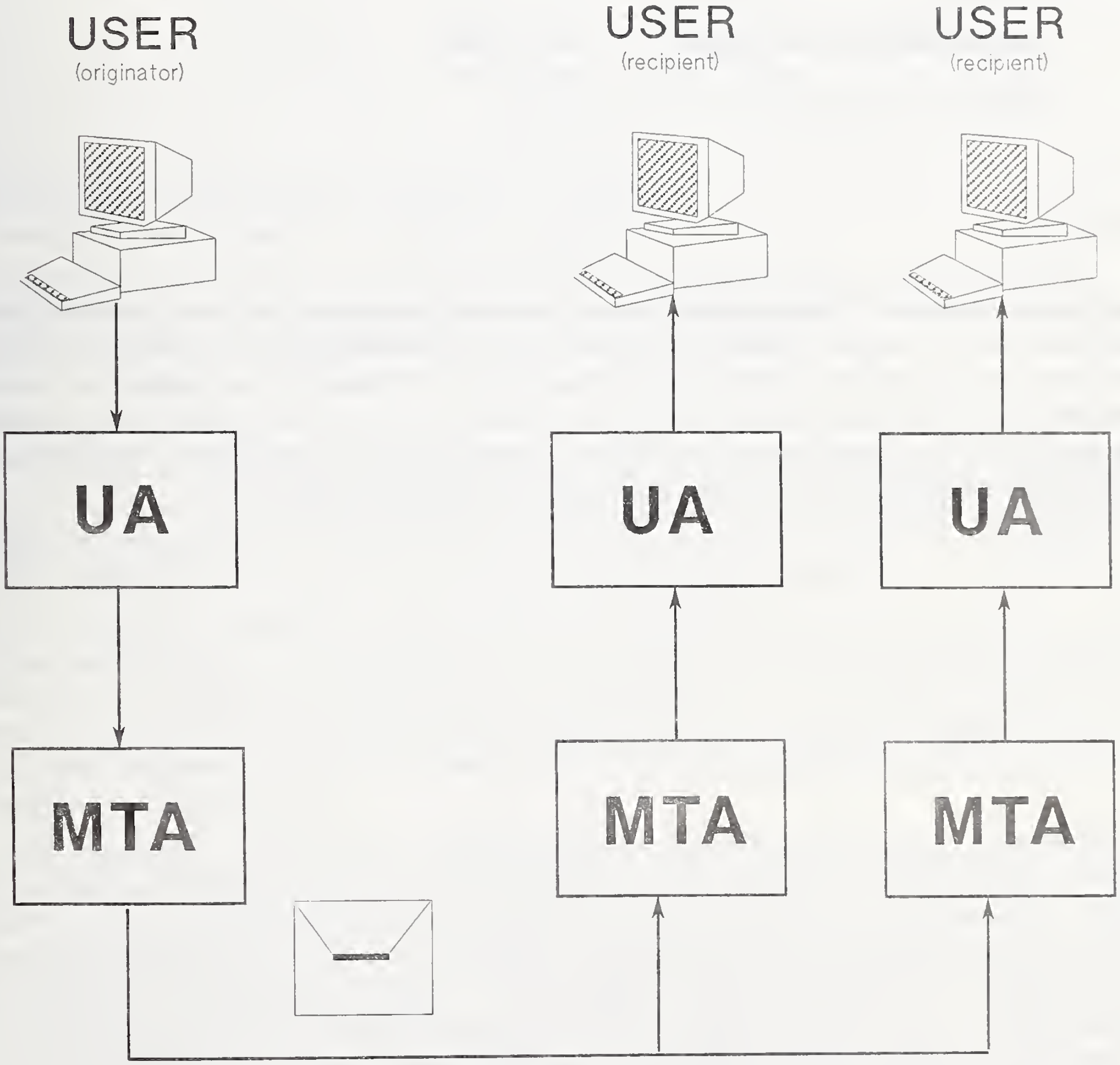


Figure 20. Experiment 4 Message Transfer Path.

This experiment involves one message originator and two message recipients, each residing on a different MTA. MHS messages are transferred serially; the originator should not submit a message until the previous message has been delivered. This may be accomplished by submitting messages according to a predefined time interval. The interval must be long enough to allow the previous message to be delivered to both recipients and all MTA connections to close. All messages submitted must be addressed to both recipients and be of uniform size. For each message transferred, the person conducting the experiment should log the submission and delivery times.

(3) User Input

The user must provide the person conducting the experiment with the following information:

- the experiment environment (i.e., ideal or real)

- the size of the messages to be transferred

(4) Example

The government agency described in Experiment #1 receives requests for information at the Overton location. Requests for multiple pieces of information are often received, and the requests must be mailed to both the Ellicott City and Washington, DC locations. Thus, it is of interest to the agency to determine the time required to transfer a message to two recipients located on two different MTAs. For this experiment, the agency requests that the experiment be performed in an ideal environment using a message size of 2000 bytes. The person conducting the experiment provides the agency with the following results.

Table 12. Example Results For Experiment #4

Message Number	Submission Time	MTA 1 Delivery Time	MTA 2 Delivery Time	T1
1	10:00.00 AM	10:04.03 AM	10:06.13 AM	6 min 13 sec
2	10:10.00 AM	10.14.14 AM	10.16.29 AM	6 min 29 sec
3	10:20.00 AM	10:24.00 AM	10:26.05 AM	6 min 05 sec
4	10:30.00 AM	10:34.17 AM	10:36.33 AM	6 min 33 sec
5	10:40.00 AM	10:44.05 AM	10:46.18 AM	6 min 18 sec
6	10:50.00 AM	10.54.12 AM	10.56.38 AM	6 min 38 sec
7	11:00.00 AM	11:03.59 AM	11:06.11 AM	6 min 11 sec
8	11:10.00 AM	11:14.10 AM	11:16.20 AM	6 min 20 sec
9	11:20.00 AM	11.24.06 AM	11.26.29 AM	6 min 29 sec
10	11:30.00 AM	11.34.02 AM	11.36.27 AM	6 min 27 sec

The minimum T1 measurement is 6 minutes 05 seconds.
The maximum T1 measurement is 6 minutes 38 seconds.
The average T1 measurement is 6 minutes 22 seconds.

The difference between the average T1 measurement of Experiment #1 (3 min 58 s) and the average T1 measurement of Experiment #4 is 2 minutes 24 seconds.

(5) Summary of Results

The person conducting the experiment determined that the experiment results were stable with the transfer of 10 messages. For each message transferred the following information is provided: the message number, the submission time, the delivery time, and T1. T1, which is the only calculated value, is the delivery time minus the submission time.

The minimum, maximum, and average T1 measurements are also provided. The minimum and maximum T1 measurements are the fastest and slowest message transfers respectively. The average T1 measurement is the sum of all T1 measurements divided by the number of messages transferred. See table 6 for the formula. The final calculated value is the difference between the average T1 measurement for this experiment and the average T1 measurement for Experiment #1.

This experiment is identical to Experiment #1 except that each message transferred is addressed to two recipients residing on two different MTAs. The user should note that this experiment does not determine the time required to send the message to each recipient. The average measurement of T1 for this experiment represents the time required for both recipients to receive the message submitted. This average T1 measurement can be compared to that of Experiment #1 to determine the effect of the additional recipient. The user should expect the extra recipient to add a certain amount of time to the message transfer. The exact amount will be affected by whether the MHS implementation supports multiple simultaneous MTA connections. If the MHS implementation can transfer the message to both recipients concurrently, the transfer time may be faster than if the transfer of the message to one recipient must be completed before the transfer to the second recipient can begin. If desired, the user may request this experiment to be performed multiple times, each time adding another recipient that resides on a different MTA. The user may then derive a formula for calculating the additional time required to transfer a message with an additional recipient.

The results recorded in this experiment do not represent actual data collected. They are listed for example purposes only. While the results simulate data obtained from one vendor's MHS implementation, the experiment must be repeated for each vendor's MHS implementation which is of interest to the user.

3.4.1.5 Experiment 5

(1) Introduction

Experiment #5 measures T1 for a single MHS message. The message size used in this experiment must be different from the message size used in Experiment #1. Comparing the results of this experiment to Experiment #1, the user can determine the impact of varying the message size on the time required to transfer the message.

(2) Methodology

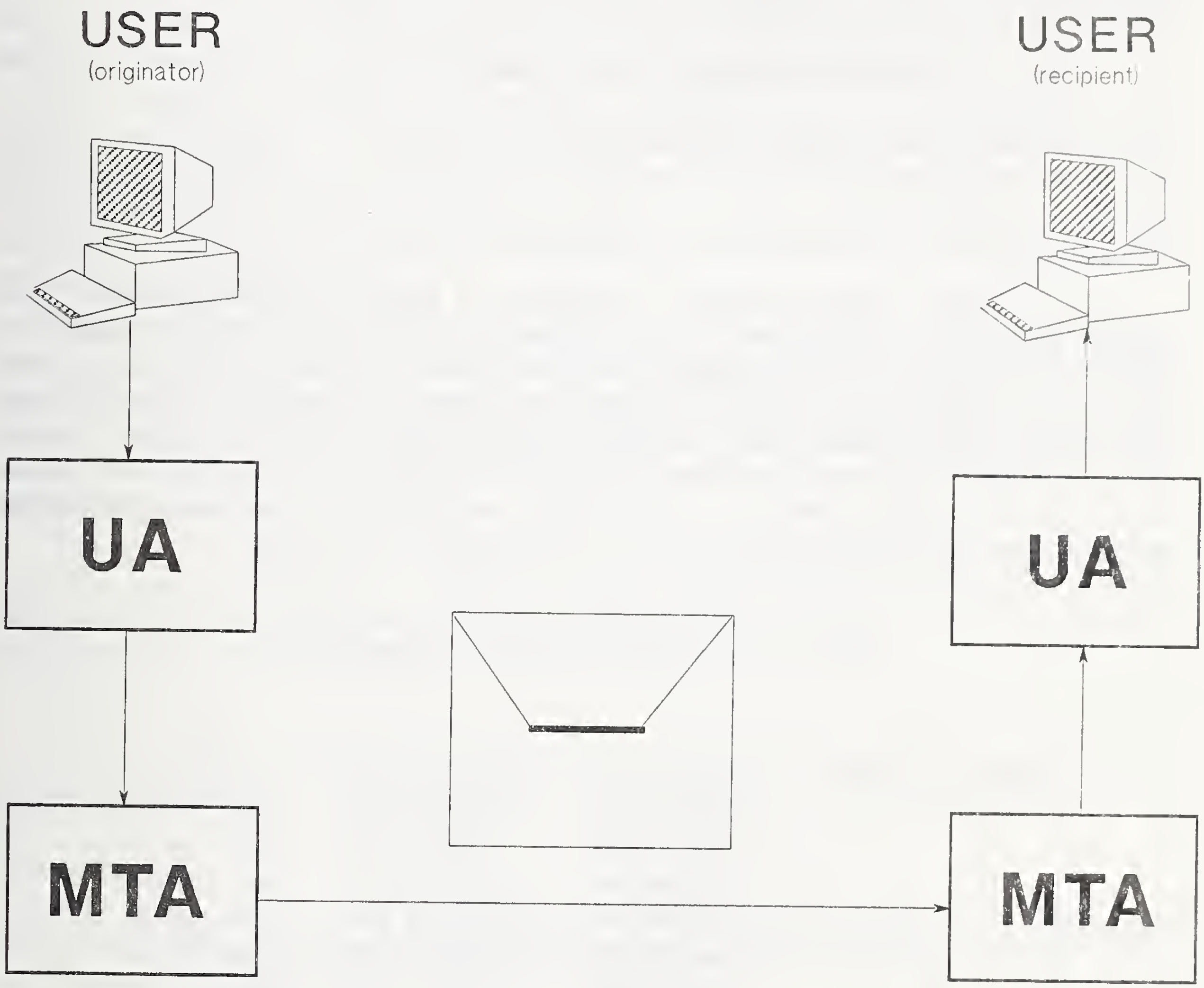


Figure 21. Experiment 5 Message Transfer Path.

This experiment involves one message originator and one message recipient, each residing on different MTAs. MHS messages are transferred serially; the originator should not submit a message until the previous message has been delivered. This may be accomplished by submitting messages according to a predefined time interval. The interval must be long enough to allow the previous message to be delivered and the MTA connection to close. All messages submitted must be of uniform size. For each message transferred, the person conducting the experiment should log the submission and delivery times.

(3) User Input

The user must provide the person conducting the experiment with the following information:

- the experiment environment (i.e., ideal or real)

- the size of the message to be transferred

(4) Example

The government agency described in Experiment #1 receives information requests at the Overton location. Although most of the requests are for multiple pieces of information, requests for a single piece of information are often received. Thus, it is of interest to the agency to determine the time required to transfer a message representing this single information request, which is smaller than the typical message transferred by the agency. For this experiment, the agency requests that the experiment be performed in an ideal environment using a message size of 500 bytes. The person conducting the experiment provides the agency with the following results.

Table 13. Example Results For Experiment #5

Message Number	Submission Time	Delivery Time	T1
1	10:00.00 AM	10:04.04 AM	4 min 04 sec
2	10:10.00 AM	10.13.54 AM	3 min 54 sec
3	10:20.00 AM	10:23.48 AM	3 min 48 sec
4	10:30.00 AM	10:33.50 AM	3 min 50 sec
5	10:40.00 AM	10:43.24 AM	3 min 24 sec
6	10:50.00 AM	10.53.51 AM	3 min 51 sec
7	11:00.00 AM	11:04.12 AM	4 min 12 sec
8	11:10.00 AM	11:13.37 AM	3 min 37 sec
9	11:20.00 AM	11.23.50 AM	3 min 50 sec
10	11:30.00 AM	11.33.39 AM	3 min 39 sec

The minimum T1 measurement is 3 minutes 24 seconds.
The maximum T1 measurement is 4 minutes 12 seconds.
The average T1 measurement is 3 minutes 49 seconds.

The difference between the average T1 measurement of Experiment #1 (3 min 58 s) and the average T1 measurement of Experiment #5 is 9 seconds.

(5) Summary of Results

The person conducting the experiment determined that the experiment results were stable with the transfer of 10 messages. For each message transferred the following information is provided: the message number, the submission time, the delivery time, and T1. T1, which is the only calculated value, is the delivery time minus the submission time.

The minimum, maximum, and average T1 measurements are also provided. The minimum and maximum T1 measurements are the fastest and slowest message transfers respectively. The average T1 measurement is the sum of all T1 measurements divided by the number of messages transferred. See table 6 for the formula. The final calculated value is the difference between the average T1 measurement for this experiment and the average T1 measurement for Experiment #1.

This experiment is identical to Experiment #1 except that the size of the message transferred is smaller. The user should expect the amount of time required to transfer a message of this size to be less than that of Experiment #1. Likewise, if the user requested a message size greater than that of Experiment #1, the user should expect the amount of time required to transfer the message to be greater. The exact amount will be determined by the size of the message.

The results recorded in this experiment do not represent actual data collected. They are listed for example purposes only. While the results simulate data obtained from one vendor's MHS implementation, the experiment must be repeated for each vendor's MHS implementation which is of interest to the user.

3.4.1.6 Experiment 6

(1) Introduction

This experiment measures the message transfer capacity of an MHS implementation. Mail system capacity is useful for determining the amount of time required to transfer a maximum number of messages.

(2) Methodology

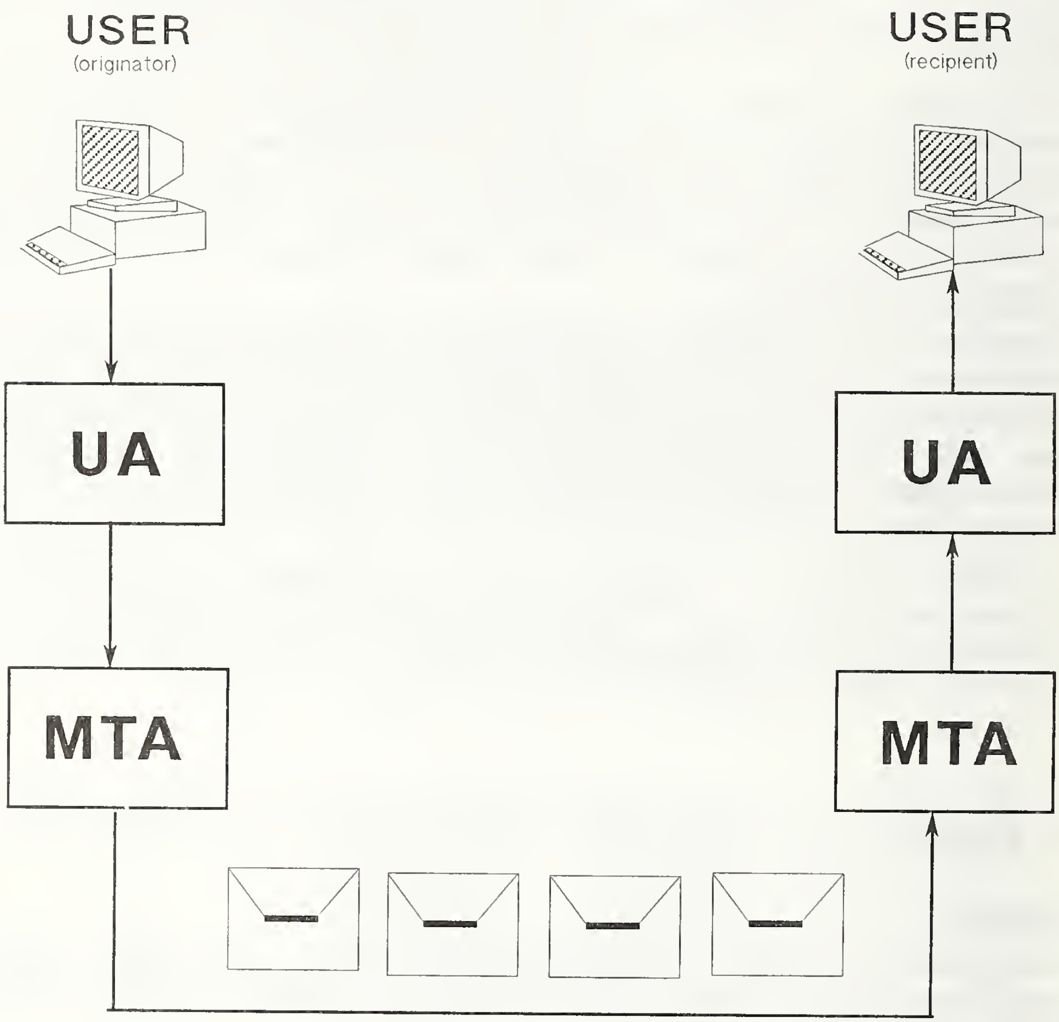


Figure 22. Experiment 6 Message Transfer Path.

For this experiment a user selects a length of time during which heavy usage of the mail system is expected, then estimates the maximum number of MHS messages that would be submitted to the mail system during that time period. The user inputs this number of messages to the experiment. In addition, the user specifies a submission time interval for the messages. This interval should be a fraction (e.g., 1/2) of the average submission time interval for one message, which is calculated by dividing the time period selected by the user by the number of messages. For example, if the user selects a time period of 1 hour, and estimates that the submission of 30 messages would represent heavy mail system usage, then the average submission time interval for one message would be 2 minutes. The user should take a fraction of this interval (e.g., 1/2) and specify that the messages be submitted at 1 minute intervals.

The experiment is performed using one message originator and one message recipient, each residing on different MTAs. All messages must be submitted according to a user-defined submission time interval, and be of uniform size. The person conducting the experiment should log the submission time of the first message and the delivery time of the last message.

(3) User Input

The user must provide the person conducting the experiment with the following information:

- the experiment environment (i.e., ideal or real)
- the size of the messages to be transferred
- the number of messages to be transferred
- the message submission time interval

(4) Example

The government agency in Experiment #1 has two MTAs at the Overton location, but only one MTA is connected to the WAN. It is of interest to the agency to determine if heavy usage of the mail system warrants a WAN connection for the second MTA. Based on a previous analysis, the agency has determined that during any given hour, not more than 80 messages will be submitted to the mail system. For this experiment the agency requests the transfer of 80 messages in an ideal environment using a message size of 2000 bytes. Messages should be submitted approximately every 22 seconds. The person conducting the experiment provides the agency with the following results.

Table 14. Example Results For Experiment #6

Number of Messages	Submission Time For First Message	Delivery Time For Last Message	Transfer Time Interval
80	01:00.00 PM	01:53.45 PM	53 minutes 45 seconds

(5) Summary of Results

For this experiment the following information is provided: the message number, the submission time for the first message, the delivery time for the last message, and the transfer time interval. The transfer time interval, which is the only calculated value, is the delivery time minus the submission time.

As input to this experiment the user estimates the maximum number of messages that would be submitted to the user's mail system during a certain period of time. Comparing the transfer time interval of this experiment with the time period selected by the user, the user can determine whether the MHS implementation can satisfactorily transfer the maximum number of messages.

The results recorded in this experiment do not represent actual data collected. They are listed for example purposes only. While the results simulate data obtained from one vendor's MHS implementation, the experiment must be repeated for each vendor's MHS implementation which is of interest to the user.

3.4.1.7 Experiment 7

(1) Introduction

This experiment measures the utilization of a system by an MHS implementation. System utilization is useful for determining the percentage of the CPU and I/O used by the implementation. This experiment can only be performed on multi-user systems capable of calculating the percentage of CPU and I/O used by a process or processes.

(2) Methodology

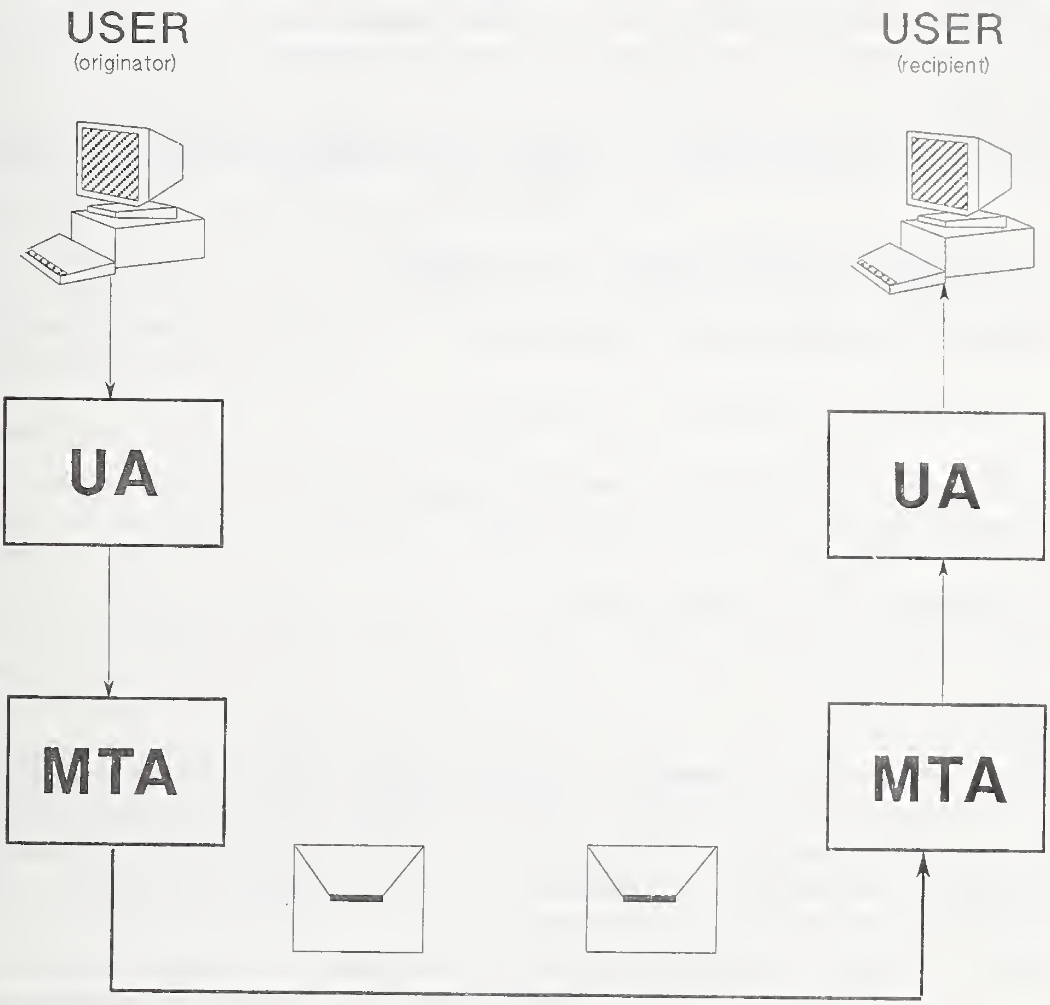


Figure 23. Experiment 7 Message Transfer Path.

For this experiment a user selects an arbitrary period of time (e.g., 1 h), then estimates a typical number of MHS messages that would be submitted to the mail system during that time period. The user inputs both the time period and number of messages to the experiment. In addition, the user specifies a submission time interval for the messages. This interval should be a fraction (e.g., 1/2) of the average submission time interval for one message, which is calculated by dividing the time period selected by the user by the number of messages. For example, if the user selects a time period of 1 hour, and estimates that the submission of 30 messages would represent heavy mail system usage, then the average submission time interval for one message would be 2 minutes. The user should take a fraction of this interval (e.g., 1/2) and specify that the messages be submitted at 1 minute intervals.

The experiment is performed using one message originator and one message recipient, each residing on different MTAs. All messages must be submitted according to a user-defined message submission time interval, and be of uniform size.

(3) User Input

The user must provide the person conducting the experiment with the following information:

- the experiment environment (i.e., ideal or real)
- the size of the messages to be transferred
- the number of messages to be transferred
- the period of time during which the messages are to be transferred
- the message submission time interval

(4) Example

The government agency described in Experiment #1 transfers information requests and billing information between its three locations. In addition to the mail system, the Overton location runs database programs to record these requests and bills. Since the database programs utilize much of the Overton system's I/O, it is of interest to the agency to measure the CPU and I/O used by the MHS implementation.

Based on a previous analysis, the agency has determined that during 1 hour, 40 messages would typically be submitted to the mail system. For this experiment the agency requests the transfer of 40 messages using a message size of 2000 bytes. The messages are to be transferred in an ideal environment for a 1 hour time period. Messages should be submitted approximately every 45 seconds. The person conducting the experiment provides the agency with the following results.

Table 15. Example Results For Experiment #7

Number of Messages	Period of Time Allowed For Message Transfers	Percentage Of CPU Utilization	Percentage Of I/O Utilization
40	60 minutes	15 %	25 %

(5) Summary of Results

For this experiment the following information is provided: the number of messages transferred, the period of time allowed for the transfers, and the percentages of CPU and I/O utilization. The CPU and I/O utilization percentages are provided by the person conducting the experiment. These percentages can be used to determine the percentage of CPU and I/O remaining for other user applications, which will run simultaneously with the MHS implementation. The user must also consider that increasing the number of applications will result in increased operating system overhead. This is because the operating system must grant and relinquish CPU and I/O control to the different applications.

The results recorded in this experiment do not represent actual data collected. They are listed for example purposes only. While the results simulate data obtained from one vendor's MHS implementation, the experiment must be repeated for each vendor's MHS implementation which is of interest to the user.

3.4.2 Mandatory Performance Requirements

This section recommends a procedure for eliminating the candidate MHS implementations which do not meet the mandatory performance requirements of the user. The user should create a list of performance measurements which must be obtainable by a candidate MHS implementation, for that implementation to be acceptable. The user should be careful not to list as mandatory, specific performance measurements which are instead highly desirable because this can unnecessarily restrict the list of candidate MHS implementations. This list may be created by reviewing the experiments, noting which performance measurements are mandatory for the user. Once the list is created, the user should verify that the candidate MHS implementations can obtain the performance measurements required by the user. A candidate MHS implementation which can not obtain the mandatory performance measurements of the user should be removed from the list of implementations at this point. As an example, if after reviewing the experiments, the user decides that the candidate MHS implementations must be capable of sending 100 1000 byte messages in an hour (Experiment 5), then MHS implementations which are not capable of sending 100 1000 byte messages in an hour are unacceptable to the user and should no longer be evaluated.

3.4.3 Performance Evaluation

This section recommends a procedure for determining which of the candidate MHS implementations, obtaining all required performance measurements, best meets the performance requirements of the user. First, the user must assign weights to each experiment based on how important the performance measurement determined by the experiment is to the user. This procedure is defined in section 3.4.3.1. Second, the user must rate each of the candidate MHS implementations, based on the weights assigned by the user and the performance measurements obtained for the candidate MHS implementations. This procedure is defined in section 3.4.3.2.

3.4.3.1 Weighing Performance

This section provides guidelines for assigning weights to each experiment based on how important the performance measurement determined by the experiment is to the user. The user should balance each of the experiments, and determine how important each experiment is to the user. This step results in the assignment of a weight to each experiment.

The experiments must be balanced because the maximum score which may be received by one experiment has no relation to the maximum score which may be received by the other experiments. For example, one experiment may be scored subjectively on a scale of 0 to 10. Then this experiment can receive a maximum score of 10. Another experiment may be subjectively scored as 0 or 1. Then this experiment can receive a maximum score of 1. If these two example experiments are of equal importance to the user, then the weight of the second experiment must be 10 times as large as the weight of the first experiment, in order to balance the experiments.

The user must determine relative levels of importance of the experiments. The user first assigns a maximum score to each experiment to reflect its importance to the user. For example, the user may decide that an experiment which is very important to the user can receive a maximum score of 100 points, an experiment which is important to the user can receive a maximum score of 50 points, and an experiment which is not of any importance to the user can only receive a score of 0 points.

The user must then compute a weight for each experiment. The weight for each experiment is calculated by dividing the maximum score the user has determined that the experiment can receive by the maximum score of the experiment. As an example the user may consider Experiment 1 to be very important (100 points), and Experiment 2 to be important (50 points), and Experiment 3 to be of no importance (0 points). If the maximum score of Experiment 1 is 10, then it is assigned a weight of 100/10 which equals 10. If the maximum score of Experiment 2 is 1, then it is assigned a weight of 50/1 which equals 50. If the maximum score of Experiment 3 is 10, then it is assigned a weight of 0/10 which equals 0.

3.4.3.2 Performance Evaluation Rating

This section recommends a procedure to rate the performance of the candidate MHS implementations, based on the weights assigned by the user and the performance measurements obtained for the candidate MHS implementations. The performance measurements for a candidate MHS implementation are obtained by performing the appropriate experiments. This performance rating procedure must be repeated for each candidate MHS implementation. The procedure for determining a performance rating for each candidate MHS implementation follows.

First, the user must score each experiment performed on a candidate MHS implementation. There are two procedures recommended for scoring an experiment.

(1) The user can subjectively score the experiment based on the user's overall impression of the measurement obtained by the experiment for the candidate MHS implementation. As an example, the user may decide to rate the experiment, performed on a candidate MHS implementation, on a scale of 0-10 where a score of 10 indicates that the measurement obtained by the experiment for the candidate MHS implementation meets all of the user's performance requirements for this experiment, a score of 5 indicates that the measurement obtained by the experiment for the candidate MHS implementation meets some of the user's performance requirements for this experiment, and a score of 0 indicates that the measurement obtained by the experiment for the candidate MHS implementation

does not meet any of the user's performance requirements for this experiment.

(2) The user can subjectively score the experiment based on the user's overall impression of whether or not the measurement obtained by the candidate MHS implementation is acceptable. If the measurement obtained from the experiment for the candidate MHS implementation is acceptable then it will receive a passing score (or 1); otherwise it will receive a failing score (or 0).

Second, the user determines the total performance rating, for a candidate MHS implementation, by summing the weight of each experiment times the score for that experiment, for each of the experiments. For example if there are experiments X, Y, and Z and experiment X has a weight of 5, experiment Y has a weight of 3, and experiment Z has a weight of 1, and an implementation has a score of 25 for experiment X, a score of 50 for experiment Y, and a score of 75 for experiment Z, then the performance rating for the implementation is ((5*25) + (3*50) + (1*75)) which equals 350. The equation for determining the total performance rating is specified in table 16. The candidate MHS implementation with the highest score is likely to be the "best" implementation, with respect to performance, for that user. Note that these ratings are not absolute ratings; another user might rate the same candidate MHS implementations differently based on a different set of requirements.

Table 16. Equation for Performance Rating MHS Implementations

$$RP = \sum_{i=1}^{m} (E_i * WE_i)$$

$RP = Total\ Performance\ Rating$

$E_i = Rating\ of\ Experiment\ i$

$WE_i = Weight\ of\ Experiment\ i$

$m = Number\ of\ Experiments$

3.5 Guidelines for Rating Implementations

This section recommends a procedure for rating the candidate MHS implementations based on the functional and performance evaluations. If the user did not require the performance of the candidate MHS implementations to be evaluated, the total score for each implementation is the functional evaluation total. Otherwise, to arrive at a total score for each implementation, the user must weigh the functional evaluation totals and the performance evaluation totals, and compute the total score. As an example, let us assume that the user decides that the functional evaluation is twice as important as the performance evaluation. Then the total score for each MHS implementation is 2*(functional evaluation total score) + (performance evaluation total score). The MHS implementation which receives the highest total score is probably the best MHS implementation for the user. The equation for determining the total rating for the candidate MHS implementations is specified in table 17.

Table 17. Equation for Total Rating MHS Implementations

$$R = (RF * WF) + (RP * WP)$$

R = *Total Rating*

RF = *Total Functional Rating*

WF = *Weight of Functional Rating*

RP = *Total Performance Rating*

WP = *Weight of Performance Rating*

3.6 Example Evaluation

This section provides an example evaluation of MHS implementations using the previously described guidelines, and contains the following sections: Section 3.6.1 provides an example functional evaluation of MHS implementations, section 3.6.2 provides an example performance evaluation of MHS implementations, and section 3.6.3 provides an example rating of MHS implementations based on the functional and performance evaluations.

In this example the user has selected four candidate MHS implementations to evaluate, which are referred to as implementations A, B, C, and D. The user has obtained product literature, users manuals, technical specifications, and other available information from each of the vendors for the candidate MHS implementations, in order to perform the evaluation.

3.6.1 Example Functional Evaluation

This section provides an example functional evaluation of the candidate MHS implementations using the previously described guidelines. Appendix B provides, in a tabular format, a complete listing of the functionality potentially available in MHS implementations, as described in section 3.3.1. This format is useful for performing the functional evaluation. This section contains the following sections: Section 3.6.1.1 provides an example of the procedure for eliminating candidate MHS implementations which do not meet mandatory functional requirements of the user. Section 3.6.1.2 provides an example of the procedure recommended for determining which candidate MHS implementation best meets the functional requirements of that user.

3.6.1.1 Example Mandatory Functional Requirements

This section provides an example of the procedure for eliminating candidate MHS implementations which do not meet the mandatory functional requirements of the user. A list of the functions which must be included in the candidate MHS implementations, for them to be acceptable is created, by reviewing the functions in each of the functional categories, noting which functions are mandatory for the user. In this example the user has determined that the Candidate MHS implementations must support Transport Class 0 over the Connection Oriented Network Service over an X.25 Wide Area Network, Transport Class 4 over the Connectionless Oriented Network Service over an 802.3 Local Area Network, Transport Class 4 over the Connectionless Oriented Network Service over an X.25 Wide Area Network, and the MTA connecting to multiple remote MTAs. The example mandatory functions are referred to as functions 25.2, 25.3, 25.4, 25.5, 25.6, and 22.1. (See app. B for the tabular listing of functions.)

Next, the user verifies that each candidate MHS implementation contains the mandatory functions. Any candidate MHS implementation which does not contain all of the mandatory functions is removed from the list of candidate MHS implementations at this point. Table 18 indicates which of the mandatory functions are contained in the candidate MHS implementations.

Table 18. Example User Mandatory MHS Functions

Function Number	Impl A	Impl B	Impl C	Impl D
25.2	Yes	Yes	Yes	Yes
25.3	Yes	Yes	Yes	Yes
25.4	Yes	Yes	Yes	No
25.5	Yes	Yes	Yes	No
25.6	Yes	Yes	Yes	No
22.1	Yes	Yes	Yes	Yes

Since implementation D does not support functions 25.4, 25.5, and 25.6 which are mandatory for the user, this implementation is removed from the list of candidate MHS implementations.

3.6.1.2 Example Functional Evaluation

This section provides an example of the procedure for determining which of the candidate MHS implementations best meets the functional requirements of that user. Four examples are provided to demonstrate the procedure for weighing the functions within a category and scoring a category, using each of the weighing options and their corresponding scoring algorithms. The scores for the remaining categories are provided without the details of how they were derived. One example is provided to demonstrate the procedure for weighing the categories and scoring the implementations. The example rating is derived by the following steps:

(1) The user must weigh each of the functions within a category using one of the options previously recommended. The user must score the category using the equation corresponding to the weighing option selected. This procedure is repeated for each category. An example rating of the functions within a category based on each of the options previously recommended follows.

An example is provided for weighing and scoring the functions within a category based on Option 1. Option 1 recommends determining a weight for each individual function in a category based on how important that function is to the user. The score of the category is the sum of the weight of each function which is present in the implementation. (See table 3 for the equation.) The Saving Messages category is used in this example. The user's requirements for this example are: Providing a storage facility for messages (14.1), allowing a hierarchical partitioning of the mail storage facility (14.3), and creating a mail storage unit (14.5) are very important and are assigned a weight of 3. Displaying a mail storage unit (14.6), renaming a mail storage unit (14.8), copying a mail storage unit (14.9), purging the contents of a mail storage unit (14.10), and deleting a mail storage unit (14.11) are important and are assigned a weight of 2. Having a default mail storage unit (14.4), Displaying statistical information about a mail storage unit (14.7), listing the messages saved inside a mail storage unit (14.12), writing messages saved inside a mail storage unit to a file (14.15), copying messages between mail storage units (14.16), and moving messages between mail storage units (14.17) are less important and are assigned a weight of 1. Allowing a flat partitioning of the mail storage facility (14.2), sorting the messages saved inside a mail storage unit (14.13), and editing messages saved inside a mail storage unit (14.14), are not of any importance and assigned a weight of 0. Table 19 contains the functions in the category, their weight for this example, their availability in each of the implementations, and a total score for this category for each implementation.

Table 19. Example Category Rating, Option 1

Function Number	Weight	Impl A	Impl B	Impl C
14.1	3	Yes	Yes	Yes
14.2	0	Yes	Yes	Yes
14.3	3	Yes	No	No
14.4	1	Yes	No	Yes
14.5	3	Yes	Yes	Yes
14.6	2	Yes	Yes	Yes
14.7	1	Yes	No	Yes
14.8	2	Yes	No	No
14.9	2	No	No	Yes
14.10	2	Yes	Yes	Yes
14.11	2	Yes	Yes	Yes
14.12	1	Yes	Yes	Yes
14.13	0	Yes	No	No
14.14	0	Yes	No	No
14.15	1	Yes	Yes	No
14.16	1	Yes	Yes	Yes
14.17	1	Yes	Yes	No
Total	25	23	16	18

In this example, out of a possible 25 points, implementation A scored 23, implementation B scored 16, and implementation C scored 18.

An example is provided for weighing and scoring the functions within a category based on Option 2. Option 2 recommends assigning each function in a category, which is of interest to the user, a weight of 1, and all other functions in the category a weight of 0. This option assumes that the user is either interested or not interested in a function, and that each of the functions of interest to the user are of equal importance. The score of the category is the sum of the number of functions, which are important to that user, and are present, in the implementation. (See table 3 for the equation.) The Optional UA Functions category is used in this example. The user's requirements for this example are: The Authorizing Users (3.1), Blind Copy (3.3), Cross Referencing (3.5), Expiry Date (3.6), Forwarded Message (3.7), Importance (3.8), Multi-part Body (3.9), Obsolete (3.10), Reply Request (3.11), Sensitivity (3.12) functions are important to the user. The remaining functions are not important to the user. Table 20 contains the functions in the category, their weight for this example, their availability in each of the implementations, and a total score for this category for each implementation.

Table 20. Example Category Rating, Option 2

Function Number	Weight	Impl A	Impl B	Impl C
3.1	1	Yes	Yes	Yes
3.2	0	Yes	No	Yes
3.3	1	Yes	Yes	Yes
3.4	0	No	No	No
3.5	1	Yes	No	No
3.6	1	Yes	Yes	Yes
3.7	1	Yes	Yes	Yes
3.8	1	Yes	Yes	Yes
3.9	1	Yes	No	No
3.10	1	Yes	Yes	Yes
3.11	1	Yes	Yes	No
3.12	1	Yes	Yes	Yes
3.13	0	Yes	Yes	Yes
3.13	0	Yes	Yes	Yes
Total	10	10	8	7

In this example, out of a possible 10 points, implementation A scored 10, implementation B scored 8, and implementation C scored 7.

An example is provided for weighing and scoring the functions within a category based on Option 3. Option 3 recommends not assigning any weight to the functions in the category. The user subjectively scores the category based on the user's overall impression of how well that category is represented in the candidate MHS implementation. (See table 3 for the equation.) The Listing a Summary of Messages category is used in this example. The user's requirements for this example are to be able to list a standard summary of messages in the default mail storage unit as well as other mail storage units. Also the user would like to have the capability to sort the summary of messages by various criteria such as message importance, message delivery time. An implementation which meets all of the users needs in this category will receive a score of 10, an implementation which meets half of the users needs in this category will receive a score of 5, and an implementation which meets none of the users needs in this category will receive a score of 0. Table 21 contains the subjective score for this category for each implementation.

Table 21. Example Category Rating, Option 3

	Max Score	Impl A	Impl B	Impl C
Subjective Score	10	8	2	5

In this example, out of a possible 10 points, implementation A scored 8, implementation B scored 2, and implementation C scored 5.

An alternate example is provided for weighing and scoring the functions within a category based on Option 3. Option 3 recommends not assigning any weight to the functions in the category. The user subjectively scores the category based on the users overall impression of whether or not the candidate MHS implementation acceptably performs the functions in this category. If the candidate MHS implementation acceptably performs the functions in this category it will receive a passing score (or 1); otherwise it will receive a failing score (or 0). (See table 3 for the equation.) The Deleting Messages category is used in this example. The user's requirements for this example are to be able to delete messages and to be able to retrieve deleted messages which have not been discarded. Table 22 contains the subjective score for this category for each implementation.

Table 22. Example Category Rating, Option 3

	Max Score	Impl A	Impl B	Impl C
Subjective Score	1	1	0	1

In this example, implementations A and C passed and received a score of 1, and implementation B failed and received a score of 0.

(2) The user assigns a maximum rating to each category of functions, to indicate how important the category of functions is to the user. The user then computes the weight for each category by dividing the maximum rating that the category can receive by the maximum score of the functions within the category. The user determines the total functional rating, for a candidate MHS implementation, by summing the weight of each category times the score for that category, for each of the categories. (See table 4 for the equation.) The user in this example has decided to assign ratings to the categories so that a category which is extremely important to the user can receive a maximum score of 400 points, a category which is very important to the user can receive a maximum score of 300 points, a category which is important to the user can receive a maximum score of 200 points, a category which is less important to the user can receive a maximum score of 100 points, and a category which is not of any importance to the user can only receive

a score of 0 points. The user has decided that the Optional UA Functions, Composing Messages, Sending Messages, Receiving Messages, Listing a Summary of Messages, Reading Messages, Saving Messages categories are extremely important, the Optional MTA Functions, O/R Name Types, Message Body Parts, Aliases, Distribution Lists categories are very important, Deleting Messages, Printing Messages, On-Line Help Facilities, Administrative Functions, Remote MTA Connections, Certification, Underlying OSI Layers, Hardware Requirements, Software Requirements categories are important, the Default Configuration Profile, UA Interface, System Interface, Access Control, Debug Capabilities, Documentation categories are less important, the Pragmatic Constraints Functions category is not important. The weight for each category is calculated using the previously indicated algorithm. Table 23 contains the MHS functional categories (Category), the maximum score the user has determined that the categories can receive for this example (Importance), the maximum score of the functions within the categories for this example (Raw Score), the computed weight of the functional categories for this example (Weight), and the total functional rating for each of the implementations. Note, since categories 1 through 3 describe mandatory functions, they are not listed in the evaluation.

Table 23. Example MHS Implementation Functional Rating

Category	Importance	Raw Score	Weight	Impl A	Impl B	Impl C
3	400	10	40	400	320	280
4	300	6	50	300	240	180
5	300	1	300	300	300	300
6	300	1	300	300	300	300
7	400	10	40	400	240	320
8	400	10	40	400	320	320
9	300	10	30	300	180	150
10	300	10	30	300	180	150
11	400	10	40	320	320	320
12	400	10	40	320	80	200
13	400	20	20	360	320	320
14	400	25	16	368	256	288
15	200	1	200	200	0	200
16	200	1	200	200	200	200
17	100	20	5	90	50	90
18	200	10	20	200	160	100
19	100	10	10	100	20	80
20	100	10	10	100	50	50
21	200	10	20	180	100	100
22	200	10	20	100	200	200
23	100	1	100	100	100	100
24	100	10	10	60	80	40
25	200	10	20	180	200	200
26	200	1	200	200	200	200
27	200	10	20	200	160	80
28	200	10	20	200	160	160
29	100	10	10	90	70	70
30	0	0	0	0	0	0
Total	6700	-	-	6260	4806	4998

Candidate MHS implementation A received the highest score in this example, and is likely to be the "best" implementation, functionally, for that user. Note that these ratings are not absolute ratings; another user might rate the same candidate MHS implementations differently based on a different set of requirements.

3.6.2 Example Performance Evaluation

This section provides an example performance evaluation of the candidate MHS implementations using the previously described guidelines. Appendix C provides, in a tabular format, a complete listing of the measurements potentially available in MHS implementations, as described in section 3.4.1. This format is useful for performing the performance evaluation. This section contains the following sections: Section 3.6.2.1 provides an example of the procedure for eliminating candidate MHS implementations which do not meet mandatory performance requirements of the user. Section 3.6.2.2 provides an example of the procedure recommended for determining which candidate MHS implementation best meets the performance requirements of that user.

3.6.2.1 Example Mandatory Performance Requirements

This section provides an example of the procedure for eliminating candidate MHS implementations which do not meet the mandatory performance requirements of the user. A list of the measurements which must be obtainable by the candidate MHS implementations, for them to be acceptable is created, by reviewing each of the experiments, noting which measurements are mandatory for the user. In this example the user has determined that the Candidate MHS implementations must be able to send up to 100 messages in an hour. The example mandatory measurements are determined by experiment 6. (See app. C for the tabular listing of experiments.)

Next, the user verifies that each candidate MHS implementation can obtain the mandatory measurements. Any candidate MHS implementation which cannot obtain all of the mandatory measurements is removed from the list of candidate MHS implementations at this point. Table 24 indicates the measurements obtained from the candidate MHS implementations.

Table 24. Example User Mandatory MHS Measurements

Experiment Number	Impl A	Impl B	Impl C
6	120	115	110

Since all of the implementations can transmit up to 100 messages per hour, no implementations are removed from the list of candidate MHS implementations at this point.

3.6.2.2 Example Performance Evaluation

This section provides an example of the procedure for determining which of the candidate MHS implementations best meets the performance requirements of that user. Two examples are provided to demonstrate the procedure for scoring the measurements obtained by an experiment, using the two scoring algorithms. The scores for the remaining measurements are provided without the details of how they were derived. One example is provided to demonstrate the procedure for weighing the measurements and scoring the implementations. The example rating is derived by the following steps:

(1) The user must score each experiment performed on a candidate MHS implementation. An example of each of the two procedures recommended for scoring an experiment follows.

In this first example, the user subjectively scores the experiment based on the user's overall impression of the measurement obtained by the experiment for the candidate MHS implementation. Experiment 6 is used in this example. The user's requirement for this example is to be able to send up to 100 messages per hour. A score of 10 indicates that the measurement obtained by the experiment for the implementation meets all of the user's performance requirements for this experiment, a score of 5 indicates that the measurement obtained by the experiment for the implementation meets some of the user's performance requirements for this experiment, and a score of 0 indicates that the measurement obtained by the experiment for the implementation does not meet any of the user's performance requirements for this experiment. Table 25 contains the subjective score for this experiment for each implementation.

Table 25. Example Experiment Rating

	Max Score	Impl A	Impl B	Impl C
Subjective Score	10	10	8	6

In this first example, out of a possible 10 points, implementation A scored 10, implementation B scored 8, and implementation C scored 6.

In this second example, the user subjectively scores the experiment based on the user's overall impression of whether or not the measurement obtained by the candidate MHS implementation is acceptable. If the measurement obtained from the experiment for the candidate MHS implementation is acceptable then it will receive a passing score (or 1); otherwise it will receive a failing score (or 0). Experiment 1 is used in this example. The user's requirement for this example is to be able to send a message in less than 15 minutes. Table 26 contains the subjective score for this experiment for each implementation.

Table 26. Example Experiment Rating

	Max Score	Impl A	Impl B	Impl C
Subjective Score	1	1	1	1

In this example, implementations A, B, and C passed and received a score of 1.

(2) The user assigns a maximum rating to each experiment, to indicate how important the measurement obtained by the experiment is to the user. The user then computes the

weight for each experiment by dividing the maximum rating that the experiment can receive by the maximum score of the measurement obtained by the experiment. The user determines the total performance rating, for a candidate MHS implementation, by summing the weight of each experiment times the score for that experiment, for each of the experiments. (See table 16 for the equation.) The user in this example has decided to assign ratings to the experiments so that an experiment which is very important to the user can receive a maximum score of 100 points, an experiment which is important to the user can receive a maximum score of 50 points, and an experiment which is not of any importance to the user can only receive a score of 0 points. The user has decided that Experiment 6 is very important, Experiments 1 - 5 are important, and Experiment 7 is not of any importance. The weight for each experiment is calculated using the previously indicated algorithm. Table 27 contains the MHS experiment number (Experiment), the maximum score the user has determined that the experiment can receive for this example (Importance), the maximum score of the measurement obtained by the experiment for this example (Raw Score), the computed weight of the experiments for this example (Weight), and the total performance rating for each of the implementations.

Table 27. Example MHS Implementation Performance Rating

Experiment	Importance	Raw Score	Weight	Impl A	Impl B	Impl C
1	50	1	50	50	50	50
2	50	1	50	50	50	50
3	50	1	50	50	50	50
4	50	1	50	50	50	50
5	50	1	50	50	50	50
6	100	10	10	100	80	60
7	0	0	0	0	0	0
Total	350	-	-	350	330	310

Candidate MHS implementation A received the highest score in this example, and is likely to be the "best" implementation, for performance, for that user. Note that these ratings are not absolute ratings; another user might rate the same candidate MHS implementations differently based on a different set of requirements.

3.6.3 Example Rating

This section provides an example of the procedure recommended for rating the candidate MHS implementations based on the functional and performance evaluations. To arrive at a total score for each implementation, the user must weigh the functional evaluation totals and the performance evaluation totals, and compute the total score. (See table 17 for the equation.) The user in this example has decided the functional evaluation is 10 times as important as the performance evaluation, and therefore assigns the functional evaluation weight as 10 and the performance evaluation weight as 1. Table 28 contains the functional and performance weights, functional and performance scores for each of the implementations, and the total rating for each of the implementations.

Table 28. Example MHS Implementation Total Rating

	Weight	Impl A	Impl B	Impl C
Functional	10	6260	4806	4998
Performance	1	350	330	310
Total	-	62950	48390	50290

Candidate MHS implementation A received the highest score in this example, and is probably the "best" MHS implementation for the user.

3.7 Other Guidelines

This section describes other factors to consider when evaluating candidate MHS implementations. The guidelines defined in this section are not as concrete as the ones in the previous sections, and therefore, are not in the functional or performance evaluation sections. They are, however, factors to be considered when evaluating implementations.

This section contains five major topics. The first two topics, effectiveness and flexibility, relate to the MHS implementation. The second two topics, commitment and support, relate to the vendor, and the final topic relates to cost.

One factor to consider when evaluating an MHS implementation is the effectiveness of the functionality provided by the implementation. For example, a user may be provided a program to assist with installing the implementation; however, the installation procedure may be very difficult and time consuming despite the installation program. Also, an implementation may provide so many functions that a simple task such as sending a one line message to one recipient may be difficult to understand or perform. Debugging functions may exist, but may be inadequate to easily solve problems encountered. Finally, the documentation provided with an implementation may not be well organized, or may be difficult to understand.

To better appreciate the effectiveness of an MHS implementation, the user has several options. The user can request a copy of the MHS documentation from the vendor. By examining the documentation in advance, the user can better determine its adequacy and understandability. The user may also be able to determine how easy the implementation is to install, configure, debug, and use. Another option is for the user to request a demonstration of the MHS implementation. A demonstration will provide an overall view of the implementation, especially concerning its "user friendliness."

A second factor to consider when evaluating an MHS implementation is the flexibility of the implementation. Flexibility pertains to the capability of a mail system to continue performing acceptably during routine maintenance (e.g., updating routing or user information, or backing up the mail system). For example, it may be important to a user that the MHS implementation continue processing messages while the system administrator adds new users to the mail system. Again, requesting an advance copy of the documentation may inform the user of the flexibility of the mail system.

Other evaluation factors relate to the vendor. The user should consider the company's commitment to OSI, and if the personal contacts (i.e., sales and service representatives) are well informed. The user may consider whether the company marketing the product also developed the product. The user should also consider the company's ability to service their product, and the company policy regarding supplying upgrades to their product. Customer service issues include the following: software support, whether the support is local or out of town, and maintenance agreements. The user should ask the vendor about the type and extent of customer support that is available.

The final topic concerning the evaluation of an MHS implementation is the cost of the implementation. This includes hardware costs, such as computer systems, LAN cards, WAN cards, etc., software costs, and maintenance costs such as maintenance contracts. The budget of the user will determine the importance of cost as an evaluation factor.

4. Conclusion

As stated in the Introduction, the intent of this document is to advance the goals of the GOSIP by providing guidelines for evaluating MHS implementations. These guidelines can assist a user in determining which implementation, among several candidates, will best meet the functional and performance requirements of that user.

The guidelines for evaluating MHS implementations contain a procedure for rating MHS implementations based on the user's requirements, and a list of factors which can not be rated by the user, but should be taken into consideration when selecting an MHS implementation.

The procedure for rating MHS implementations includes procedures for evaluating the functional and performance capabilities of MHS implementations, based on the user's requirements, and a procedure for rating MHS implementations based on their functional and performance evaluations. The functional evaluation is important because MHS implementations may greatly vary in the non-standard and optional standard functionality that they provide. The performance evaluation is important for users who will be extensively using the MHS implementation, because MHS implementations may vary in their level of performance. Users who will not be extensively using the MHS implementation may not be as concerned with the performance of candidate MHS implementations. If the user evaluates the performance of the candidate MHS implementations, the overall rating of these implementations is a combination of the functional and performance evaluation. Otherwise the overall rating of these implementations is the functional evaluation.

It is recommended that users follow these guidelines for selecting an MHS implementation, in order to procure the "best" MHS implementation for their needs.

APPENDIX A: Lab Configuration

An MHS laboratory, containing a representative sample of the MHS implementations currently available, was set up. Figure 24 depicts the configuration of the MHS Laboratory used in this project.

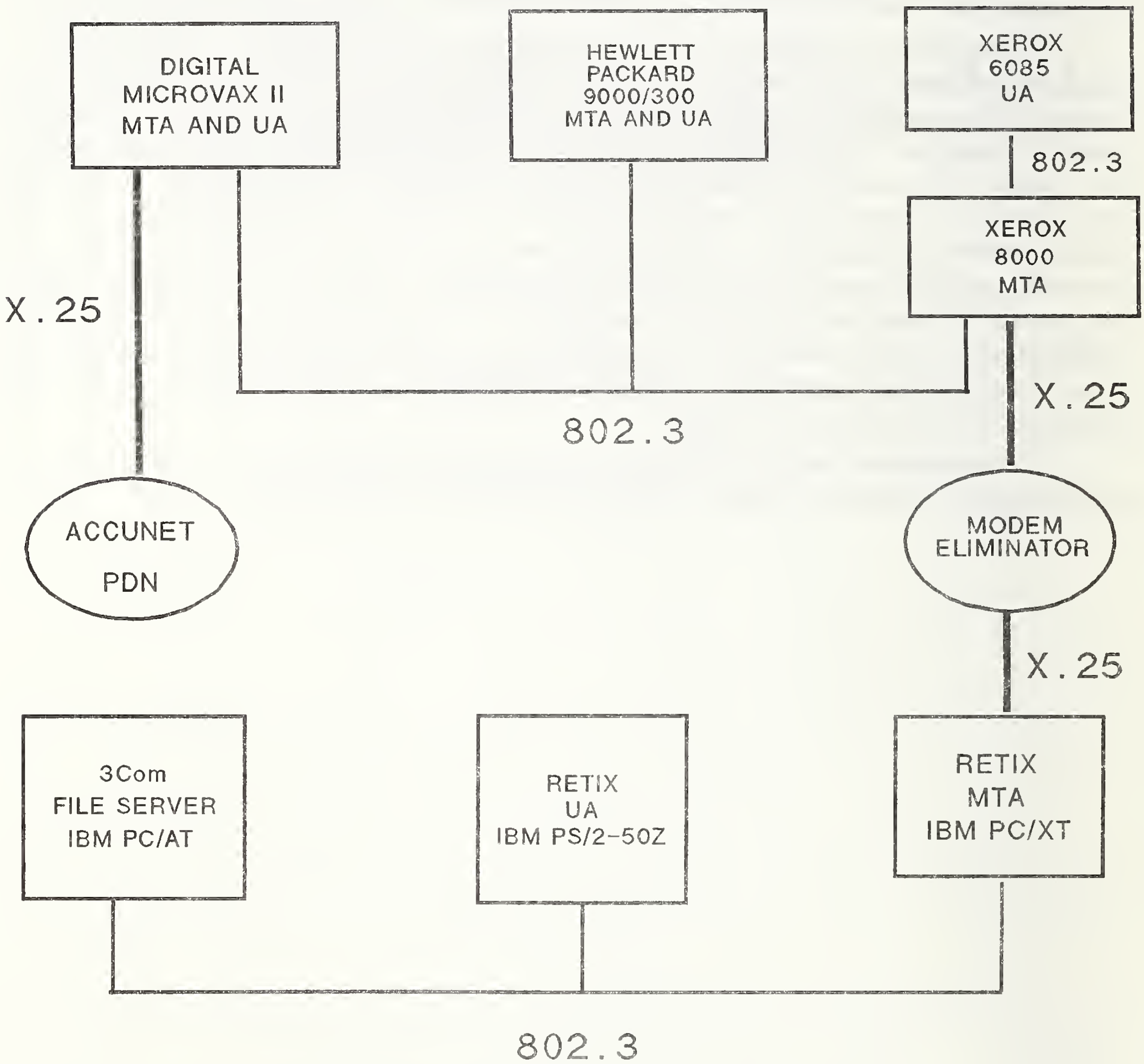


Figure 24. NIST Network Applications Lab.

APPENDIX B: Tables of Functions

This appendix contains a listing of the MHS functions described in these guidelines. The functions are presented here in tabular form. Each table consists of a title and a list of entries. The title corresponds to a category of functions detailed in section 3.3.1. The list of entries correspond to the set of functions contained in that category. Each function listed is preceded by two numbers separated by a period. The numbers to the left of the periods match a category subsection of 3.2.1. The numbers to the right of the periods represent a numerical listing of functions within a category.

Mandatory UA Functions
The Following Functions are Mandatory For Originating and Recipient UAs
1.1 IP-message Identification
1.2 Originator
1.3 Primary and Copy Recipients
1.4 Replying Message
1.5 Subject
1.6 Typed Body
The Following Functions are Mandatory For Recipient UAs and Optional for Originating UAs
1.7 Authorizing Users
1.8 Auto-forwarded
1.9 Blind Copy
1.10 Body Part Encryption
1.11 Cross Referencing
1.12 Expiry Date
1.13 Forwarded Message
1.14 Importance
1.15 Multi-part Body
1.16 Obsolete
1.17 Reply Request
1.18 Sensitivity

Mandatory MTA Functions
2.1 Alternate Recipient Allowed
2.2 Content Type
2.3 Conversion Prohibition
2.4 Converted
2.5 Delivery Notification
2.6 Delivery Time Stamp
2.7 Disclosure of Recipients
2.8 Grade of Delivery
2.9 Message Identification
2.10 Multi-destination Delivery
2.11 Non-delivery Notification
2.12 Origination Encoded Information Types
2.13 Submission Time Stamp
2.14 Class 3 MTA

Optional UA Functions	
	The Following Functions are Optional for Originating UAs and Mandatory For Recipient UAs
3.1	Authorizing Users
3.2	Auto-forwarded
3.3	Blind Copy
3.4	Body Part Encryption
3.5	Cross Referencing
3.6	Expiry Date
3.7	Forwarded Message
3.8	Importance
3.9	Multi-part Body
3.10	Obsolete
3.11	Reply Request
3.12	Sensitivity
	The Following Functions are Optional for Originating and Recipient UAs
3.13	Non-receipt Notification
3.14	Receipt Notification

Optional MTA Functions	
4.1	Access Management
4.2	Alternate Recipient Assignments
4.3	Deferred Delivery
4.4	Deferred Delivery Cancellation
4.5	Disclosure of Recipients
4.6	Explicit Conversion
4.7	Hold for Delivery
4.8	Implicit Conversion
4.9	Prevention of Non-delivery Notification
4.10	Probe
4.11	Return of Contents

O/R Name Types	
5.1	Form 1, Variant 1
5.2	Form 1, Variant 2
5.3	Form 1, Variant 3
5.4	Form 2
5.5	Allow Fully Conformant MHS Name Assignment

Message Body Parts	
6.1	Transfer of Internationally Recognized Body Parts Other than IA5
6.2	Transfer of Private Message Body Parts

Composing Messages
7.1 Internal Text Editor
7.2 External Text Editor
7.3 Internal Text Editor Type
7.4 External Text Editor Type
7.5 Text Editor Selection
7.6 No Text Editor Limitations
7.7 Attach File to Message
7.8 Attach Message to Message
7.9 Modify Attachment
7.10 Fill-in Form for MHS Message Options
7.11 Create Table with MHS Message Option Values
7.12 Specify Who Message Is From
7.13 Save Composed Message

Sending Messages
8.1 Save Copy of Outbound Message
8.2 Send Multiple Messages with One Command
8.3 Return Contents of Original Message with Reply
8.4 Reply To All O/R Names in Message
8.5 Prepend Explanatory Note to Forwarded Message
8.6 Modify Message Being Forwarded
8.7 Invalid Recipient Instructions
8.8 Send File as Message
8.9 Send File as Reply

Aliases	
9.1	Use Personal Alias
9.2	Create Personal Alias
9.3	Display Personal Alias
9.4	Modify Personal Alias
9.5	Delete Personal Alias
9.6	Create Multiple Personal Aliases for Single Recipient
9.7	Expand Personal Alias on Screen
9.8	Create Personal Aliases from O/R Names of Incoming Messages
9.9	Create Personal Aliases for Recipients on Same MTA as User
9.10	Create Personal Aliases for Recipients on Different MTA than User
9.11	Use System Alias
9.12	Create System Alias
9.13	Display System Alias
9.14	Modify System Alias
9.15	Delete System Alias
9.16	Create Multiple System Aliases for a Single Recipient
9.17	Expand System Alias on Screen
9.18	Create System Aliases from O/R Names of Incoming Messages
9.19	Create System Aliases for Recipients on Same MTA as User
9.20	Create System Aliases for Recipients on Different MTA than User

Distribution Lists	
10.1	Use Personal Distribution List
10.2	Allow Personal Distribution List to Contain Aliases
10.3	Allow Personal Distribution List to Contain Other Distribution Lists
10.4	Create Personal Distribution List
10.5	Display Personal Distribution List
10.6	Modify Personal Distribution List
10.7	Delete Personal Distribution List
10.8	Recognize Duplicate Entries in Personal Distribution List
10.9	Create Personal Distribution List for Recipients on Same MTA as User
10.10	Create Personal Distribution List for Recipients on Different MTA than User
10.11	Use System Distribution List
10.12	Allow System Distribution List to Contain Aliases
10.13	Allow System Distribution List to Contain Other Distribution Lists
10.14	Create System Distribution List
10.15	Display System Distribution List
10.16	Modify System Distribution List
10.17	Delete System Distribution List
10.18	Recognize Duplicate Entries in System Distribution List
10.19	Create System Distribution List for Recipients on Same MTA as User
10.20	Create System Distribution List for Recipients on Different MTA than User

Receiving Messages	
11.1	New Mail Notification
11.2	Select Notification type
11.3	Set MTA Polling Frequency
11.4	Manually Query MTA
11.5	Discriminate Message
11.6	Receive Unsupported Body Part
11.7	Automatic Reply
11.8	Redirect Message
11.9	Secondary Address

Listing a Summary of Messages	
12.1	Statistical Indicators
12.2	List According to Criteria
12.3	Perform Operation on Message Listed in Summary

Reading Messages	
13.1	Read Any Message in Any Order
13.2	Read Multiple Messages Conforming to Criteria
13.3	Modify Message
13.4	Redraw Screen
13.5	Automatically Transfer Message After Reading

Saving Messages	
14.1	Provide Storage Facility for Messages
14.2	Allow Flat Partitioning of Facility
14.3	Allow Hierarchical Partitioning of Facility
14.4	Default Mail Storage Units
14.5	Create Mail Storage Unit
14.6	Display Mail Storage Unit
14.7	Display Statistical Information about Mail Storage Unit
14.8	Rename Mail Storage Unit
14.9	Copy Mail Storage Unit
14.10	Purge Contents of Mail Storage Unit
14.11	Delete Mail Storage Unit
14.12	List Messages Saved Inside Mail Storage Unit
14.13	Sort Messages Saved Inside Mail Storage Unit
14.14	Edit Messages Saved Inside Mail Storage Unit
14.15	Write Messages Saved Inside Mail Storage Unit to File
14.16	Copy Messages Between Mail Storage Units
14.17	Move Messages Between Mail Storage Units

Deleting Messages	
15.1	Purge Wastebasket on Command
15.2	Purge Wastebasket when Exiting MHS Application
15.3	Recover Deleted Mail

Printing MHS Information	
16.1	Print Messages
16.2	Print Local User List
16.3	Print Aliases
16.4	Print Distribution Lists
16.5	Select Printer
16.6	Select Time Printing Is to Begin
16.7	Select Output Format
16.8	Select Number of Copies

Default Configuration profile	
17.1	Provide Default Configuration Profile
17.2	MHS Message Options in Profile
17.3	User's Working Environment in Profile
17.4	Display Profile Entries
17.5	Modify Profile Entries

On-line Help Facilities	
18.1	Provide On-line Help Facility

UA Interface	
19.1	Type of Interface
19.2	Menu Options for Command-driven Interface
19.3	Select Sophistication Level
19.4	Programmatic Interface to UA
19.5	Programmatic Interface to MTA
19.6	POSIX Conformant UA Programmatic Interface
19.7	POSIX Conformant MTA Programmatic Interface

System Interface	
20.1	Execute Operating System Command
20.2	Use MHS Message as Input to Operating System Command
20.3	Use MHS Message to Trigger Operating System Program
20.4	Write MHS Message to File
20.5	Create File as Outbound Body Part
20.6	Send File as Message
20.7	Send File as Reply
20.8	Send File Containing Binary Data

Administrative Functions	
21.1	On-line Training Program for Installing and Configuring MTA
21.2	Installation Verification Facility
21.3	MTA Run in Local Mode
21.4	MTA Statistics
21.5	Modify MTA Options
21.6	Automatic Maintenance
21.7	Manual Maintenance
21.8	Utility Program for Administrative Tasks
21.9	On-line MTA Help Facility
21.10	Backup/Restore MHS Implementation
21.11	Restore MHS Implementation to Different Machine
21.12	Save Copy of All Outbound Messages on MTA

Remote MTA Connections	
22.1	Connect to Multiple Remote MTAs
22.2	Multiple Simultaneous MTA Connections
22.3	Modify Multiple Simultaneous MTA Connection Limit
22.4	MTA Relaying
22.5	Connection Options Set on Per MTA Basis
22.6	Maintain Connection for Period of Time after Message Transfer
22.7	Modify Period of Time Connection is Maintained after Message Transfer

Access Control	
23.1	Limit Access to Registered User Subset
23.2	Limit User to Only Sending or Receiving
23.3	Exclusive Access

Debug Capabilities	
24.1	Log File
24.2	Monitor MHS Implementation Status
24.3	Verify Routing of O/R Name to Another MTA
24.4	Queue Message for Retransmission
24.5	Access Message Queued for Retransmission
24.6	Translate Queued Message into English
24.7	Translate Queued Message into ASN.1 Format
24.8	Display Error on Console
24.9	Send MHS Message Detailing Error to System Administrator
24.10	MTA Test Facility

Underlying OSI Layers	
25.1	Session Services
25.2	Transport Class 0
25.3	Transport Class 4
25.4	CONS over X.25
25.5	CLNS over X.25
25.6	CLNS over LAN
25.7	X.25 Version
25.8	X.25 Prototype
25.9	Modify X.25 Options
25.10	Lower Layer Statistics
25.11	Optimization Facility
25.12	Troubleshooting Facility

Certification	
26.1	Passed Conformance Test Procedure
26.2	Passed Interoperability Test Procedure

Hardware Requirements	
27.1	Number of Machines
27.2	CPU
27.3	Disk Space
27.4	Memory
27.5	External Devices

Software Requirements	
28.1	Number of Software Components
28.2	Underlying OSI Software Installation
28.3	Operating System and Version
28.4	Proprietary Mail System Installation

Documentation	
29.1	Guide for Installing
29.2	Guide for Using
29.3	Guide for Administrating
29.4	Guide for Troubleshooting
29.5	MHS Reference Guide
29.6	Guide for Quick Reference

Pragmatic Constraints	
30.1	UA Pragmatic Constraints
30.2	MTA Pragmatic Constraints

APPENDIX C: Tables of Experiments

This appendix contains a listing of the MHS experiments described in these guidelines. The experiments are presented here in tabular form. Each table consists of a title, which corresponds to an experiment described in section 3.4.1, a purpose, and a list of experiment inputs and outputs.

Experiment #1
Purpose: To measure the optimum message transfer time, and to obtain a base measurement against which message transfer times of other experiments can be compared.
Inputs: 1. The experiment environment. 2. The size of the messages to be transferred.
Outputs: 1. The message number for each message transferred. 2. The submission time for each message transferred. 3. The delivery time for each message transferred. 4. The transfer time for each message transferred. 5. The minimum message transfer time. 6. The maximum message transfer time. 7. The average message transfer time.

Experiment #2
Purpose: To measure the effect of estimated mail system usage on message transfer time.
Inputs: 1. The experiment environment. 2. The size of the messages to be transferred. 3. Two message submission time tables.
Outputs: 1. The message number for each message transferred. 2. The submission time for each message transferred. 3. The delivery time for each message transferred. 4. The transfer time for each message transferred. 5. The minimum message transfer time. 6. The maximum message transfer time. 7. The average message transfer time. 8. The difference between the average message transfer times of Experiment #2 and Experiment #1.

Experiment #3
Purpose: To measure the effect of an MTA relay on message transfer time.
Inputs: 1. The experiment environment. 2. The size of the messages to be transferred. 3. The network types between the MTAs.
Outputs: 1. The message number for each message transferred. 2. The submission time for each message transferred. 3. The delivery time for each message transferred. 4. The transfer time for each message transferred. 5. The minimum message transfer time. 6. The maximum message transfer time. 7. The average message transfer time. 8. The difference between the average message transfer times of Experiment #3 and Experiment #1.

Experiment #4
Purpose: To measure the effect of an additional message recipient on message transfer time.
Inputs: The experiment environment. The size of the messages to be transferred.
Outputs: 1. The message number for each message transferred. 2. The submission time for each message transferred. 3. The delivery time for each message transferred. 4. The transfer time for each message transferred. 5. The minimum message transfer time. 6. The maximum message transfer time. 7. The average message transfer time. 8. The difference between the average message transfer times of Experiment #4 and Experiment #1.

Experiment #5
Purpose: To measure the effect of message size on message transfer time.
Inputs: 1. The experiment environment. 2. The size of the messages to be transferred.
Outputs: 1. The message number for each message transferred. 2. The submission time for each message transferred. 3. The delivery time for each message transferred. 4. The transfer time for each message transferred. 5. The minimum message transfer time. 6. The maximum message transfer time. 7. The average message transfer time. 8 The difference between the average message transfer times of Experiment #5 and Experiment #1.

Experiment #6
Purpose: To measure message transfer capacity.
Inputs: 1. The experiment environment. 2. The size of the messages to be transferred. 3. The number of messages to be transferred.
Outputs: 1. The number of messages transferred. 2. The submission time for the first message transferred. 3. The delivery time for the last message transferred. 4. The transfer time for all the messages.

Experiment #7
Purpose: To measure CPU and I/O utilization.
Inputs: 1. The experiment environment. 2. The size of the messages to be transferred. 3. The number of messages to be transferred. 4. The period of time during which the messages are to be transferred.
Outputs: 1. The number of messages transferred. 2. The period of time during which the messages were to be transferred. 3. The percentage of CPU utilization. 4. The percentage of I/O utilization.

APPENDIX D: Abbreviations

This appendix defines the abbreviations used in this document.

ADMD	Administration Management Domain
AE	Application Entity
APDU	Application Protocol Data Unit
ASN.1	Abstract Syntax Notation 1
BCC	Blind Courtesy Copy
C	Conditional
CALS	Computer-Aided Acquisition and Logistics Support
CC	Courtesy Copy
CCITT	Consultative Committee on International Telephone and Telegraph
CONS	Connection Oriented Network Service
CLNS	Connectionless Network Layer Service
FIPS	Federal Information Processing Standard
GOSIP	Government Open Systems Interconnection Profile
IA5	International Alphabet No. 5
ID	Identity
IM-UAPDU	IP-message User Agent Protocol Data Unit
IP	Interpersonal
IPM	Interpersonal Messaging
IPMS	Interpersonal Messaging System
ISO	International Organization for Standardization
LAN	Local Area Network
M	Mandatory
MD	Management Domain
MH	Message Handling
MHS	Message Handling System
MPDU	Message Protocol Data Unit
MT	Message Transfer
MTA	Message Transfer Agent
MTAE	Message Transfer Agent Entity
MTL	Message Transfer Layer
MTS	Message Transfer System
NIST	National Institute of Standards and Technology
NDN	Non-delivery Notification
NSAP	Network Service Access Point
NSDU	Network Service Data Unit
OPDU	Operation Protocol Data Unit
O/R	Originator/Recipient
OSI	Open Systems Interconnection
POSIX	Portable Operating System Interface for Computer Environments
PRMD	Private Management Domain
PTT	Postal Telephone and Telegraph

P1	Message Transfer Protocol
P2	Interpersonal Messaging Protocol
P3	Submission and Delivery Protocol
RTS	Reliable Transfer Server
SSAP	Session Service Access Point
SDE	Submission and Delivery Entity
SSDU	Session Service Data Unit
SFD	Simple Formatable Document
SMPDU	Service Message Protocol Data Unit
SR-UAPDU	Status Report User Agent Protocol Data Unit
TIF0	Text Interchange Format 0
TIF1	Text Interchange Format 1
TPDU	Transport Protocol Data Unit
TSAP	Transport Service Access Point
TSDU	Transport Service Data Unit
TTX	Teletex
UA	User Agent
UAE	User Agent Entity
UAL	User Agent Layer
UAPDU	User Agent Protocol Data Unit
UTC	Coordinated Universal Time
WAN	Wide Area Network

APPENDIX E: Glossary

This appendix provides a glossary of MHS terms.

Administration management domain (ADMD) – A management domain managed by an Administration.

Base attribute set – A minimum set of attributes whose values unambiguously identify a particular management domain.

Body – The body of the IP-message is the information the user wishes to communicate.

content – The piece of information that the originating UA wishes delivered to the recipient UA. For IPM UAs, the content consists of either an IP-message or an IPM-status-report.

cooperating user agent – A UA that cooperates with another recipient's UA in order to facilitate the communication between the originator and recipient.

delivery – The interaction by which the Message Transfer Agent transfers to a recipient User Agent the content of a message plus the delivery envelope.

delivery envelope – The envelope which contains the information related to the delivery of the message.

descriptive name – A name that denotes exactly one user in the MHS.

encoded information type – It is the code and format of information that appears in the body of an IP-message. Example of encoded information types are Teletex, TIF0 (Group 4 facsimile) and voice.

envelope – A place in which the information to be used in the submission, delivery and relaying of a message is contained.

heading – The Heading of an IP-message is the control information that characterizes the IP-message.

interpersonal messaging (IPM) – Communication between persons by exchanging messages.

interpersonal messaging service – The collection of UAs and MTAs, which provide the Interpersonal Messaging Service.

IP-message – An IP-message carries information generated by and transferred between IPM UAs. An IP-message contains a Heading and a Body.

IPM-status-report – The pieces of information generated by an IPM UAE and transferred to another IPM UAE, either to be used by that UAE or to be conveyed to a user.

management domain(MD) – The set of MHS entities managed by an Administration or organization that includes at least one MTA.

message – In the context of Message Handling Systems, the unit of information transferred by the MTS. It consists of an envelope and a content.

message handling address – An O/R address which is comprised of an Administration Management Domain name, a country name and a set of user attributes.

message handling system(MHS) – The set of UAs plus the MTS.

message transfer agent(MTA) – The functional component that, together with other MTAs, constitutes the MTS. The MTAs provide message transfer service elements by: (1) interacting with originating UAs via the submission dialogue. (2) relaying messages to other MTAs based upon recipient designations, and (3) interacting with recipient UAs via the delivery dialogue.

message transfer agent entity(MTAE) – An entity, located in an MTA, that is responsible for controlling the MTL. It controls the operation of the protocol to other peer entities in the MTL.

message transfer layer (MTL) – A layer in the Application Layer that provided MTS service elements. These services are provided by means of the services of the layer below plus the functionality of the entities in the layer, namely, the MTAEs and the SDEs. message transfer protocol(P1) - The protocol which defines the relaying of messages between MTAs and other interactions necessary to provide MTL services.

message transfer service – The set of optional service elements provided by the Message Transfer System.

message transfer system(MTS) – The collection of MTAs, which provide the Message Transfer Service elements.

open systems interconnection (OSI) – A term referring to the capability of interconnecting different systems.

O/R address – A descriptive name for a UA that contains certain characteristics which help the MTS to locate the UA's point of attachment. An O/R address is a type of O/R name.

originating UA – A UA that submits to the MTS a message to be transferred.

originator – A user, a human being or computer process, from whom the MHS accepts a message.

primitive name – A name assigned by a naming authority. Primitive names are components of descriptive name.

private management domain (PRMD) – A management domain managed by a company or non-commercial organization.

recipient – A user, a human being or computer process, who receives a message from the MHS.

recipient UA – A UA to which a message is delivered or that is specified for delivery.

relaying – The interaction by which one Message Transfer Agent transfers to another the content of a message plus the relaying envelope.

relaying envelope – The envelope which contains the information related to the operation of the MTS plus the service elements requested by the originating UA.

submission – The interaction by which an originating User Agent transfers to a Message Transfer Agent the contents of a message plus the submission envelope.

submission and delivery entity (SDE) – An entity that is located in the MTL, co-resident with the UA and not with an MTA, and responsible for controlling the submission and delivery interactions with an MTAE.

submission and delivery protocol(P3) – The protocol which govern communication between an SDE and a MTAE.

submission envelope – The envelope which contains the information the MTS requires to provide the requested service elements.

user – A person or computer application or process who make use of MHS

user agent (UA) – Typically, a set of computer processes (for example, an editor, a file system, a word processor) that are used to create, inspect, and manage the storage of messages. There is typically one user per UA. During message preparation, the originator communicates with his UA via an input/output(I/O) device (for example, a keyboard, display, printer, facsimile machine, and/or telephone). Also by means of these devices, the UA communicates to its user message received from the MTS. To send and receive messages, the UA interacts with the MTS via the submission and delivery protocol.

user agent entity (UAE) – An entity in the UAL of the Application Layer that controls the controls the protocol associated with cooperating UAL services. It exchanges control information with the MTAE or SDE in the layer below. The control information is the information the MTL requires to create the appropriate envelope and thus provide the desired message transfer service elements.

user agent layer (UAL) – The layer that contains the UAEs.

REFERENCES

The documents referenced in this project include:

[1] CCITT 1984 X.400, Message handling systems: system model-service elements.

[2] CCITT 1984 X.401, Message handling systems: basic service elements and optional user facilities.

[3] CCITT 1984 X.408, Message handling systems: encoded information type conversion rules.

[4] CCITT 1984 X.409, Message handling systems: presentation transfer syntax and notation.

[5] CCITT 1984 X.410, Message handling systems: remote operations and reliable transfer server.

[6] CCITT 1984 X.411, Message handling systems: message transfer layer.

[7] CCITT 1984 X.420, Message handling systems: interpersonal messaging user agent layer.

[8] Stable Implementation Agreements for Open Systems Interconnection Protocols Version 3 Edition 1, NIST Spec. Publ. 500-177; 1989 December.

[9] Boland, T., Government Open Systems Interconnection Profile Users' Guide, NIST Spec. Publ. 500-163; 1989 August.

[10] Government Open Systems Interconnection Profile (GOSIP), Version 1. FIPS 146.

[11] COS Stack Specification COS/AMH113.

[12] Digital, Introduction to MAILbus.

[13] Digital, Introduction to Message Router.

[14] Digital, Message Router Installation Guide.

[15] Digital, Message Router Configuration Guide.

[16] Digital, Message Router Management Guide.

[17] Digital, Message Router Management Reference Manual.

[18] Digital, Message Router Management Action Procedures.

[19] Digital, Message Route X.400 Gateway Installation Guide.

[20] Digital, Message Route X.400 Gateway Management Guide.

[21] Digital, Message Route X.400 Gateway User's Guide.

[22] Digital, VMS X400 Mail User's Guide (CLI).

[23] Hewlett Packard, HP OfficeConnect to X.400/V, Administrator's Guide.

[24] Hewlett Packard, Elm Users Guide.

[25] Hewlett Packard, Elm Reference Guide.

[26] Hewlett Packard, Elm Configuration Guide.

[27] Hewlett Packard, Elm Alias Users Guide.

[28] Hewlett Packard, Elm Filter Guide.

[29] Hewlett Packard, Elm Forms Mode Guide.

[30] Retix, RetixMail: Quick Reference.

[31] Retix, PC-320 WAN Coprocessor: Installation Guide.

[32] Retix, RetixMail/OpenServer 400: MH-410 Administrator's Reference.

[33] Retix, RetixMail/OpenServer 400: MH-410 Software Installation Guide.

[34] Retix, RetixMail: User's Reference.

[35] Xerox, Xerox ViewPoint: Mailing.

[36] Xerox, Xerox Network Services: Mail Service Technical Reference.

NIST-114A
(REV. 3-89)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NIST/SP-500/182

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
August 1990

4. TITLE AND SUBTITLE

Guidelines for the Evaluation of Message Handling Systems Implementations

5. AUTHOR(S)

Steve Trus, Curtis Royster, Paul Markovitz

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED
Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

U. S. Army Information Systems Command, Ft. Huachuca, AZ 85613-5300
U. S. Air Force Communications Command, Scott Air Force Base, IL 62225-6001
Internal Revenue Service, 1111 Constitution Ave., N.W., Wash., D.C. 20224

10. SUPPLEMENTARY NOTES

☐ DOCUMENT DESCRIBES A COMPUTER PROGRAM; SF-185, FIPS SOFTWARE SUMMARY, IS ATTACHED.

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

This document advances the goals of the Government Open Systems Interconnection Profile (GOSIP) by providing guidelines for evaluating Message Handling Systems (MHS) implementations. These guidelines can assist a user in the determination of which implementation, among several candidates, will best meet the functional and performance requirements of that user. Specifically, this document contains: (1) guidelines for evaluating the functional specifications of MHS implementations, (2) guidelines for measuring the performance of MHS implementations, and (3) guidelines for matching the functional and performance specifications of an MHS implementation to the functional and performance requirements of the user.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

electronic mail; functional evaluation; Government Open Systems Interconnection Profile; Message Handling Systems; Message Transfer Agent; Open Systems Interconnection; performance evaluation; user agent.

13. AVAILABILITY

- [X] UNLIMITED
- [ ] FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).
- [X] ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.
- [X] ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
128

15. PRICE

ELECTRONIC FORM

☆ U.S. GOVERNMENT PRINTING OFFICE 1990 -- 261-913 20661

ANNOUNCEMENT OF NEW PUBLICATIONS ON COMPUTER SYSTEMS TECHNOLOGY

Superintendent of Documents
Government Printing Office
Washington, DC 20402

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Institute of Standards and Technology Special Publication 500-.

Name ____

Company ____

Address ____

City ____________________ State ______________ Zip Code ____________

(Notification key N-503)

NIST Technical Publications

Periodical

Journal of Research of the National Institute of Standards and Technology—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published quarterly for NIST by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW., Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NIST research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NIST publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NIST Interagency Reports (NISTIR)—A special series of interim or final reports on work performed by NIST for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Institute of Standards and Technology
(formerly National Bureau of Standards)
Gaithersburg, MD 20899