

IEN 141  
INDRA Note 897  
11th April 1980

## Message System Issues

C. J. Bennett

ABSTRACT: This INDRA Note discusses the design choices for the message server system to be built at UCL. Particular issues considered include: the nature of the UK user community; the nature of the message service to be offered on the server; the message formats and transfer protocols to be used; addressing; interworking with the ARPANET community; and the design of the message management system on the message server.

## Table of Contents

1. Introduction.....	1
2. The User Community.....	1
3. Message Movement.....	2
3.1 Message Format.....	2
3.1.1 Message Format Staging.....	3
3.2 Message Protocol.....	3
3.3 Message Transport.....	4
3.3.1 FTP Staging.....	5
3.4 Addressing.....	5
3.5 Status Reporting.....	8
4. Message Server Design.....	8
4.1 User Interface.....	8
4.2 Message Management.....	10
5. Conclusions.....	12

## 1. Introduction

Electronic message services have historically been one of the most successful services to have developed from the use of packet switched computer networks. However, these facilities have not been available to users of United Kingdom research data networks in the past, and UK users who wished to send mail to remote sites were required to obtain mailboxes on remote machines in the United States, accessible via ARPANET. With the development of public networks, in particular IPSS and PSS, and in view of the UKPO's policy of requiring users to move to these networks, it is no longer economically feasible to continue this mode of usage.

For these reasons it is proposed that University College London will develop a message server system based on a PDP-11/35 running UNIX and accessible initially to users through the DARPA Catenet, and later through PSS. This server would allow users to exchange messages with other users on the same site, users of ARPANET mail systems, and eventually users of other UK and US message servers. The aim of this INDRA note is to identify the design constraints on this system and to suggest approaches that may be taken to meet them.

## 2. The User Community

Five major groups of users can be identified who can be expected to interact with such a service in the short term. These are:

- (i) Current users of the ARPANET mail system, especially UK users who have (until recently) had dialin access through the TIP. The message server would become the prime mail server for this group. US users of ARPANET systems must be able to send messages to this site. This group will require messages formatted according to the rules specified in RFC 733 (as modified by actual practice).
- (ii) Users of the DARPA Catenet, who will be using at least three formats for intersite mail: those of RFC 733; those of the Internet Mail Protocol as defined in IEN 85; and the private formats being developed by RSRE.
- (iii) Users who wish to exchange messages between the UCL server and other servers which may become available through PSS. This group will initially require only PSS access to the server

and will exchange messages locally, but in the longer term it can be anticipated that other mail servers will emerge on PSS.

(iv) Users who wish to exchange messages with US message servers available through Telenet and IPSS. In particular, such traffic may arise through the US EDUNET project.

(v) UCL users who will exchange messages through the UCL ring, and who will wish to exchange messages with users in one or more of the other three categories.

### 3. Message Movement

This section is concerned with the questions which affect the movement of messages between the message server and other message sites. Four major questions must be considered: choice of message format; choice of transport mechanism; mail protocol; and addressing.

#### 3.1 Message Format

The message format may be based on one of the following choices:

- (i) ARPANET Format (RFC 733)
- (ii) Internet Mail Format
- (iii) RSRE Mail Format
- (iv) Other format not currently in use amongst the user community, such as those that may arise through the work of IFIP TC6.5, or through Telenet and EDUNET.

Of these choices, only the first is feasible at present. It is that which is most widely used at the moment, as it provides the current ARPANET mail service, and the internal UCL Unix mail service, and it is intended that it shall be used for initial DARPA Catenet mail. The DARPA Internet Mail format is very experimental, and although it is expected to remain stable for the time being no experience has been gained with it. Much the same comment applies to the RSRE



system. The fourth choice involves either obtaining an existing commercial system such as COMET, or devising a new format from scratch. Both these possibilities would result in considerable delay, and a UCL home-brewed format would be unlikely to be any more satisfactory, and would be much less acceptable to the users, than other alternatives.

As it may be anticipated that the server will have to interwork eventually with other formats, notably that of RSRE and whatever emerges amongst the EDUNET group, the development of other formats should be closely tracked. It is expected that conversion will eventually take place through the use of a common Internet Format such as that being developed in the DARPA Internet scheme.

### 3.1.1 Message Format Staging

One result of this is that users who will eventually require a different format for messages for their own server - initially, RSRE in particular - will require a conversion between the two. It is expected that this will take place at the UCL message server. As noted above, it is to be hoped that conversions will take place through a common intermediary format.

An important longterm question in this regard is how widely the UCL message server system will be distributed in the UK. If other message servers are built along the same lines, then the format chosen will become a UK standard, at least among the UK research community.

### 3.2 Message Protocol

The current ARPANET message protocol is essentially a trivial extension to the ARPANET file transfer, obtained through the MAIL option. This causes each message to be sent as a separate file to be appended to the message file of an individual user at that site. Given future use of IPSS and PSS this is an uneconomic option. There are two reasons for this.

- (i) Demultiplexing for a message which is to be copied to several users at the same site occurs at the sender, not the receiver. Thus a message for N users at site X is transferred N times, even though it is identical. If mailers

were capable of parsing the message headers properly, the message need only be sent once.

- (ii) For each message transferred a separate data connection is set up. Thus a queue of  $N$  messages for  $M$  sites ( $M < N$ ) will require  $N + M$  calls to be made. If the messages were mailbagged by site, only  $2M$  calls need be made. (Note that if FTP control and data were mixed on the same call, as in the NIFTP (see below), these figures reduce to  $N$  and  $M$  respectively).

Both these changes have some impact on message format. The first requires, as a minimum, that all recipients of a message at a given site be visible in the To: and Cc: fields - that is, it is not possible if the mailing list facility is used in its current form. In such cases, the sender must provide the list, and the receiver must recognise that this list should be suppressed or separated from the users' copies. It is to be hoped that the Internet group will accept this proposal as a minimum change to be made for use in the Catenet, and that similar procedures will be set up by other groups.

Mailbagging requires that different messages in a file transferred must be clearly delimited. This requires a mailbag structure to be defined - at the very least, by defining a standard message separator. However, it does not require restructuring of individual messages. This is a much more important change than the first, and as the saving is likely to be less, it is proposed here that it should await the results of experiments with the Internet Mail Protocol.

### 3.3 Message Transport

There are two major choices to be made for the message transport service, namely the TCP FTP, derived from the ARPANET FTP, and the NI FTP. It is expected that the first will be used for mail within the Catenet, using the same MAIL option as used within the ARPANET. As has been seen above, however, this protocol is unsuited to our needs because it is uneconomic. It may be retained initially, as it gives direct compatibility with other Catenet sites.

In the slightly longer term, the NI FTP is the more attractive option. The reasons for this are its independence of specific transport services and the fact that it will be widely adopted in the UK. UCL already has implementations on its research Unix and at ISIE (though these will have to be changed to reflect the final specification); an implementation at RSRE is planned; and future mail servers in the UK will prefer to use it. The fact that many of these will run above X25 networks while Catenet sites will use TCP is immaterial; the necessary transport-level conversion will be handled by the UCL Protocol Converter. The existing ARPANET FTP is demonstrably NCP-specific, and the TCP version of this will at the minimum be Catenet-specific in its use of Telnet.

### 3.3.1 FTP Staging

An important consequence of this is that FTP staging will be required, for three reasons.

- (i) It will be necessary to stage messages into and out of the ARPANET. This applies regardless of the FTP used, as ARPANET mail is restricted to use of the ARPANET FTP.
- (ii) It will be necessary to stage messages between mailers in the Catenet using the TCP FTP and those using the NI FTP. If UCL does decide to use the TCP FTP, this decision is merely postponed until a UK community emerges based on the NI FTP.
- (iii) It may eventually be necessary to stage messages between UCL and Telenet/Tymnet servers, even if they adopt a common format, if a different transport mechanism is used.

It is proposed here that experiments with the first two stagings be performed at ISIE, or some other TOPS20 on the ARPANET which has all three systems. In its final form, the staging system would consist of a daemon which would process the mail file at a special account and forward messages to the appropriate sites. The structure of such a system is shown in Figure 1.

### 3.4 Addressing

Only four message sites in the UK are initially

is the slightly longer term, the MI FTP is the more attractive option. The reasons for this are the independence of specific transport services and the fact that it will be widely adopted in the UK. UCL already has implementation on its research Unix and at ISIE (though these will have to be changed to reflect the final specification); an implementation at RSRE is planned; and future mail servers in the UK will prefer to use it. The fact that many of these will run above X25 networks while Catenet sites will use TCP is immaterial; the necessary transport-level conversion will be handled by the UCL Protocol Converter. The existing ARPANET FTP is demonstrably NCP-specific, and the TCP version of this will at the minimum be Catenet-specific in its use of Telnet.

### 3.3.1 FTP Staging

An important consequence of this is that FTP staging will be required, for three reasons.

(i) It will be necessary to stage messages into and out of the ARPANET. This applies regardless of the FTP used, as ARPANET mail is restricted to use of the ARPANET FTP.

(ii) It will be necessary to stage messages between mailers in the Catenet using the TCP FTP and those using the MI FTP. If UCL does decide to use the TCP FTP, this decision is merely postponed until a UK community emerges based on the MI FTP.

(iii) It may eventually be necessary to stage messages between UCL and Telenet/Tymnet servers, even if they adopt a common format. If

Figure 1: Staging Daemon System

expected to be heavily involved in the system. Initially, development will be in the UCL message server itself (UCL-MUnix), while at a later stage the UCL teaching and research machines (UCL-TUnix and UCL-RUnix), and at least one machine at RSRE will become involved. While other message servers may emerge at a later date, it is not expected that this will happen rapidly. Staging to Catenet and ARPANET sites will be through ISIE; the problem of staging to Telenet/Tymnet



sites must be considered if and when it arises.

The UK sites should be able to exchange mail directly through the use of addresses of the form 'user@site' (e.g. Ruth@UCL-TUnix). This format could be used throughout the mailing address space, although it involves the message sites not under UCL control to make special modifications to their mailers. Thus an ARPANET mailer presented with a return address 'Ruth@UCL-TUnix' would have to recognise that this should be sent to ISIE; the ISIE mailer would have to recognise that the message should be added to the UCL daemon's mailbox and the UCL daemon would then forward the message to UCL-TUnix.

Two other alternatives are source routing and hierarchical addressing. A source routed form of the address might be identical in appearance to the ARPANET (by making 'UCL' a synonym for ISIE, in much the same way the 'UDEL-EE' is a synonym for 'Rand-Unix'), although for parsing purposes it would be preferable to rearrange it: (Ruth-(TUnix@(UCL))). Local messages would then appear as: Ruth-TUnix. An ARPANET address would appear to a message server user in a form such as: Kirstein-ISI@ISIE. Staging message servers would be required to parse the address into intermediate forms. Further, the terminal staging server for the catenet and for ARPANET would be required to suppress intermediate fields. Thus the UCL daemon at ISIE would have to transform all addresses of the form: Kirstein-ISI@ISIE to Kirstein@ISI and back again for traffic in the reverse direction. Source routing is the favoured solution of the University of Delaware's MMDF group.

Hierarchical addressing is actually the official ARPANET standard as described in RFC 733, although it is not implemented. It is also the solution favoured in Postel's Internet system. Under this scheme UCL would refer to a widely-known addressing domain, and addresses would take the form: Kirstein-ISI@ARPA and Ruth-TUnix@UCL. In practice, since only two hops and only one staging point are involved the two forms are virtually synonymous - which is a good argument for postponing a real decision until we see an addressing hierarchy actually emerging! The differences will be seen when an RSRE server becomes active. In this case, an ARPANET site has the choice of the following forms:

Bryan@NSide	(global)
Bryan-NSide@PPSN	(hierarchical)
Bryan-NSide-MUnix@ISIE	(source routing)

Note that in any form changes of the type above are required to ARPANET mailers. With global and hierarchical addressing, ARPANET tables must be modified to recognise mail servers (global address) or mail address spaces (hierarchical address). This is not required with source routing. The mailer at the staging site must additionally recognise that account names taking a certain format should automatically be accepted and routed to the UCL mail daemon at that site. Both solutions therefore require some structuring of the address. In the examples above, a hyphen ('-') has been used as a component separator. In fact, this is probably a bad choice. Two possibilities are:

- (i) Use of some other separator, such as %.
- (ii) Use of the comment fields allowed by the mail protocol.

The second choice has the convenient side effect that the account checking procedure need not be changed at the staging site, as addresses may then look like: UCLfor a source-routed format). However not all message preparation facilities will include comment fields (e.g. 'answer' under MSG).

Since this note was first drafted my attention has been drawn to RFC754 (Out-of-Net Host Addresses for Mail by J. Postel). This note considers four solutions: three are variants on the global solution, and the fourth involves name structuring. Postel's note favours a structured name solution. This is compatible with either a source routed or hierarchically structured solution.

### 3.5 Status Reporting

Finally in this section there is the issue of status reporting. Currently, most ARPA-type message systems give an immediate report, with possibly a mailer-generated message if there is some subsequent failure. For staged or mailbagged messages an immediate report of success can only imply success at the first stage. Thus it is important that staging daemons which cannot successfully deliver a message must be prepared to generate messages indicating why failure occurred. This can be done simply through the use of the current message generation mechanism.

#### 4. Message Server Design

##### 4.1 User Interface

The primary service which must be provided is a reliable, efficient and cheap method of sending and processing text messages exchanged amongst the user community. It is not intended to provide a multimedia service, although this is an important research goal of the program. Within this constraint, a user of the message server must be able to:

- (i) Prepare messages.
- (ii) Send messages to remote users.
- (iii) Receive messages from remote users.
- (iv) Read messages.
- (v) Be assured that messages are safely stored and are recoverable in the event of system failure.
- (vi) Be able to obtain adequate online help on the use of the server.

In addition it is desirable that the user be able to:

- (i) Prepare message files which may not be sent immediately.
- (ii) Archive and dearchive messages.
- (iii) Manipulate messages in file structures of his own creation.
- (iv) Answer and forward messages.
- (v) Obtain hardcopy listings.
- (vi) Maintain mailing lists.
- (vii) Annotate messages.

This list is clearly not exhaustive, and the aims of the user interface should be continually reevaluated in the light of user experience, development experience, and the recommendations of other message groups, such as IFIP TC6.5. Nor does it imply any evaluation of the difficulty of implementation: answering and forwarding



messages should be comparatively trivial; while a satisfactory remote hardcopy listing service is a major problem.

Following the general approach taken in this note, it is proposed that MSG be used at least initially as the basis of the user interface in the message server. The user would enter MSG automatically as his login shell. It is expected that the repertoire of commands will be changed and extended in order to provide the full range of services listed above (e.g. for the maintenance of mailing lists). This may require the single-letter command interface to be modified. It is also expected that the character-at-a-time interface and the use of TV editors would have to be altered to fit the needs of users accessing the system via XXX terminals, which favour line-oriented commands and editors. These issues will be reexamined in the light of experience gained.

#### 4.2 Message Management

An important issue is the internal design of the message server. The current system of personal mailbox files each containing a copy of all messages is complex and wasteful in a Unix system solely devoted to message handling. It is proposed here that database structures be examined in which only one copy of a message is kept in a central directory, and that the user's current mail file, and any other mail files he keeps, consist solely of descriptors pointing to the message and to other cross-referencing descriptors which may be needed. The structure of the system is shown in Figure 2.

The details of the descriptor structure are not considered in this note. However, a number of important issues arise. The fundamental question is: should all messages be kept in a single file, or each message in a separate file? The answer chosen has important implications for the limits on the size of the system, the method of updating the system, methods of accessing messages, and many other issues.

In the second method, messages may be found rapidly by filename, and garbage collection is considerably simplified through the use of Unix file management facilities, but on average 256 bytes (half a disc block) will be wasted per message. Further, at most an entire file system of 64K blocks can be allocated to message service, although this is not a serious



The first method allows more efficient use of space and places a much looser restriction on the number of messages that may be retained, but requires building searching and garbage collection facilities parallel to Unix's. In order to use these, moreover, either a complex file structure must be defined, or a master descriptor file retained.

Pending further investigation, the second choice is favoured at this stage. The fact that only one copy of a message need be kept should help to minimise the effects of the restrictions. Ensuring this may be a problem, especially if multiple copies of a message are received. Hence an important aspect of the system may be to examine incoming messages and attempt to detect duplicates of existing messages.

## 5. Conclusions

The message system discussed here is centred around text messages based largely on ARPANET-style formats, at least initially. Nevertheless there are several important issues which must be resolved in order to bring up a workable system. These issues include:

- (i) Economic use of transfer and storage resources.
- (ii) The structure of UCL-style mail daemons (e.g. `sendmail`).
- (iii) The modification of other mail servers to handle UCL mail.
- (iv) Basic addressing style.
- (v) Detailed user interface.

Figure 2: Message Management Structure

This note has indicated some lines of approach to these problems. They will be examined in more detail in subsequent work.

restriction. Assuming that most messages will be small, of the order of 2K characters, the file system would allow something less than 16K messages, wasting some 4K bytes of space. Thus a more serious limitation is the number of inodes (file descriptors) allocated to the system, which is currently about  $2^{13}$  - allowing 8K files. Increasing this to  $2^{14}$  is not difficult and will allow 16K files, of which a significant proportion would be for user descriptor information.

The first method allows more efficient use of space and places a much looser restriction on the number of messages that may be retained, but requires building searching and garbage collection facilities parallel to Unix's. In order to use these, moreover, either a complex file structure must be defined, or a master descriptor file retained.

Pending further investigation, the second choice is favoured at this stage. The fact that only one copy of a message need be kept should help to minimise the effects of the restrictions. Ensuring this may be a problem, especially if multiple copies of a message are received. Hence an important aspect of the system may be to examine incoming messages and attempt to detect duplicates of existing messages.

## 5. Conclusions

The message system discussed here is centred around text messages based largely on ARPANET-style formats, at least initially. Nevertheless there are several important issues which must be resolved in order to bring up a workable system. These issues include:

- (i) Economic use of transfer and storage resources.
- (ii) The structure of UCL-style mail daemons at staging site(s).
- (iii) The modification of other mail servers to handle UCL mail.
- (iv) Basic addressing style.
- (v) Detailed user interface.
- (vi) Message management issues.

This note has indicated some lines of approach to these problems. They will be examined in more detail in future notes, prior to the commencement of actual work on the system later this year. It is clear that satisfactory progress requires cooperation and discussion with other parties, notably the DARPA Catenet group and groups using various public carrier services. While the projects of the former are more advanced at this point, it is expected that the latter groups will become increasingly important in the long term.