INDRA
Working
Paper

INDRA Note 1025
IEN 169
23rd January 1981

A Simple NIFTP-Based Mail System

C. J. Bennett

ABSTRACT: This INDRA Note proposes a draft mail protocol for use in a wide variety of network situations. This protocol draws heavily on existing facilities and could be brought up very quickly - in months rather than years. It could thus be used as an interim solution until international standards emerge. An experiment in the ARPA Catenet is proposed, and the use of the system in the UK is discussed.

Department of Computer Science
University College London

Table of Contents

1. Introduction

One of the major uses of computer networks is for electronic mail services. Now that networking technology is proliferating rapidly, it is highly desirable that such services should be made available with the new networks. However, despite the recent adoption of Teletex proposals by CCITT, it may be some time before one can expect international standards to be finalised and widely available. Hence there is a strong need for a simple interim scheme, which could cover the basic needs for mail transfer across multiple networks and through intermediate mail relays in a flexible fashion. We at UCL are particularly concerned with this, as we see the need for a scheme which will operate both with the facilities available in the United States, and those rapidly becoming available in Britain and Europe.

This note contains a proposal for such an interim protocol, discusses the requirements on mail relays, and also the requirements for experimental systems based on it, both in the US and the UK. Partly because the proposed system uses ARPA facilities, the document has a certain bias towards familiarity with ARPA systems and terminology. In particular, the note assumes familiarity with RFC733 mail formats [Crocker77], the ARPA standard. However, it is believed that the system will also be useful in other environments, particularly the X25 and Network Independent Transport Service [SG3/80] environments which will be available shortly in the UK. ARPA-oriented readers should note that familiarity is assumed with the NIFTP [HLPG81], the interim UK file transfer standard.

Comments are solicited and welcomed.

2. Requirements

The basic requirements for an immediate mail service are as follows:

(1) Maximum use must be made of existing standards. No radical new mail formats, transfer protocols, etc may be imposed. Using these facilities it should be possible to bring up an initial service within months rather than years.

(2) The service should be economic. In particular, only one copy of a message should be transmitted to each destination mail server.

(3) The service should be independent of particular networks.

(4) Address information must be transmitted in a way which is usable by mail relays.

(5) Total global knowledge of mail server addresses should not be necessary.

(6) The service should be as flexible as possible. Where possible it should be easy to extend it to meet new needs as they emerge.

The last requirement is perhaps the least important. It implies that the service will be around for sufficiently long that experimental technical advances in message processing can be usefully incorporated. For instance, the minimum requirement is that text messages must be transferable. Requirement (vi) suggests that it should be possible to extend the service simply to transfer mixed-media messages. When assessing whether a given proposal meets this criterion, the multi-media case will be used as a test.

No existing system in wide use meets all these constraints. Hence it is necessary to combine elements with the right properties from a number of sources not all of which are currently used for mail.

It should be noted that the service defined here is not intended to be perfect. In particular, it does not of itself guarantee bidirectionality, endpoint reliability, or use of address lists. These will normally be available for direct endpoint transfer, and any problems are most likely to arise if intermediate relays are used. It is possible to support all these things using it, and of course they are encouraged. However, I feel that a service adequate for ordinary use can be achieved without defining a great deal of complicated additional mechanism to guarantee these properties. Finally, the problem of conversion between mail formats is not discussed at all.

3. Basic Elements

3.1 Mail Format

The mail format defines the standards for the appearance of a message; it covers such things as the definition of header fields to allow messages to be processed by a standard program.

The mail format most readily available is the ARPANET mail standard commonly known as RFC733, defined in [Crocker77], which is compatible with all except the last of the objectives listed above. RFC733 has been widely and successfully used for many years. In practice only a subset of the standard is actually used - in particular, the standard allows extended addressing by specifying destination networks, but no implementation supports this. It is proposed that the common subset of this standard be used. The only change suggested is that the text code should be IA5 rather than the Telnet version of ASCII, as IA5 is an international standard. In practice, the two codes are virtually identical. An example of an RFC733 message, and a summary of the RFC733 syntax, is given in Appendix I.

This standard is completely text-oriented, for both message header (control) and message text (data) information. This makes it readily compatible with most of the text-processing software generally available on most interactive systems, such as text editors, pro forma preparation etc.

Other formats may be agreed to in the near future. These may extended facilities, such as facsimile or multi-media mail. These can be accommodated provided any mail servers needing to process the mail body understand the format in use. Hence it may be necessary to provide a means for labelling the format in use. Such a label is not defined at this time.

3.2 Mail Addresses

In any mail system it is essential to provide addresses for the mail recipient, and usually for the mail sender as well. While mail will frequently be transferred directly between the sender and the recipient, there will be occasions when it will be routed, and possibly temporarily stored, through an intermediate mail relay.

Mail addresses are constrained to be compatible with the RFC733 format as generally used, viz. <username>@<hostname>. The <hostname> field defines the host used for an initial mail transfer, and in standard RFC733 usage, no further structuring is defined or required. If this name is understood by all message relays along a route, then no further structuring is ever required. If further structuring is required, it should be through the use of element separators in the <username>. The addressing structures encountered when mail relays must be used are discussed further in section 5.1.

3.3 Mail Transfer

In addition to the appearance of the messages and the identity of the parties, it is necessary to define a protocol for transferring mail between the two. At first sight, it would seem natural to take over ARPA-developed procedures here too, so that one could use a complete, preexisting mail package. Despite the great success which have attended these procedures, and their undoubted appropriateness for the environment for which they were designed, they do not fulfil the needs of the wider message community, for reasons which are discussed below.

The standard ARPANET Mail Transfer procedure [Postel76] fulfils only the first of the requirements listed in the introduction, and is therefore not acceptable. In particular, it has the following failings:

(1) It is uneconomic in that it transmits one copy per destination user.

(2) It is specific to the ARPANET.

(3) It transfers address information in a way which is only usable for direct source/destination transfer.

(4) It requires all hosts to be aware of all host names, and hence requires a globally understood global address space.

(5) It can only ever handle text data. If binary data in a mixed-media message were encoded as text symbols and a code conversion was required between NVT-ASCII and a local text encoding, that data would be corrupted.

A recent ARPA proposal [Sluizer80] for a new Mail Transfer Protocol (MTP) remedies in whole or in part some of these deficiencies. In particular, it provides facilities for economic uses of resources, and transfers addressing information in a way compatible with objectives (iv) and (v). However the MTP as it is currently proposed has some problems of its own:

(1) The MTP allows a single copy of a message to be sent to multiple recipients, and is thus potentially more economic than the standard ARPA protocol. However, this is only an option which need not be implemented. Moreover, the mail sender can only specify one path for a given message to pass through intermediate relays. Thus MTP does not allow a single copy to be sent to a relay which could then decide to create multiple copies for different destinations at the point of a path division.

(2) In MTP, address lists may be transferred either before or after the message body. With the 'recipients first' option, it is only possible to check the current validity of the addresses - for the actual message transfer, only total success and total failure can be indicated. If a given recipient became inaccessible for some reason between the two stages, the entire transfer would fail.

(3) The MTP is defined to operate in an environment similar to that of the ARPA Catenet. In particular, it relies heavily on the use of Telnet command procedures, which are both little known and little used outside the ARPA community - especially within Europe. It does not define the mechanisms by which it will operate across more than one network (just as Telnet does not), or where Telnet procedures are not appropriate. To this extent, it is not network-independent.

(4) The MTP provides no restart procedures to recover from errors signalled by either the host or the transmission medium. It is therefore only as reliable as the weakest underlying network. For example, an X25-based version could not recover from Resets.

(v) It can only ever handle text data. If binary data in a mixed-media message were encoded as text symbols and a code conversion was required between NVT-ASCII and a local text encoding, that data would be corrupted.

Accordingly, it is felt here that mail transfer outside the ARPA environment should use an alternative base. It is proposed here that mail transfer be achieved using defined conventions with the Network Independent File Transfer Protocol (NIFTP - [HLPG81]). This FTP is, as the name implies, network independent, has been implemented on a number of different existing networks, including the ARPANET and ARPA Catenet, and has been successfully used for direct file transfers across several intermediate networks. The revised version will be adopted as an interim standard in Britain and has evoked wide interest in Europe. There are numerous implementations of the existing version, and it is expected that the revised version will be implemented fairly rapidly after the new specification is released, which should be within the next two months. It thus meets the criteria of availability, standardisation and network independence. It remains to define a transfer procedure which meets the other criteria.

4. Point to Point Mail Transfer

Using RFC733 mail formats and RFC733-compatible addressing, it is necessary to define the procedure used to transfer mail from a mail donor to a mail server with the NIFTP. This section defines that procedure.

4.1 Mail Structure

4.1.1 Proposal

A letter shall be transferred as a single file from the mail donor to the mail server. The file name used shall be provided by the mail server. The structure of the file is as follows:

<address list><one or more blank lines><mail text>

The <address list> is a list of full, explicit RFC733 addresses to whom the mail shall be distributed by the mail server.

The <mail text> shall conform with RFC733 formats.

4.1.2 Discussion

4.1.2.1 Address Lists

This structure is designed to fulfil the requirement that the service be economic. The passing of the <address list> minimises the number of copies of the message which must be made by the donor, but allows decisions to create additional copies to be made simply by intermediate relays.

The <address list> must contain explicit addresses as the mail server is not necessarily able to access address lists, and in any case requires additional mechanism to do so. The addresses should be full (i.e. contain an explicit <hostname> component) as the server may be a relay using address mapping mechanisms (see below).

4.1.2.2 Possible Extensions

There are two simple extensions which may be desirable:

(1) To distinguish between message formats. This requires simply the addition of an extra field in the mail file, and the definition of text encodings. Such a field should be inserted between the <address list> and the <message text>. The use of other formats will particularly affect the design of mail relays.

(2) Mailbagging. The file may contain several messages separated by defined message delimiters. (A simple one, which is widely used in message files on UNIX systems in the ARPANET, is ↑A↑A<cr>. Another alternative, preferred here, is to insert a delimitered character count, encoded in IA5, as in TENEX.) Mailbagging also has other implications. For instance, if the mail donor wishes to initiate a mailbagged transfer and it knows the name of an existing mailbag at the server from a previous transfer, it may append the file to the existing

mailbag. The advantages of a mailbagging system are for further study.

4.1.2.3 Alternative Structures

Two alternative structures were considered, both involving the explicit separation of the address list from the mail text.

The first was to require the donor to specify the file name, which would be the address list. This has a number of disadvantages:

(1) The NIFTP allows a maximum string length of 255 characters for string-valued attributes. This would allow at most about 12-20 addresses.

(2) It would be difficult to trace the progress of a message through a series of relays. With an explicit and known filename, it would be possible to initiate manual or automatic tracing procedures.

(3) It is unlikely that most mail servers or relays would be able to use such a filename directly. The internal filename would be created using local mappings. The potential costs of these mappings could be very high.

The second alternative considered was to transfer the address list and mail text as two separate files. This also has disadvantages:

(1) The two files must be linked, e.g. by requiring that an Action Message be passed on STOP or STOPACK containing the filename to be used for the text portion of the transfer.

(2) An additional level of error handling procedure must be defined, to cover cases such as the loss of a portion of the message text, or the arrival of two address lists with no intermediate message text.

.hese problems are avoided by the mechanism proposed. By sending the address list and the body as a single file, address lists of arbitrary length can be sent; their link to the text is assured; maximum use can be made of the NIFTP reliability mechanisms; and the donor can be assured that the mail has at least reached the server host.

The new MTP proposal from the ARPA community has a fairly similar proposal, but allows as an additional option the possibility of transmitting the text before the addressing data, arguing that in some cases this is more efficient for the destination host. Although it is conceivable that this may be true for MTP given the details of the MTP mechanisms, I think it is most unlikely that any advantage would be gained in the current context; moreover, in order to achieve it additional mechanism is required to specify which option is being used. Hence it has not been proposed.

4.2 Mail Server Identification

4.2.1 Proposal

The mail server will be identified by its transport service subaddress. This subaddress will be network-specific and possibly host-specific; for instance, on the ARPANET it will be an NCP server socket, on the ARPA Catenet a TCP port, on public data networks an X25 or Transport Service subaddress as appropriate.

If the mail donor and server are not on the same transport service, it is the responsibility of the intermediate virtual call gateways to perform any address transformations required, e.g. mapping the NCP mail socket to the TCP mail port.

4.2.2 Discussion

This proposal is in line with the recommendation in the NIFTP specification for distinguishing different services utilising NIFTP. An alternative, which is not favoured here, is to reserve a value for the Username attribute, such as MAIL.

4.3 Mail Server Communication

4.3.1 Proposal

Synchronisation shall be achieved via the establishment of an virtual connection between the mail donor and the mail server. This connection may be used for one or more mail transfers. The connection may have one or more segments, where each segment may use a different transport protocol.

4.3.2 Discussion

This is normal NIFTP usage, and is essentially a restatement of the network-independence criterion.

Normally, the donor and server will use the same transport protocol, and no additional procedures need be used. Exceptionally, the mail donor and server may not be on the same transport service. In this case direct NIFTP file transfer is still possible, but an additional degree of support is needed, through the provision of one or more virtual call gateways. At the least, the following services are necessary:

(1) One-to-one connection mapping.

(2) Addressing procedures. This could be any acceptable procedure, e.g. source routing, creation of a hierarchical address space, or mapping of transport service subaddress to destination host.

(3) Call request facility. This carries all the addressing information necessary for establishing an end-to-end path.

(4) Call accept facility. This is necessary to confirm that an end-to-end path has been established.

(5) Data transfer. This should commence only after a call accept has been received.

(6) Push preservation. Data should be forwarded if any push indication (e.g. TCP EOL, X25 No More Data) is received. The gateway may decide to forward data in other circumstances.

(7) Closure propagation. If a connection closure is received from one side, it shall be mapped into the appropriate closure procedure on the other.

(8) Reset propagation. If any network in the path is capable of generating resets, these must be forwarded in some fashion by the gateway. For instance, an X25 RESET may be mapped into TCP URGENT. If resets cannot be generated this may be ignored by the gateway.

It should be noted that the address transformations mentioned above need not be visible to the mail donor and server, and do not in any circumstances require address processing of the mail text at the virtual call gateway. For bidirectionality, however, it is necessary that the donor and server can recognise their respective hostnames as embedded in the mail file.

4.4 Mail Transfer

4.4.1 Proposal

The transfer may be initiated by either the mail donor or the mail server.

The file will be transferred by the NIFTP using IA5 text codes. If the file only contains a text message, then the Data Type attribute will indicate text only.

If the transfer is initiated by the mail donor, then the Mode of Access used will be 'Replace or Make' and the Filename attribute on the SFT will indicate 'no value available'; the server should supply a value on the RPOS for reporting purposes.

If the transfer is initiated by the mail server, then the Mode of Access used will be 'Read Only'. The donor will nevertheless often wish to delete the file after it has been successfully transferred. The Filename attribute on the SFT will supply a value.

No other constraints are imposed on the use of NIFTP attributes.

4.4.2 Discussion

This section defines the actual mail transfer procedure, and places minimal restrictions on NIFTP use.

Alternative structures lead to greater restrictions or special interpretation of NIFTP attributes, or both.

If a message format other than RFC733 is used, then mixed-media transfers may be possible. The NIFTP procedure would then have to be modified. If the file contains a mixed-media message, then the Data Type attribute will indicate either mixed file or mixed records as appropriate, and the binary formats for the non-text portions will be negotiated.

Mailbagging may also require extension. In this case, the mode of access used by a mail donor initiating the transfer could be 'Append or Make', though this is not recommended.

Other facilities which may be useful are: data compression, text formatting, record structuring, restart and resumption recovery facilities, account name, various passwords etc.

4.5 Reliability

The NIFTP has several grades of recovery action, which can be exploited to ensure that a message will be delivered to a server despite the occurence of system or communication errors. However, the successful delivery of a message to a server does not guarantee it will be successfully delivered to the recipient. If it cannot be delivered, notification should be sent to the sender by the server forced to discard it, where this is possible. This notice will take the form of a message, and the sender's address will be determined from inspection of the appropriate fields (i.e. "Reply-to:" and "From:") in the message. A non-delivery notice should make maximum use of the NIFTP reliability procedures to ensure that it itself is delivered.

5. Mail Relays

For a number of reasons it may not be possible to transfer a message direct between the sender and the receiver. In particular, they may not use the same mail transfer procedure.

In such cases, mail must be staged through an intermediate mail server, which acts as a mail relay.

The function of the relay is to redistribute received mail to the destinations or to other mail relays. I do not intend to specify a mail manipulation protocol here, but it is necessary to discuss the functions which must be provided, and the options which are available, in order to determine to what extent it is possible to allow variation and still provide an acceptable service. Since the actions of other mailing systems cannot be specified here, these functions will be discussed, where necessary, in relation to the NIFTP-based system described above.

It is important to distinguish these relays from the virtual call gateways discussed above. Either may be used at the interface between two transport services, but the virtual call gateway gives the minimum facilities which must be provided at this point, and is invisible to the endpoint mail transfer. Mail relays must be used when different mail transfer protocols are used by mail donor and recipient, and will be visible to both the sender and recipient; in this case a virtual call gateway will be totally inadequate.

5.1 Address Processing

It is not possible to prescribe an addressing format for use by relays, except that it be RFC733-compatible. The actions to be taken by the relay on processing addresses are dependent on local conventions and private agreements. It is expected that there are three major types of address processing which may occur:

(1) The address of the next stage may be defined by a received <hostname> component, which is understood by the relay to map to the name of some other host. For example, the name 'Linington@Cambridge', if received at UCL from the US, might cause the mail to be transferred to Cambridge, or to some intermediate relay.

(2) If no <hostname> is received (which can only happen from a non-NIFTP mailing system), the address of the next stage may be defined by a received <username> component, which is understood by the relay to map to

the name of some other relay host, or to the <username> and <hostname> of the final user. For example, the name 'DCPU', if received at UCL from the US, might be understood to map to 'Linington@Cambridge', and be forwarded. This form is not encouraged as user names must then become widely known.

(3) If the <hostname> is the relay itself, and the destination is not at the relay, then either case (ii) applies, or the username is structured in some fashion understood by the relay. A recommended form is through the use of % as a separator, which leads to a source-routed form, e.g:

Linington%Cambridge%PSS@UCL.

On reception at UCL from the US, the <hostname> component will be discarded, the last % converted to an @, and the structured name becomes: Linington%Cambridge@PSS. The relay then injects the message into the appropriate mailing system over PSS, subject to the constraints of that system.

Any or all of these approaches may coexist. As a general approach, I prefer the third. All suffer from the disadvantages that they are not global, and that address transformation may be required. The user who uses relays must use an address format he is sure will be understood along the route. In practice, however, it is unlikely that more than one or two relays will be involved in the transfer of most messages.

The address processing at mail relays, which may affect the text of the message received, must be carefully distinguished from the address processing which may be required at virtual call gateways between the donor and server, which does not. Two different levels of addressing are involved here - the former is visible only in mail, the latter only within a particular file transfer.

5.2 Return Paths

The system provides no guarantee that a message can be replied to, although this will normally be possible. Relays can provide assistance in the following ways, both of which involve the processing of the header content of the message:

(1) Each relay can insert a "Via: " field in the message.

(2) Each relay can add its name to a "Reply-to: " field, thus building up a return source route. The name added must be the name known to the message system into which it is forwarding the mail.

The general problem is to allow replies to be sent to recipients other than the sender. One possibility is to allow replies to be redistributed by the original sending host, if that host is willing to do so. Alternatively, intermediate relays could process the names of "To: " and "Cc: " fields in the message to provide a shorter path. This problem is exactly analogous to the "third-party addressing" problem of the UK Transport Service [SG3/80], and the techniques discussed in that document could be used here.

Although this procedure requires relays to process the message text, programs to do such processing already exist which could be used for this purpose. If such processing is not done, it is necessary to insist that replies can only be sent if the recipient knows the location of the sender, which for full answerability amounts to a requirement for a global address space. This contravenes one of the stated limitations on the system.

5.3 Economy

Where the mail system protocols allow, the relay should minimise the number of copies of a message injected into the system. Thus a relay may receive a single copy of a message destined for several different hosts, some of which may be only accessible through another relay. For each host which can be reached directly the relay will send a single copy of the message; for the remaining hosts, a single copy will be sent to the next relay which will redistribute it in turn.

This minimisation may require considerable intelligence to do properly, and may not always be practicable. If the relay is receiving mail from a system which creates one copy per user, and injecting it into a system similar to the NIFTP-based system described here, there is no easy option. One possibility is to parse the header to identify, so far as possible, how many copies of the message it may expect to receive, detect the duplicates, and discard them. Another is for a relay to create a mail list and instruct a recipient to "Reply-to: <mail list name>@<relay>". It must then have criteria for deciding how long to keep such lists around. No doubt other equally inspiring schemes can be devised.

5.4 Reliability

Relays should make use of the mail system to give delivery failure notifications, as described above. There is, however, no guarantee that the return path can be constructed.

6. The ARPA Mail Transition Plan: A Case Study

In this section a proposal is made for using the NIFTP-based protocol to allow mail transfer between ARPANET users who have only NCP-based mail transfer available to them, and those who have only TCP-based network access.

6.1 Current Proposals

The current proposals for ARPANET mail transition centre on the implementation of the MTP discussed above, and the definition of relay procedures between NCP-based mailing systems and TCP-based mailing systems [Postel80]. In general, these relays fit into the context discussed in the previous section, and most of the comments of the current proposal on the complexity of maintaining relays, mapping tables etc are fully endorsed here.

Aside from the features of the MTP, there are a two specific points that need discussion:

(1) As noted in the October 1980 Internet meeting, the transition plan requires that user names be unique throughout the ARPA Catenet (and hence a growing portion of the ARPANET), in order to allow ARPA mail from a standard ARPANET NCP-based mail server to be sent to a relay for forwarding to the ARPA Catenet. This is clearly unacceptable, and can most simply be

avoided by allowing the relays to parse, and if necessary modify the header fields in the message text. Such solutions are preferable.

(2) The solution is inextensible. The transition plans assumes that transfer of mail between two MTP servers on hosts using only NCP and TCP must be achieved through an MTP-based mail relay at a site with access to both. This is rather wasteful, since essentially the same protocol is involved in both cases. A similar relay is required if other transport services, or network services such as X25, require mail service.

6.2 NIFTP and the Transition Plan

The major fault of the current transition arrangements relevant here is that a complex message relay is required even for hosts which both talk MTP. This is not the case for the NIFTP-based scheme outlined here. All that is necessary is a relatively simple virtual call gateway at an NCP/TCP transition point, along the lines discussed above. Such a gateway has been built and its feasability demonstrated [Bennett80]. Moreover, it has been demonstrated that the NIFTP can be readily extended to non-Catenet networks, whereas it is far from clear that this is true for the MTP-based scheme, because of its reliance on Telnet.

The advantage of an explicitly network-independent approach is that mail transition can now be entirely divorced from NCP/TCP transition. The development of future mail protocols can carry on without requiring an immediate, new, solution for the Catenet. Any host with NIFTP can do direct mail transfers using existing formats.

Of course, very few ARPANET hosts have access to NIFTP, although this is easy to change. However, it is not necessary that they should. There is indeed a staging problem to be solved - the staging between NIFTP-based mail and current ARPA mail. This must take place along the lines discussed above, but once solved, it is solved, in theory, not just for NCP and TCP but for any transport protocol.

6.3 A Proposal

It is believed that an NIFTP scheme of this sort can be built and proliferated very quickly. The following components already exist:

(1) NIFTP implementations written to a transport service interface for TOPS20 DEC20s, UNIX PDP-11s (Version 6), and MOS LSI-11s. These need to be upgraded to conform to the new specification of the protocol. The first is available above TCP and NCP; the second will shortly be accessible above TCP and X25; and the third is available above TCP.

(2) A simple NCP/TCP virtual call gateway, for NIFTP support, on a TOPS20. This was built for demonstration purposes, and needs some modification for general use.

(3) Message relays for heterogenous mail systems, in the form of the MMDF system developed by the University of Delaware [Crocker79], for UNIX. Such relays must be built in any case for the MTP scheme.

The following components are needed:

(1) Specification and agreement to the virtual call extensions needed for direct NCP/TCP file transfer. If these are done using a subset of the protocol proposed for implementing the UK Transport Service above TCP [Bennett80a] (and a similar protocol for NCP), then direct mail transfers from NCP sites to sites on the UK public network PSS could also be done.

(2) Allocation of NIFTP-mail server sockets and ports. Again, for extension to the UK public network, Transport Service and/or X25 subaddresses must also be defined.

(3) Agreement on an addressing scheme to allow transfer from ARPA mail to NIFTP-based sites. It is proposed that a structured user name of the form outlined above be used.

(4) Implementation of an NIFTP mail channel in a form suitable for incorporation under MMDF by UCL.

(5) At least one system in the US supporting MMDF which can act as a relay between ARPA mail and NIFTP mail, using the NIFTP channel supplied by UCL.

(6) Code interfacing the transport service interface of the UNIX NIFTP directly to UNIX TCP (and possibly NCP) implementations, supplied by the US MMDF site.

This allows us to define the minimum configuration necessary to provide NIFTP-based mail transfer between UCL systems and systems in the CONUS ARPANET using either the current ARPANET mail transfer protocol or the new MTP. The path would be as follows:

(1) UCL donor passes mail to UCL UNIX NIFTP in core.

(2) NIFTP initiates a connection to the remote NIFTP, which is which is driven as a mail channel by an MMDF system (e.g. SRI-UNIX or UDEL-EE). This path will be: through the UCL UIPP protocol to the Front End; thence via TCP through UCLNET, SATNET and the CONUS ARPANET, either directly to the remote system, or to a TCP/NCP virtual call gateway resident at ISIE (which in turn forwards the NIFTP traffic to the remote MMDF server through NCP).

(3) The remote MMDF injects the mail into either the standard ARPANET mail channel or an MTP channel; at the remote end of which it is delivered to the user's mailbox.

In addition, it would be highly desirable to construct an MMDF-like system which could act as a multi-channel mail relay on TOPS20 systems in the ARPANET. All relevant mail channels could be driven through it in a precisely similar manner. However, rather more work is required to make this service available.

The above discussion ignores MTP-based mail altogether. This has been done for illustrative purposes only - I have no doubt that the current transition plans will be implemented, though possibly not quite in their current form. In practice, the two systems could coexist quite happily. The main impact of allowing the two systems to proceed in parallel would be in the design of mail relays. The more mail transfer systems exist, the more important it is that relays be designed in a 'mail-independent' fashion. The advantages of this have already been demonstrated for the UNIX MMDF. The same principles should be followed in the design of relays for other computer systems.

7. NIFTP-Based Mail in the UK.

Within Britain, the problems of building a mail system along these lines are rather different, but on the whole less complex. The basic communications facilities are only now coming into widespread use, and the investment in higher level protocols is much lower. However, the higher level protocols which are coming into use are much friendlier to the system, for the following reasons:

(1) NIFTP is already available on most widely used machine types in Britain. Implementing the mail transfer procedure is a trivial additional exercise.

(2) The UK transport service proposals assume the need for network interconnection to provide a semantically equivalent service rather than a superimposed common protocol. Hence the need for extensibility through virtual call gateways is widely accepted.

(3) Because there is no heavy investment in first generation protocols, there is no absolute requirement for mail relays at this stage.

The major need is for message processing and preparation programs, as such programs are not widely available in the UK, except for UNIX systems. In particular, these should be based on RFC733 procedures. For many system types these may be available from similar systems in the US; others would have to be developed from scratch.

8. References

[Bennett80] - Bennett, C. J.: The Design and Implementation of

Transnetwork Systems. UCL TR 65. Available from Univeristy College London.

[Bennett80a] - Bennett, C. J.: Realisation of the Yellow Book Above TCP. INDRA Note 965. Available from University College London.

[Crocker77] - Crocker, D. H., Vittal, J. J., Pogran, K. T., Henderson Jr, D. A.: Standard for the Format of ARPA Network Messages. RFC733. Available from SRI International, Menlo Park, California. Obsoletes earlier versions.

[Crocker79] - Crocker, D. H., Szurkowski, E. S., Farber, D. J.: An Internetwork Memo Distribution Capability - MMDF. Proc. 6th Data Comm. Symp.

[HLPG81] - HLPG/FTPIG: A Network Independent File Transfer Protocol. Available from DCPU, National Physical Laboratory, Teddington, UK. Obsoletes earlier versions.

[Postel76] - Postel, J. B.: Mail Protocol. NIC 29588. Available from SRI International, Menlo Park, California.

[Postel80] - Postel, J. B., Cerf, V. G.: Mail Transition Plan. RFC773. Available from SRI International, Menlo Park, California.

[SG3/80] - PSS/SG3: A Network Independent Transport Service. Available from the DCPU, National Physical Laboratory, Teddington, UK.

[Sluizer80] - Sluizer, S., Postel, J. B.: A Mail Transfer Protocol. RFC772. Available from SRI International, Menlo Park, California.

APPENDIX I

RFC733 Formats

Below is an example of an RFC733 message taken from the specification.

```
Date: 26 August 1976 1430-EDT
From:George Jones<Group at Host>
Sender:Secy at SHOST
To:Al Neuman at Mad-Host,
           Sam Irving at Other-Host
Message-ID:  <some string at SHOST>
This is an example of an RFC733 message. Both
simpler and more complex headers are possible.
```

The entire RFC733 syntax specification is summarised in the following listing, extracted from the original specification:

A. ALPHABETICAL LISTING OF SYNTAX RULES

```
address       =  host-phrase / quoted-string
              /  (*phrase "<"  address ">" )
              /  (*phrase ":"  address ";" )
              /  (":" ("Include" / "Postal" / atom) ":" address)
ALPHA         =  <any TELNET ASCII alphabetic character>
atom          =  1*<any CHAR except specials and CTLs>

CHAR          =  <any TELNET ASCII character>
comment       =  "(" *(ctext / comment / quoted-pair) ")"
CR            = <TELNET ASCII carriage return>
CRLF          =  CR LF
ctext         =  <any CHAR excluding "(", ")", CR, LF and
                  including linear-white-space>
CTL           =  <any TELNET ASCII control character and DEL>

date          =  1*2DIGIT ["-"] month ["-"] (2DIGIT /4DIGIT)
date-field    =  "Date"        ":" date-time
date-time     =  [ day-of-week "," ] date time
day-of-week   =  "Monday"    / "Mon"  / "Tuesday"   / "Tue"
              /  "Wednesday" / "Wed"  / "Thursday"  / "Thu"
              /  "Friday"    / "Fri"  / "Saturday"  / "Sat"
              /  "Sunday"    / "Sun"
delimiters    =  specials / comment / linear-white-space
```

```
DIGIT        =  <any TELNET ASCII digit>

extension-field = <Any field which is defined in a document
                 published as a formal extension to this
                 specification>

field        =  field-name ":" [ field-body ] CRLF

fields       =  date-field  originator-fields  *optional-field
field-body   =  field-body-contents
                [CRLF LWSP-char field-body]
field-body-contents = <the TELNET ASCII characters making up the
                field-body, as defined in the following sections,
                and consisting of combinations of atom, quoted-
                string, and specials tokens, or else consisting of
                texts>
field-name   =  fnatom *(LWSP-char [fnatom])
fnatom       =  1*<any CHAR, excluding CTLs, SPACE, and ":">

host-indicator =  1*( ("at" / "@") node )
host-phrase =  phrase  host-indicator
hour         =  2DIGIT [":"] 2DIGIT [ [":"] 2DIGIT ]
HTAB         =  <TELNET ASCII horizontal-tab>

LF           =  <TELNET ASCII linefeed>
linear-white-space =  1*([CRLF] LWSP-char)
LWSP-char    = SPACE / HTAB

mach-id      =  "<" host-phrase ">"
mailbox      =  host-phrase /  (phrase mach-id)
message      =  fields *(CRLF *text)
month        =  "January"   / "Jan"  / "February"  / "Feb"
             /  "March"     / "Mar"  / "April"     / "Apr"
             /  "May"              / "June"      / "Jun"
             /  "July"      / "Jul"  / "August"    / "Aug"
             /  "September" / "Sep"  / "October"   / "Oct"
             /  "November"  / "Nov"  / "December"  / "Dec"

node         =  word / 1*DIGIT

optional-field  =
                "To"          ":" address
             /  "cc"          ":"  address
             /  "bcc"         ":" address
             /  "Subject"     ":" *text
             /  "Comments"    ":" *text
```

```
            /  "Message-ID:" mach-id
            /  "In-Reply-To"":"  (phrase / mach-id)
            /  "References:"  (phrase / mach-id)
            /  "Keywords"   ":"  phrase
            /  extension-field
            /  user-defined-field
originator-fields =
               (  "From"     ":" mailbox
                 ["Reply-To:"  address] )
            /  (  "From"     ":"1 address
                  "Sender"   ":" mailbox
                 ["Reply-To:"  address] )

phrase      =  1*word

quoted-pair =  "
quoted-string =  <"> *(qtext / quoted-pair)  <">
qtext        = <any CHAR except <">, CR, or LF and including
               linear-white-space>
SPACE        =  <TELNET ASCII space>
specials     =  "(" / ")" / "<" / ">" / "@"/ "," / ";" / ":"
            / "

text         =  <any CHAR, including bare CR and/or bare LF, but
               NOT including CRLF>
time         =  hour zone

user-defined-field = <Any field which has not been defined in
               this specification or published as an extension to
               this specification; names for such fields must be
               unique and may be preempted by published
               extensions>

word         =  atom / quoted-string

zone         = ( ("+" / "-") 4DIGIT )
             / ( ["-"] (1ALPHA
               / "GMT" / "NST"  / "AST" / "ADT" / "EST" / "EDT"
               / "CST" / "CDT" / "MST" / "MDT" / "PST" / "PDT"
               / "YST" / "YDT" / "HST" / "HDT" / "BST" / "BDT" ))

<">          =  <TELNET ASCII quote mark>
```

APPENDIX II

UCL Mail System Plans for 1981

The following is an extract from the UCL annual report for the Ministry of Defence, summarising our planned activities in 1981.

Plans for the message systems work in 1981 have not been finalised, as some areas depend on decisions and choices which have not yet been made. However, a rough scheme is as follows:

(1) Complete installation of MMDF in local systems (January). Supportive tasks will be required throughout the year, eg bugfixing as found, transferring and reconfiguring MMDF for new systems (eg the 11/44) as they become available.

(2) Issue specification of proposed NIFTP-based mail channel (January)

(3) Either

implement NIFTP-based mail channel over TCP under MMDF; if possible, after upgrading UNIX NIFTP to the 1980 specification. Optionally (but preferably) the UNIX NIFTP should also use Yellow Book TCP commands.

or

implement or obtain MTP mail channel over TCP under MMDF. If one is obtained from an outside source it must be modified to access TCP through the UIPP, and very probably modified so that it can be driven as a channel through MMDF. (1st - 3rd quarter)

(4) Depending on the choice made above, appropriate supportive action must be taken. (2nd to 4th quarter and beyond)

(1) NIFTP-based channel

(1) This should be released to an MMDF site in the US (either SRI or the University of Delaware) which could act as a relay.

(2) Help and assistance as required to interface NIFTP directly to TCP or NCP in the remote system.

(3) TCP/X25 virtual call gateway to allow access through IPSS and PSS. This should be in existance in any case.

(2) MTP channel

(1) TCP/X25 virtual call gateway, as above.

(2) Possibly TCP/NCP virtual call gateways. This depends on the progress of MTP/ARPA mail relays at TCP/NCP boundaries as required by the ARPA Mail Transition Plan.

(5) Replace MSG message processing system by MH, to allow remote access to UCL mail handling. (3rd/4th quarter and beyond).

While the above lists the primary path to providing mail services, there are a number of subsidiary or optional pathways which will also be undertaken, if necessary or desirable. These include the following:

(6) Continue efforts to provide ARPANET mail access through a terminal channel. (January/February).

(7) Undertake necessary activity to incorporate TOPS20/TENEX systems into either the NIFTP or MTP based scheme above. The latter case should presumably involve very little UCL activity. The former case requires the following from us (1st to 4th quarter and beyond):

(1) Optionally (but preferably) the upgrading of the TOPS20 NIFTP to the 1980 specification.

(2) Optionally (but preferably) the interfacing of the NIFTP to a Yellow Book TCP Service.

(3) The design and development of a mail relay system analogous to MMDF for TOPS20. Even if it is decided that UCL will go the NIFTP path, such a relay may well be developed by a US ARPA contractor for MTP support. In this case, our task is to interface the NIFTP channel on TOPS20 to this relay.

(8) There may arise interest in providing mail servers within the UK community, eg on SRCNet or PSS. Such services are more likely to be NIFTP-based than MTP-based (though in the longer term Teletex is a more favoured candidate than either). In this case the following UCL activities would be required (2nd/3rd quarter to 4th quarter and beyond):

(1) NIFTP-based mail channel over Yellow Book over X25.

(2) The use of the UCL MMDF server as an actual mail relay, if Catenet sites are using an MTP channel.

(3) A TCP/X25 virtual call convertor if they are not.

(9) Additionally, UCL will continue to take an active interest in message standardisation activity, in areas such as Teletex, IFIP WG6.5, etc.