Philippine Engineering Journal (1993) XIV (2): 75-84

MMDF ELECTRONIC MAIL

Efren F. Abaya, Ph.D.
J.J. Vergara Professor of Communications Systems
Department of Electrical & Electronics Engineering
College of Engineering
University of the Philippines
Diliman, Quezon City, Philippines

ABSTRACT

E-mail is a system for sending electronic messages through a computer network. This paper discusses the internal processes of a mail system in a Unix and TCP/IP environment, focusinng on hte Multi-channel Memorandum Distribution Facility (MMDF).

INTRODUCTION

Electronic mail, popularly called e-mail, is a system for sending "mail messages" in electronic form over a computer network. The mail message is composed electronically on a computer terminal, stored and forwarded between computers, and read at another terminals. Unlike fax or telex messages, e-mail can be completely paperless. Until recently, the "mail" was a plain ASCII text file. Newer mail systems can now accommodate other document forms, e.g., wordprocessor files, spreadsheets, graphics and program files. The transmission of mail takes the form of store-and-forward on a time scale of minutes, hours, or even days.

This paper discusses the internal processes of a mail system in a Unix and TCP/IP environment, focusing on the Multi-channel Memorandum Distribution Facility (MMDF).

FUNCTIONS OF AN E-MAIL SYSTEM

The general purpose of an e-mail system is to allow a user, whether it be a person or a running program, to send mail to another user (person or program) on the same network. To be able to accomplish this task, several basic functions are needed. These are:

1. The mail system must allow the user to compose or to edit a message, just like in ordinary word-processing. However, since e-mail has a structure (discussed later), the mail editor can provide assistance by providing prompts for required information such as addresses, or actively formatting the message to the required

structure. In addition, the mail editor can automatically add certain parts of the message, such as the date and time.

2. The mail system must electronically transfer the message from one computer to another using the network protocols and facilities provided (e.g., TCP/IP or uucp). This function should be transparent to the user, meaning that the mail system has to determine from the mail message itself such things as network addresses and host names. If one computer is connected to several computers, the mail system may have to deal with more than one transport protocol. Ideally, different forms such as text, graphics, and fax should be transportable.

3. The mail system should tell the sender what happened to the message, e.g., whether or not it was delivered; or the reason why it was not delivered. Ideally, such information should be given without any extra effort on the part of the receiver (i.e., he should not be obliged to compose an "I got your mail" reply and have to mail it).

4. The mail system should make it possible for the message to display properly on different kinds of computer terminals. In the case of Unix, this function is built into the operating system "termcap' and "terminfo" definitions (SCOb).

5. The mail system should display the message in a form that is convenient for the human reader. The simplest display format is to show the entire mail file, but this may be cluttered with unneeded headers. A better presentation is to separate the header from the body, and suppress unneeded headers.

6. The mail system should assist the recipient in handling the mail message. Manually initiated actions may include displaying, saving, filing, deleting or forwarding mail. Automatic actions by the mail system may include automatic acknowledgement upon receipt, automatic forwarding, or automatic filing.

7. The mail system should maintain mailboxes, which are directories or files in which the actual messages are stored. The mailbox must be readable by the recipient, but not by anyone else. To prevent spurious messages, no one except the mail system itself should be able to deposit a mail message in the mailbox.

8. The mail system should support the mapping of shorthand names (aliases) to long network addresses, as well as short names for groups of recipients (distribution list).

INTERNAL ARCHITECTURE

Physically, a mail system consists of computers interconnected by a network. Some computers are used directly by users to compose and to read mail messages. Other computers may only be used to store or to forward mail. Not all computers in the network need to have mail capability.

Following the CCIT model [TAN], a mail system logically consists of two types of software--a User Agent (UA) and a Message Transfer Agent (MTA).

The User Agent (UA) is the user's interface into the mail system, and has several functions. First, this software is used to compose or to edit a mail message. While an ordinary wordprocessor can be used for this task, a mail editor assists the user to attain the prescribed mail format by prompting for required information (e.g., addressee's name), or by automatically supplying information in the proper place (e.g., date). Second, a UA assists the user to read and manipulate received mail. The UA retrieves mail from a mailbox, and presents it in a convenient form. The UA also responds to any user commands that act on the mail. For example, mail may be deleted, saved, filed, or forwarded. Third, the UA interacts with an MTA to send or to receive mail.

The MTA performs the actual transfer of mail, by identifying a suitable path or route from the originating machine to the recipient machine. This route may pass through MTAs in intermediate machines. One MTA hands off the mail message to the next MTA, which then becomes responsible for its onward delivery, and so on in store-and-forward fashion. Finally, an MTA maintains user mailboxes, which are simply files or directories where the mail messages are deposited

In Unix operating systems, a program called "mail" is the commonly supplied UA. For sending mail, the program allows composing, editing. and reviewing of a message as it is entered, using either a built-in mail editor, or the standard Unix editor "vi". for reading mail, the program accepts commands to read, save, delete, reply to, or forward a message. The "mail" program can also classify received mail into folders (directories), just as paper mail is filed for storage.

The MTA function is performed by either MMDF or "sendmail". The Multi-Channel Memorandum Distribution Facility (MMDF) is a group of programs that route mail locally (within the same machine) or to counterpart MTAs in other machines through an appropriate communications channel such as TCP/IP, uucp, or the like. "Sendmail" is another more common system of programs that performs similar functions [LYN].

The processes that occur in one machine under MMDF to send mail are shown in Fig. 1. It begins when a user creates a mail message using the "mail" UA. This mail message, existing as a Unix file (usually in directory /tmp), is passed to the "submit" and "deliver" programs, and thence to one of several "channel programs", that collectively perform the MTA functions.

Program "execmail" is an interface between the UA and the MTA that makes the proper invocation to "submit". It adds headers as necessary if they are missing.

Program "submit" is the MMDF mail queue manager, and the entry point to the MTA. It resolves the address of the destination by reading information from the "To:" header line of the mail message (or from parameters supplied when it is called) and constructing a full address that is sufficient for the communication protocols to route the message. It then queues the message in an appropriate outgoing channel. The program can determine from the mail address the next machine to forward to, and which network protocol to use. The program can accept multiple addresses, so that the same mail message is sent simultaneously to several addresses. The calling program (e.g. execmail) can elect to have the mail delivered immediately, or left to the scheduling of the "deliver" program. (For example, mail may be delivered periodically, say every 10 minutes.)

The mail queues reside in several subdirectories in /usr/spool/mmdf where only authorized programs can access them. Here, mail is kept temporarily until it can be delivered.

Program "deliver" is the MMDF mail delivery process. Contrary to its name, "deliver" does not actually deliver the mail. What it does is to periodically scan (or "sweep") the mail queues and transfer their content to one of several "channel programs". It is the channel program that handles the interface with the network (e.g., uucp or TCP/IP) to actually send the message out of a physical port. If a particular message cannot be delivered immediately, "deliver" will keep it in the queue for the next sweep. It will keep trying for a reasonable period of time before giving up. As it is doing so, "deliver" can provide feedback on the progress of mail delivery, or warnings (in the form of e-mail) if a message stays too long in the queue or cannot be delivered.

If the mail is addressed to a user on the same machine ("local"), it goes directly to the addressee's mailbox. The "local channel program" simply appends an incoming message to the file in /usr/spool/mail that serves as the mailbox, separated from other messages by some control characters such as CTL-A.

If the mail is addressed to a foreign machine, the appropriate "channel program" will initiate a session with the foreign machine. The "channel program" handles the actual communications between two machines passing mail, and understands the specific mail protocol in use. Examples of protocols are "smtp" [LYN], widely used on the Internet, and "uucp" commonly used for dial-up mail exchanges. The interaction follows a client-server model.

MMDF channel programs are executable files residing in /usr/mmdf/chans. For example, if the local and foreign machines are connected by a TCP/IP serial line, the sending 'smtp channel program" will start a TCP/IP session with the receiving machine, which will cause a counterpart channel program to start on the foreign machine. The TCP/IP protocols take care of exchanging packets between machines and correcting line errors, so the mail system simply sees a reliable virtual circuit between mail machines. The actual dialogue between the sending and receiving channel programs follows the SMTP (Simple Mail Transfer Protocol) protocol [COM, HEDR]

The process of receiving mail is shown in Fig. 2, and is very similar to the sending process. Incoming mail is received by a channel program (acting as the server), and passed to 'submit" which places it in an appropriate queue. From here, "deliver" takes over to either place the mail in a local mailbox, or send it onward to another machine.

The UA "mail" program can access the user's mailbox in directory /usr/spool/mail for reading or disposition. Privacy of the mail is determined by the read and write privileges of this mailbox file. Usually, only the owner should be able to read a mailbox.

RFC 822 MAIL FORMAT

An electronic mail message may be likened to a physical latter. The physical document is enclosed within an envelope that contains delivery information (e.g., addressee's name and address, sender's name and address, special instructions). The document itself may be in any format agreed upon the sender and addressee, but the envelope information has a standard format.

Electronic mail in most Unix systems follows the Internet RFC 822 standard [HEDR, LYN]. The envelope part of an e-mail message is a set of header lines at the very beginning of the message, terminated by a blank line. The body of the message follows immediately.

An RFC 822 mail header consists of header lines with the structure

keyword: value

The minimum header contains the following three lines in any order:

From: sender_mail_address
To: addressee_mail_address
Date: date-sent

The "To:" line may contain one or more addressees.

Other useful header lines include:

Reply-To: return_reply_mail_address
Cc: carbon_copy_mail_address
Bcc: blind_carbon_copy_mail_address
Subject: subject _description
Message-ID: message_identification_number

The mail UA ("mail" and also "execmail") assists the user to compose these headers by providing the keywords and prompting for the required information. On reading mail, the UA extracts some header information. For example, the "Subject:" lines of several messages may be presented together as a table of contents of the mailbox.

The mail MTA on the sending side can extract the addressee's mail address from the "To:" line (as well as the "Cc:" and "Bcc:" lines) for purposes of routing the mail. The MTA on the receiving side can extract the "Reply-To:" mail address to generate an automatic acknowledgment (something like a postal return card).

The body of the e-mail may be in any format, except that the message (headers and body) must be transmitted as net ASCII (ASCII code with lines delimited by carriage return/linefeed combination) with only printable characters (no control or graphics characters). Also, because the Unix operating system handles the null character peculiarly, an e-mail message must not have null characters.

It is possible to get around these restrictions by converting the body of mail into net ASCII before sending, and reversing the process at the receiving side. For example, SCO Unix provides uucp utilities "uuencode" and "uudecode". The first takes a source file and converts it into a form that contains only printable ASCII characters. The second does the reverse. However, the printable ASCII file is about 35% longer than the original.

The restriction to net ASCII is a serious limitation because of the current interest in multimedia mail. This is being addressed by newer mail standards, such as the Multipurpose Internet Mail Exchange (MIME) extension to RFC 822 (LYN).

SIMPLE MAIL TRANSFER PROTOCOL

On TCP/IP channels, the dialogue between a sending channel program and a receiving channel program is specified by the Internet RFC 821 standard, better known as SMTP (Simple Mail Transfer Protocol). After the sender initiates a connection, the dialogue goes is shown below (adapted from [HERD]). The first two columns are not part of the SMTP dialogue and are only placed for purposes of the explanation that follows.

```
1       RCV         220 RED.RUTGERS.EDU SMTP Service at
                            29 Jun 87  05:17:18 EDT
2       SEND        HELO topaz.rutgers.edu
3       RCV         250 RED.RUTGERS.EDU -
                            Hello, TOPAZ. RUTGERS. EDU

4       SEND        MAIL From: <hedrick@topaz.rutgers.edu>
5       RCV         250 MAIL accepted
6       SEND        RCPT To: <levy@red.rutgers.edu>
7       RCV         250 Recipient accepted

8       SEND        DATA
9       RCV         354 Start mail input; end with <CRLF>.<CRLF>
10      SEND        Date: Sat, 27 Jun 87 13:26:31 EDT
11      SEND        From: hedrick@topaz.rutgers.edu
12      SEND        To: levy@red.rutgers.edu
13      SEND        Subject: meeting
14      SEND
15      SEND        Let's get together Monday at 1 pm.
16      SEND
17      RCV         250 OK

18      SEND        QUIT
19      RCV         221 RED.RUTGERS.EDU Service closing
                            transmission channel
```

All commands and responses use net ASCII in readable text form, which makes debugging or monitoring easy for a person (not necessarily efficient fro the machine point of view). At the beginning, the machines identify themselves. Line 1 identifies the receiving machine as red.rutgers.edu (case is not significant). Line 2 identifies the sending machine as topaz.rutgers.edu. The sender (hedrick@topaz.rutgers.edu in line 4) and the recipient (levy@red.rutgers.edu in line 6) are then identified. If there is more than one recipient, the "RCPT To:" portion in line 6 is repeated. Finally, the message is sent, beginning with the headers (lines 10-13), followed by the body (line 15), and terminated by a line containing a single.

The mail addresses (which are not part of the SMTP specification) use the standard format

user@domain

where "user" is a mail user name, and "domain" is a dot separated list (e.g., red.rutgers.edu) specifying a particular machine on the network. The MMDF channel program uses tables (channel files) in directory /usr/mmdf/table to translate a mail address of an IP address (which looks like 100.21.5.30) for purposes of setting up a TCP/IP virtual circuit to another machine.

Other tables in the same directory define "aliases" which are short, easily remembered names equivalent to the full mail address. For example, a definition like

levy: levy@red.rutger.edu

allows mail addressed to "levy" to reach machine "red.rutgrs.edu".

The mmdf tables can also define distribution lists like

eedept: bong@ee.up.edu, mike, lma@cc.up.edu

which allows mail addressed simply to "eedept" to reach three different addresses.

Program "submit" performs the alias expansion into the so-called fully-qualified mail address.

CONCLUSION

We have discussed the internal processes of a mail system using MMDF in a Unix and TCP/IP environment (more particularly, in SCO Unix). We have seen how the processing provides the desired functionalities of an e-mail system. There are also limitations to the current mail systems, such as inadequate provision for non-text mail and mail security which are being addressed by current work.

REFERENCES

[COM] Douglas E. Comer (1991). Internetworking With TCP/IP, vol. 1, 2nd ed., Prentice-Hall.

[HEDR] Charles L. Hedrick (1987). "Introduction to the Internet Protocols," Computer Science Facilities Group, Rutgers University, July 3, 1987.

[LYN] Daniel C. Lynch and Marshall T. Rose, eds. (1993). Internet System Handbook, Addison-Wesley Publishing Co.

[SCOa] SCO Unix System Administrator's Guide (1992). The Santa Cruz Operation, Inc.

[SCOb] SCO Unix System Administrator's Reference (1992). The Santa Cruz Operation, Inc.

[TAN] Andrew S. Tanenbaum (1988). Computer Networks, 2nd ed., Prentice-Hall.

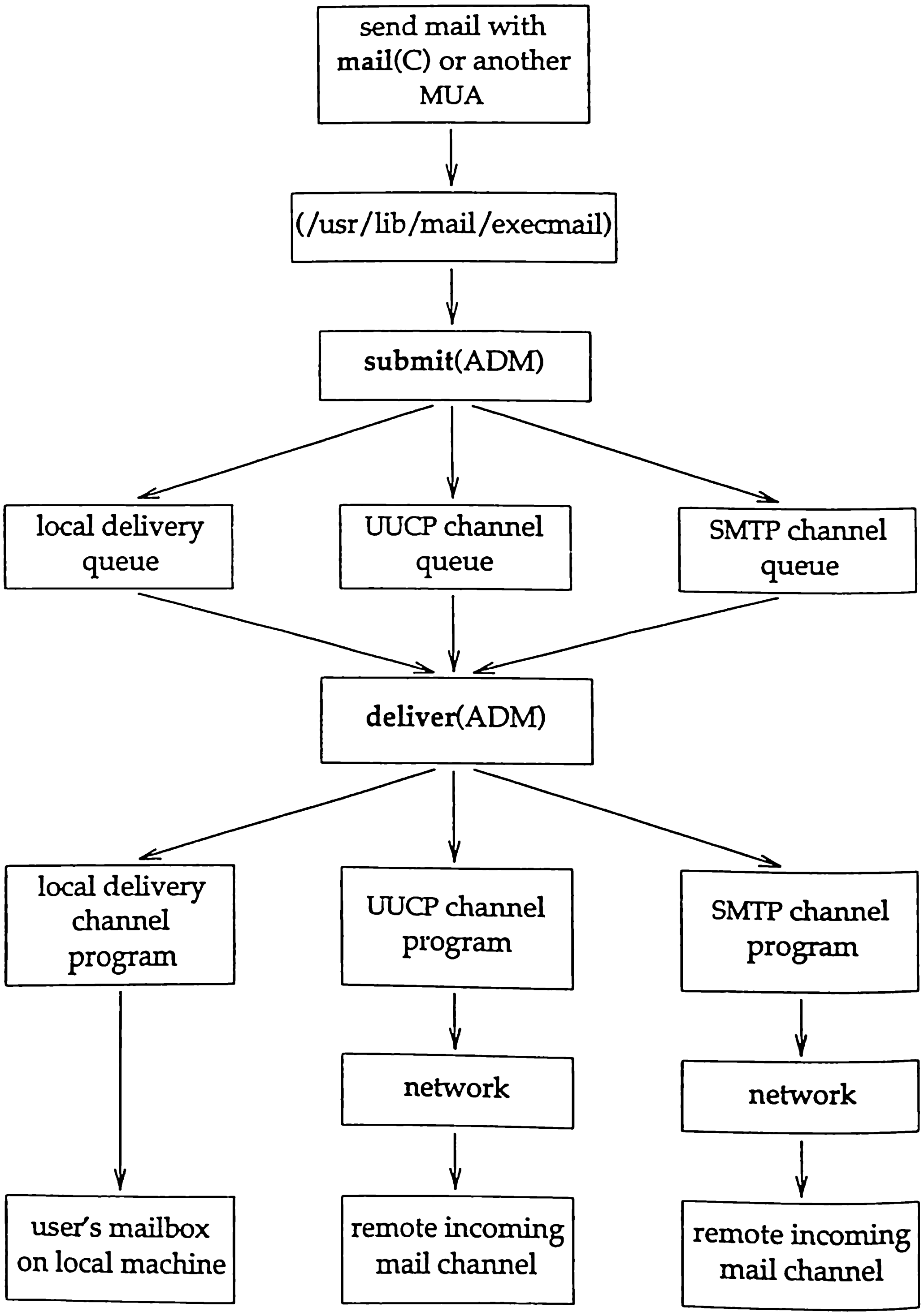

Figure 1. MMDF processes for outgoing e-mail. (From [SCOa])

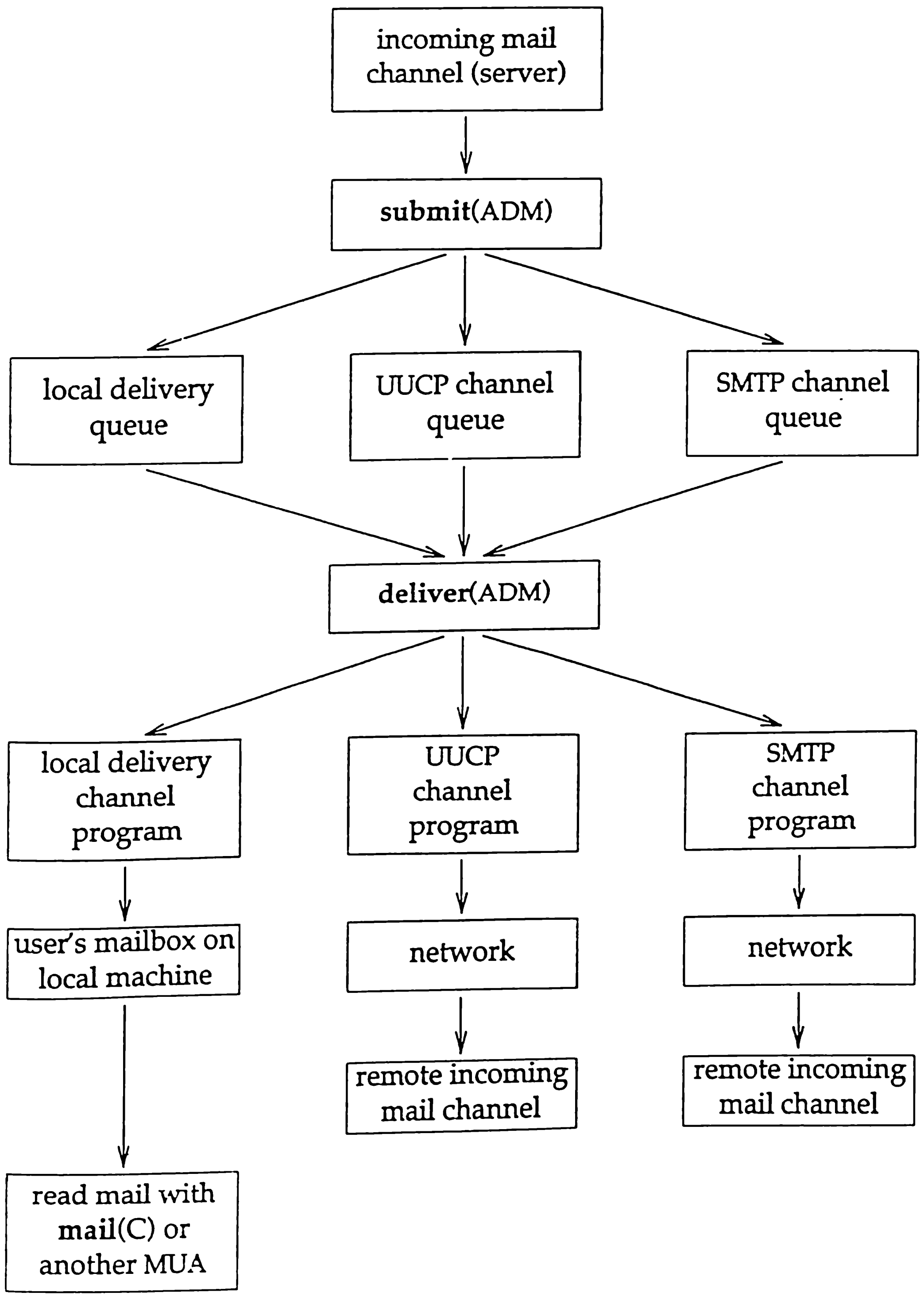
incoming mail channel (server)
submit(ADM)
local delivery queue
UUCP channel queue
SMTP channel queue
deliver(ADM)
local delivery channel program
UUCP channel program
SMTP channel program
user's mailbox on local machine
network
network
remote incoming mail channel
remote incoming mail channel
read mail with mail(C) or another MUA

Figure 2.MMDF processes for incoming e-mail. (From [SCOa])