

## DCNET Mail Plan

D.L. Mills and M.J. O'Connor  
COMSAT Laboratories  
23-Jul-81

## 1. Introduction

The Distributed Computer Network (DCNET) is an experimental research network in use at COMSAT and elsewhere. It includes a number of PDP11-compatible hosts connected to each other and to ARPANET, SATNET and other networks accessible to the DARPA Internet Project. The network and its hosts are used for research in computer network protocols and for general-purpose software development. One of the principal functions of the network is to support electronic mail, including the capability to construct and edit messages on-line, forward them to one or more hosts on DCNET, ARPANET or elsewhere and to retrieve and archive incoming messages from these hosts. These capabilities have, of course, been available and extensively used for some time on ARPANET hosts and on commercial services such as HERMES, ONTYME and TELEMAIL.

All DCNET hosts presently support both the Internet Protocol (IP) and Transmission Control Protocol (TCP), which have been implemented on many computers and operating systems and have been adopted as DoD standards [5]. High-level protocol modules allow DCNET hosts to connect to ARPANET hosts as virtual terminals and to perform mail functions using the ARPANET hosts in the ordinary way. In addition, files can be exchanged between DCNET and ARPANET hosts so that, in principal, messages arriving at ARPANET hosts can be relayed to DCNET hosts for further processing and archiving.

One of the tasks addressed in our present Internet Project activities is to investigate mechanisms with which mail functions can be performed directly in small hosts, rather than requiring support from larger ARPANET service hosts. Besides reducing the network resources required and providing potentially better performance, such mechanisms would greatly simplify integration of speech and facsimile media into the message system. We have been working to define and develop such mechanisms for some time now and have completed a version suitable for general use. We believe that this establishes the feasibility of performing nearly all mail functions in small hosts of the LSI-11 variety and yet maintain complete compatibility with existing hosts and their protocols. The remainder of this memorandum describes the architecture of this system and demonstrates its use.

## 2. DCNET Functions and Features

The DCNET was conceived as a prototype and testbed for distributed network architectures. All DCNET hosts use variants of a common operating system called the Basic Operating System (BOS), which includes the usual supervisor services together with the capability to emulate the DEC RT-11 operating system environment and run ordinary RT-11 system and application programs. Support for IP and TCP is embedded in the BOS together with an interface to application programs, including a set of high-level protocol modules supporting virtual-terminal (TELNET), file-transfer (FTP, NIFTP) and various utilities (XNET, PING, NAME and WATCH).

The most common application of a DCNET host is in single-user mode. Although the software can support simultaneous access by a number of users, the typical hardware configuration includes only a modest amount of on-line storage and can support only a limited set of applications in multi-user mode. In the case of those hosts equipped with dual floppy-disk drives, for example, the usual mode of operation is for each user to mount a personal disk containing private files on one of the drives and a system disk containing public files on the other.

All DCNET hosts participate in network functions such as routing, multiplexing, network monitoring, timekeeping and various utility "fake host" functions useful for testing with other internet hosts. Some hosts are given more specific responsibilities. For instance, two LSI-11 machines are presently used as multiplexors for a number of other machines and as gateways to ARPANET and SATNET. Another instance of specialization involves a host equipped with digital-facsimile and digital-speech peripherals which are used in experiments in multi-media message systems.

## 3. The DCNET Mail System

The most essential component in any electronic mail system is on-line storage. It has been common experience that rather a lot of it is required for even a modest number of users involved in an active research community such as the Internet Project. To be useful, a modest amount of this storage should be reachable from other internet hosts at all times, since those hosts holding unsent mail typically attempt to forward it immediately upon receipt. We are currently using disks with a capacity of between 10 and 20 megabytes for this purpose and believe this sufficient for the volume of mail expected, as well as for a general DCNET data and archive base. One of the disks is attached to a DCNET host expected to be available for mail access substantially all the time; however, this host may

occasionally be unavailable for short periods due to program development. For subsequent reference this host will be called the mail host.

The mail host serves as a DCNET post office and forwarding depot, but is not ordinarily used for general-purpose application programs. The remaining hosts are used for these programs by various individuals on an intermittent basis. In the typical scenario, a user mounts his personal disk on one of the local hosts, contacts the mail host and interrogates its data base for his new mail. Upon inspection of the mail, the user disposes of it in one of several ways, including: (1) deletes it, in which case it is gone forever; (2) forwards it to the local host for later processing; (3) copies it to a file or printer at the mail host, local host or some other host. Implicit in this scenario is the expectation that the volume of mail will require that each user individually archive his mail on his cache of personal disks as required. Also implicit is the requirement that some forms of mail, in particular multi-media speech and facsimile, will require access to a host with the required peripheral equipment. We expect eventually to provide automatic forwarding features that do not require direct interrogation of the mail host.

In order to send mail, the user constructs and edits each message at the local host, perhaps incorporating messages and files from other hosts including the mail host. Once the message has been prepared and prefixed with a list of recipients in a standard header, the user initiates transmission in one of two ways: (1) opens connections with each recipient internet host and transmits the message directly or (2) forwards the message to the mail host for later onward relay. The user would naturally elect the former if speed was important and the latter if the recipient host was not responding at that time.

#### 4. Mail Protocols

The mechanisms designed and implemented in DCNET hosts to support the above scenarios must be compatible with those used elsewhere in the internet community. The existing ARPANET mail system evolved as a feature of the File Transfer Protocol (FTP) used to transport files between ARPANET hosts. The original FTP was described in a working document called RFC-542 and the format of the mail message itself in RFC-733 [1]. The FTP described in RFC-542 is, however, not compatible with the present internet protocols and therefore is unsuitable for use outside the ARPANET. A version of FTP compatible with TCP has been proposed in RFC-765 [2]; however, there are few servers operational at present which conform to this protocol.

In order to provide mail support for systems using TCP and for interworking with the existing ARPANET systems, a new protocol called the Mail Transfer Protocol (MTP) [3] was developed and described in RFC-780. The MTP is designed to operate with current internet protocols and, in addition, to provide for onward relay of mail into networks that do not conform to these protocols, such as MMDF and NITS [7]. Preliminary versions of MTP have been implemented at ISI for TOPS-20 hosts, at MIT for Multics hosts, at DCEC for Unix hosts and at COMSAT for DCNET hosts.

The MTP is regarded as an intermediate step between the ARPANET-specific FTP-based mail and a proposed new system called the Internet Message Protocol (IMP) [4], which provides much greater operational flexibility together with the capability to process multi-media data types. The MTP can provide that now only through ad-hoc protocol extensions. However, it is likely that the MTP will be used for some time until the necessary software can be constructed for the ARPANET hosts. The IMP, which is described in RFC-759, is now being implemented by ISI for TOPS-20 hosts and is being planned for DCNET hosts.

In order to evaluate these protocols and assess their feasibility for use in the DCNET environment, we have constructed protocol modules to support MTP and have tested them with other hosts on the ARPANET, including those at ISI, DCEC and MIT. Although yet to be thoroughly debugged, our initial experience indicates this to be a practical way to join the ARPANET mail community. We intend this implementation to be an intermediate step; however, and expect to proceed to the IMP and multi-media data types in the future.

#### 5. Data Structures

In the RFC-733 model messages are sent by a user to a specified recipient in the format <user>@<host>, where <host> is the name of a host and <user> is the name of an user known to that host. The implied address, usually called a mailbox, is typically associated with a mail file belonging to the recipient and is easily generalized to include organizations as well as users and networks as well as hosts. As messages arrive at a host the mail server dispatches them to the various mailboxes by appending them, together with an appropriate header, following the messages already in the mailbox.

Since most of our interactions with internet hosts have been with TOPS-20 machines, we have tried to maintain a degree of compatibility with the mail file formats used by that system. The file is line-structured, with each line terminated by the ASCII sequence <CR><LF>, and contains only ASCII printing characters and format effectors. Messages

consist of a file header, which contains a character count, followed by the message text. Messages are stored one after the other with the last followed by an ASCII <SUB> character for compatibility with other RT-11 components. Figure 1 shows the format of a typical message.

```
29-May-81 01:01:14,342;000000000000
MRCP to:<OConnor@ISIE>
MRCP to:<@Multics,Mills@ISIE>
MRCP to:<Mills@COMSAT>
MAIL from:<Mills@COMSAT-DLM>
Date: 29-May-81 01:00:42-UT
From: Mills@COMSAT-DLM
Subject: Mail example
To: OConnor@ISIE
cc: <@Multics,Mills@ISIE>,Mills@COMSAT
```

Folks,

This message demonstrates RFC-733 and RFC-780 formats.

Regards,  
Dave

-----

The first line is the file header, including the date, time and count of characters in the message text, and followed by an array of twelve flag characters used by the MSG program described below. The message text begins immediately following the <CR><LF> which terminates this line and includes first the transport (RFC-780) header followed by the message (RFC-733) header and, finally, the body of the message itself. In the above example the lines beginning with MRCP and MAIL belong to the transport header and the lines following that up until the blank line belong to the message header.

Mail files can be created in three ways: (1) by copying a mail file from a TOPS-20 or DCNET host as-is to a DCNET host, (2) by creating and appending a new message locally and (3) by receiving and appending a message from another host. Case (1) has been found useful during testing and in cases involving large amounts of mail which can be bulk-transferred using more efficient file-transfer protocols like FTP and NIFTP. However, TOPS-20 files do not include the transport header, so that messages sent from these files require manual intervention. Case (2) is implemented by an interactive program called SNDMSG, which operates much like the TOPS-20 program of the same name to construct and edit messages and append them, along with their transport and message headers, onto a specified mail file. Case (3) is implemented by the MTPSRV server program, which listens for messages from the network and appends them

onto a well-known mail file. All three cases produce identical file formats, so that a common message display program can be used. This interactive program, called MSG, is the focus of most mail operations and is used in the same fashion as its TOPS-20 counterpart of the same name.

## 6. Mail Operations

Display and editing of mail file contents is performed by the MSG program. This program includes the capability to select messages and groups of messages and to display them on the operator's terminal and/or append them to other mail files. Since DCNET files and devices can be accessed in the same way, this amounts to the capability to print them and even to display them on the Dacom facsimile machine or speak them on the LPCM speech decoder (assuming the correct source encoding, of course). When a message is copied to a file its headers are copied as well, so that copying a message to a mail file results in a mail file in correct format.

A message is constructed using the SNDMSG program, which builds the transport and message headers shown in Figure 1 according to an interactive dialog and then edits the text as specified by the user. Note that, as illustrated in Figure 1, separate MRCP lines are created for each recipient and a MAIL line is created for the sender. These lines are edited by the MTP user program to indicate the delivery status for each recipient. Features in the present SNDMSG implementation provide for the inclusion of data files, such as might be produced by a standard text editor, and address lists.

Mail is sent to recipients at the various internet hosts by the MTP user program. This program first searches a specified mail file and constructs a data structure including a set of pointers to the MRCP transport-header lines, along with the internet address associated with each recipient host. It then sorts this structure by host and constructs a source-route string, if necessary, as specified by the operator. Finally, it connects to each host in turn and sends its messages in either text-first or recipients-first format, as required by the host. As delivery to each recipient is confirmed, the MTP user program overwrites the corresponding MRCP string with the string SENT. Other strings are possible in case of errors.

It may happen that some messages sent to a host may specify recipients not at that host, in which case these messages must be forwarded to the final destination as required by RFC-780. This would be the case when an operator at a local host wishes to stage a batch of messages at the mail host for later relay to other hosts not on-line at the moment. In addition, forwarding is also required

when the final destination host supports some transport protocol other than TCP, so that an intermediary supporting both protocols is required. The present system supports two operational modes, one in which mail is sent automatically either directly to the destination or via an intermediate relay, as directed by internal tables, and the other in which it is sent manually according to a source route specified by the operator.

Mail is ordinarily received automatically at a DCNET host by the MTPSRV server program. This program appends each message as it is received to a public, controlled-access mail file called UNSENT.MSG. For those messages addressed to a recipient at the receiving host, the corresponding MRCP string is overwritten with the string DLVD; the remainder are left for later relay by the MTP user program.

#### 8. On Hosts, Users and Mailboxes

Upon reflection on the above it may be noted that no mention is made of the concept of a DCNET mailbox. This is intentional and reflects the fact that each user is identified in fact only by his personal data disk and an informal convention of assigned user names. Matters become considerably more complicated when DCNET virtual hosts are involved, for this mechanism (described in detail elsewhere) provides the capability to associate one or more internet addresses with a single physical host and to move this association from time to time. Thus, the mail host may pop up in various physical hosts depending upon any of several considerations; however, the internet addressing will be transparent to this and the routing will be performed automatically.

For these reasons we have chosen to identify a particular internet address with the mail host, to be called simply COMSAT and the remaining hosts as COMSAT-xxx where xxx corresponds to the number (or name) of each of the other virtual hosts. Ordinarily, mail for all DCNET users is sent to mailboxes such as <user>@COMSAT. It would then remain at the mail host for later inspection by <user>. If, on the other hand, it is known that <user> is active on some DCNET host at the time the mail is to be sent, then it could be sent directly to that host.

In order to preserve privacy when accessing messages at the mail host via virtual-terminal connection from a local host, a feature has been incorporated into the MSG program normally used for this purpose. Ordinarily, all messages received at the mail host are saved in a public file called UNSENT.MSG, while the messages belonging to each user are saved in private files. MSG normally operates with a

private file as specified by the user, with access granted to UNSENT.MSG only by means of a keyword, normally the recipient's name. A special MSG command provides for searching UNSENT.MSG for messages which have been "delivered" (marked DLVD) to the recipient name matching the keyword. These messages can then be appended to the user's private file and forwarded to his local host for further processing or archiving, if required.

#### 9. Internet Name Domains

In the long run, it will not be practicable for every internet host to include all internet hosts in its name-address tables. Even now, with over four hundred names and nicknames in the combined ARPANET-DCNET tables, this has become awkward. Some sort of hierarchical name-space partitioning can easily be devised to deal with this problem; however, it has been wickedly difficult to find one compatible with the known mail systems throughout the community. The one described here, which has been partially implemented in the DCNET mail system, is the product of several discussions and meetings and is believed both compatible with existing systems and extensible for future systems involving thousands of hosts.

We first observe that every internet host is uniquely identified by one (or more) 32-bit internet addresses and that the entire system is fully connected. For the moment, the issue of protocol compatibility will be ignored, so that all hosts can be assumed MTP-competent. We next impose a topological covering on the space of all internet addresses with a set of so-called name domains. In the natural model, name domains would correspond to institutions such as ARPA, USC and COMSAT, and would not be necessarily disjoint or complete. While in principle name domains could be hierarchically structured, we will assume in the following only a single-level structure.

Every name domain is associated with one (or more) internet hosts called mail forwarders and the name of that domain is a name or nickname for any of these hosts. Each mail forwarder is expected to maintain name-address tables containing all other hosts in its domain and, in addition, at least one mail forwarder for every other domain. All hosts other than mail forwarders are expected to maintain name-address tables containing at least one mail forwarder for every domain together with additional hosts as convenient. Following current practice, several nicknames may be associated with the principal name of a host in any domain and the names and nicknames need not be unique in any other domain. Furthermore, hosts can be multi-homed, that is, respond to more than one address. For the purpose of mail forwarding and delivery, we will assume that any of



these addresses can be used without prejudice.

In its most general form, an internet mailbox name has the syntax

`<user>.<host>@<domain>`

where `<user>` is the name of a user known at the host `<host>` in the name domain `<domain>`. This syntax is intended to suggest a three-level hierarchically structured name (reading from the right) which is unique throughout the internet system. However, hosts within a single domain may agree to adopt another structure, as long as it does not conflict with the above syntax and as long as the forwarders for that domain are prepared to make the requisite transformations. For instance, let the name of a domain including DCNET be COMSAT and the name of one of its hosts be COMSAT-DLM with Mills a user known to that host. From within the COMSAT domain the name Mills@COMSAT-DLM uniquely identifies that mailbox as could, for example, the name Mills.DLM@COMSAT from anywhere in the internet system. However, Mills@COMSAT-DLM is not necessarily meaningful anywhere outside the COMSAT domain (but it could be).

The functions required of the forwarder are straightforward. When it receives a message for `<user>.<host>@<domain>`, it transforms `<host>` as necessary and forwards the message to its address found in the name-address table for `<domain>`. Note that a single host can be a mail forwarder for several independent domains in this model and that these domains can intersect. Thus, the names Mills@USC-ISIE, Mills.USC-ISIE@ARPA and Mills.USC-ISIE@COMSAT can all refer to the same mailbox and can, conceivably, all be entries in the same name-address table of some host. In this example the ARPANET host USC-ISIE appears as a domain with a null host name. Such use would be permissible only in case the name USC-ISIE was unique and known to all forwarders in the internet system.

In order for this scheme to work properly, it is necessary that messages transiting forwarders always contain complete internet mailbox names. When this is not feasible, as in the current ARPANET mail system, the forwarder must be able to determine which domain the message came from and edit the names accordingly. This would be necessary in order to compose a reply to the message in any case.

#### 10. Current Status and Unresolved Issues

The present system is working with DCNET hosts and certain other ARPANET hosts mentioned above, although some minor problems remain to be worked out. The experience gained has guided the implementation of features recently

incorporated into MSG and SNDMSG. Additional features are to be incorporated into MTP and MTPSRV as the result of current discussions and revisions of RFC-780.

There are a number of system-specific issues which need to be resolved before the DCNET implementation could be considered user-fortified, among them the following:

1. There are no provisions to resolve conflicting accesses to public files such as UNSENT.MSG which might occur if a message arrives while SNDMSG is active on the same file.
2. There are no provisions to compact UNSENT.MSG automatically as messages are forwarded or processed by the recipients.
3. The MTP user program must be initiated manually, rather than automatically as the result of a timeout, for example.
4. Forwarder mailbox name transformations as described above are not supported.
5. A facility is needed to re-address misrouted messages and to return failure messages to the sender in case they cannot be forwarded.

#### 11. Summary

This memorandum has described the environment and features required of the DCNET mail system, as well as an interim plan for compatibility with other developing internet mail systems. The interim plan focuses on the Mail Transfer Protocol of RFC-780 and on the transition of the existing DCNET mail system to be compatible with it. We believe this to be a useful and reasonably easy thing to do and that it will both shake the bugs from existing internet mail transport systems as well as smooth the way for the eventual implementation of the Internet Message Protocol and multi-media data types.

## 12. References

1. Crocker, D., J. Vittal, K. Pogran, and D. Henderson. Standard for the Format of ARPA Network Text Messages. Request for Comments RFC-733, NIC 41952, November 1977. Also in: Feinler, E. and J. Postel, eds. ARPANET Protocol Handbook. NIC 7104, for the Defense Communications Agency by SRI International, Menlo Park, California, Revised January 1978.
2. Postel, J., ed. File Transfer Protocol. Request for Comments RFC-765, Internet Experiment Note IEN-149. USC Information Sciences Institute, Marina del Rey, California, 1980.
3. Postel, J., and S. Sluizer. Mail Transfer Protocol. Request for Comments RFC-780. USC Information Sciences Institute, Marina del Rey, California, May 1981.
4. Postel, J. Internet Message Protocol. Request for Comments RFC-759, Internet Experiment Note IEN-113. USC Information Sciences Institute, Marina del Rey, California, August 1980.
5. Postel, J., ed. DoD Standard Transmission Control Protocol. Request for Comments 761, Internet Experiment Note 129, NTIS ADA082609. USC Information Sciences Institute, Marina del Rey, California, January 1980. Also in: ACM Computer Communication Review 10, 4 (October 1980).
6. PSS/SG3. A Network Independent Transport Service. Study Group 3, The Post Office PSS Users Group, February 1980. Available from: DCPU, National Physical Laboratory, Teddington, UK.