

# A New Electronic Mail System

Karl Auerbach

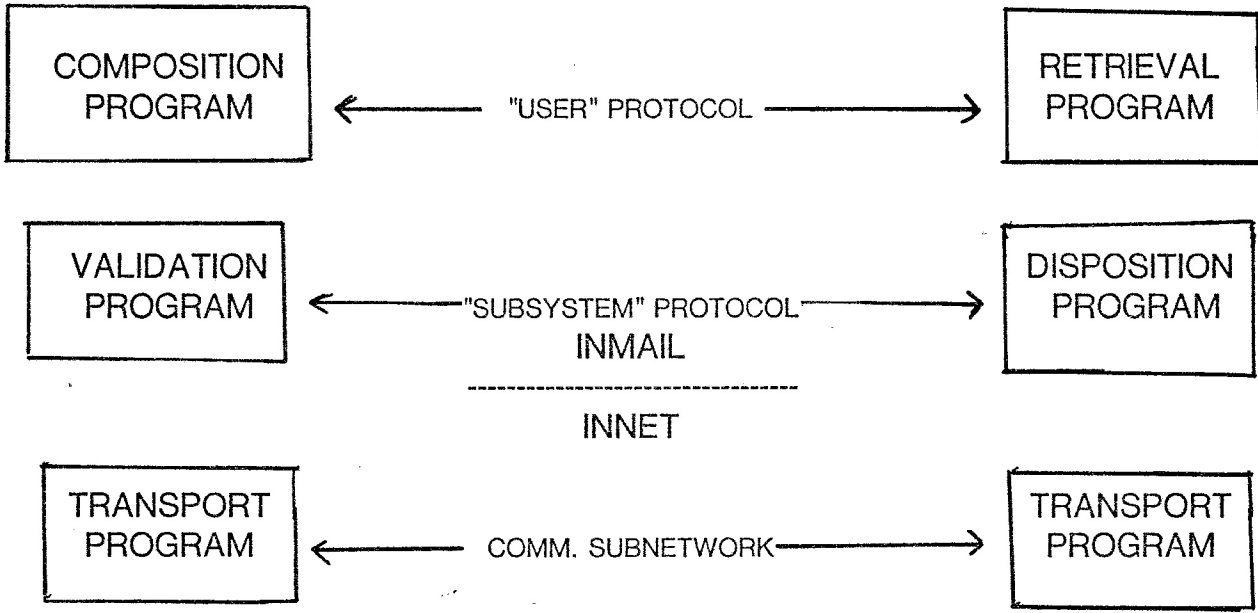
INTERACTIVE Systems Corporation

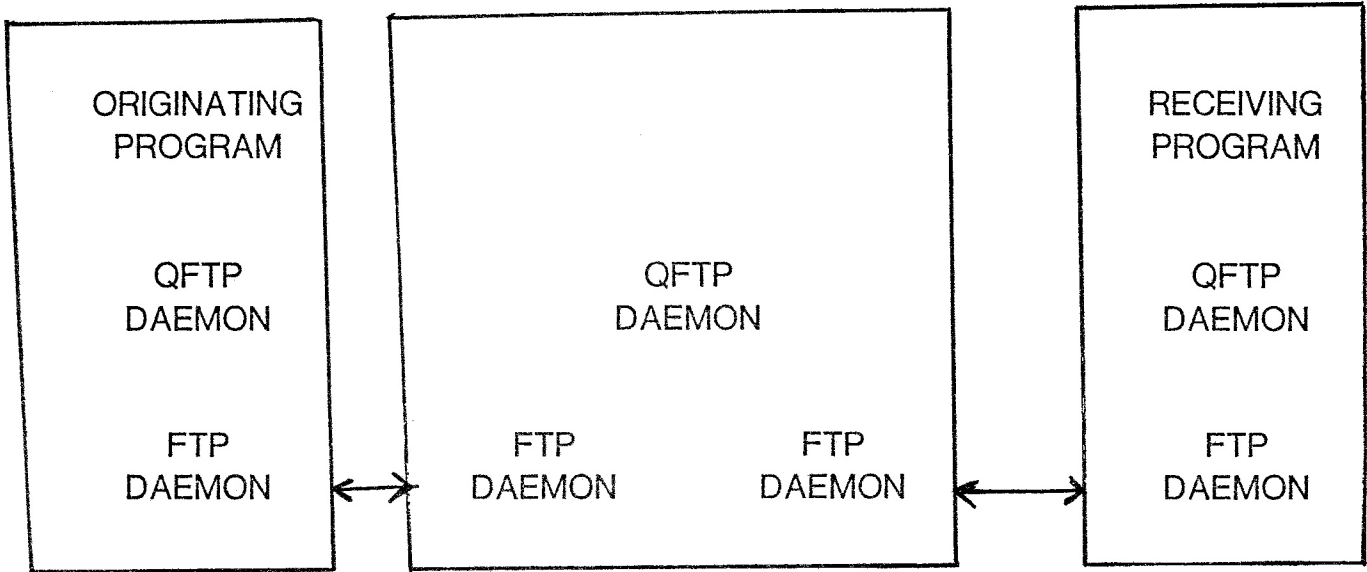
1212 Seventh Street

Santa Monica, California, 90401

(213) 450-8363

- Network operation using a wide variety of telecommunications media.
- May be configured to meet local or personal needs.
- May be supplemented by local software.
- Potential for growth into non-mail applications.
- Easily ported onto new computers and operating systems.





ALIAS NAME	OWNER	GROUP	PROGRAM	PARAMETERS
Abe	lincoln ntl-arcv	pres congr	postmail redistrib	/c/lincoln/.mail "National Archives at Washington"
Alaska	lincoln lincoln  seward lincoln	pres pres  cabinet pres	postmail sendmail  redistrib /bin/print	/c/lincoln/Alaska.mailbox "to: Abe@SM" "Subject: See Alaska box" "Seward at SM"

## New Mail System Features

- Asynchronous delivery
- Full name addressing
- Alias names for recipients
- Automatic redistribution of mail to recipient's host
- Redefinition of delivery mode
- Delivery confirmation
- Urgency
- Interactive correction of errors and 'return to sender' messages
- "cc" and "blind" copy lists
- "From:" and "Sender:" header lines
- Automatic file copies
- Private mailing lists
- Public mailing lists
- Nesting of mailing lists
- Elimination of duplicate deliveries
- Profile control of composition

# INTERACTIVE Mail Network

- Modified "Star" network.

## COMPUTERS:

- PDP-11/70 (IS/1)
- VAX-11/780 (VMS)
- PDP-11/34 (IS/1)
- PDP-11/60 (IS/1)
- ONYX (ONIX)

## INTERFACES:

- X-25 (4800 bps)
- INcard (50,000+ bps)
- DMC-11 (50,000 bps)
- DH-11 (9600 bps)
- DZ-11 (9600 bps)
- Ethernet-like local network (4,000,000 bps)

## SLIDE #1

1. We have recently redesigned and reimplemented INmail, our electronic mail system product, ~~INmail~~.
2. The new system presently operates on a number of different types of computers and operating systems, including Unix.
3. We have retained the basic characteristics of the older system's user interface but have completely revised the internal message transport and delivery mechanisms.
4. Message traffic is handled by a general purpose network transaction system known as the Queued File Transfer Protocol, or QFTP.
5. The system has been divided into a number of discreet components with well defined and documented interfaces. The system can accommodate a multiplicity of user interface tools.

## SLIDE #2

1. A number of constraints were placed upon the new system. Some of these are listed in this slide.

### NETWORK OPERATION

- a) The system must operate in a network without imposing knowledge of network structure or operation upon the user.
- b) Not everyone is using local networks or high bandwidth telecommunication links.

Dial-up is still a major, if not primary, mode of interconnection.

- c) The system must operate even if one or more links on a path between a source and destination are temporarily non-operative.

This constraint precludes any form of real-time end-to-end protocols.

Consequently, the transport mechanism operates like a bucket-brigade moving messages stepwise from host to host.

### CONFIGURABILITY

- a) The system must provide users and system managers with a limited ability to define how the standard mail composition and retrieval tools interact with users.

### SUPPLEMENTATION WITH LOCAL SOFTWARE

- a) The system must allow system managers and users to define alternate mail generation and handling tools.

From our point of view, this facility allows testing of new versions of user interface tools without disrupting normal users.

Our support group has implemented a set of delivery programs which index and catalogue incoming mail.

### GROWTH INTO NON-MAIL APPLICATIONS



- a) A mechanism which moves mail between a composition program on one machine and a mailbox on another can be considered as a special case of a more general file transfer system.

Thus, the system should be structured as a general file transfer system having the mail system as but one of many potential clients.

## PORTABILITY

- a) The system must be structured so that it may easily be moved onto new computers and operating systems, some of which may bear only a passing resemblance to Unix.

## SLIDE #3

1. This diagram describes the protocol layer relationship of the various pieces of the mail system.
2. The underlying file transport mechanism is known as INnet.

It's job is to relay a file through a communications subnetwork, perhaps through intermediary systems running INnet, to a destination.

3. One of the subsystems using INnet is INmail.

- a) INmail may be considered as two layers of protocol.
- b) At the top is the human-to-human communication.

The sender is aided by an agent program known as a composition program. (e.g. 'to')

A composition program is a collection of prompts, formatters, and editors which help the user construct a message.

The recipient is aided by an agent program known as a retrieval program. (e.g. 'msg')

A retrieval program is a simple database access system allowing a user to examine and dispose of received messages.

A retrieval program may become, or invoke, a composition program in order to allow a user to answer or forward a message.

- c) The mail subsystem protocol supports the user-to-user communication.

The subsystem protocol carries the message from composition programs to the databases used by the retrieval programs.

A validation program checks a message for consistency with format and content requirements. (e.g. 'sendmail')

For example, a message header must contain a line properly naming the sender.

Validation programs are given the privilege to introduce the message into INnet.

Thus, no mail may be sent which does not pass through a validation program.

A disposition program writes a received message into a received mail database.

Disposition programs are invoked by INnet on the basis of a table lookup using the addressee as an index.

Sequences of disposition programs may be defined to handle a single delivery. This will be described in more detail later.

A number of standard disposition programs are supplied.

However, many Unix commands or user programs may be used.

#### SLIDE #4

1. This slide describes the structure of processes used in an INnet file transfer.
2. The Originating program is some program which initiates transmission.  
It corresponds to an INmail validation program.
3. The receiving program is some program, or set of programs, to be used to effectuate delivery.  
It corresponds to an INmail disposition program.
4. There is one QFTP daemon per computer system.  
This daemon handles sorting, relaying, and delivery program invocation functions.
5. There is one FTP daemon for each port onto a communications link.
6. TWX, TELEX, TELEMAIL, and so forth are usually connected using a gateway program operating either in the role of a delivery or validation program or as an FTP daemon.

#### SLIDE #5

1. This slide describes how delivery programs are found and used when a file reaches its destination.
2. The QFTP daemon uses a file, /etc/alias, to map addressee names into a sequence of disposition program invocations.
3. For efficiency /etc/alias is structured as an indexed sequential file.
- 4.

#### SLIDE #6

1. This slide lists some of the more important features found in the new mail system.

#### SLIDE #7

1. The INTERACTIVE mail network consists of a half dozen computers split between Santa Monica, California and Gaithersburg, Maryland.

2.- some of the arms of the star are a chain of two or more computers.

## A New Electronic Mail System\*

### ABSTRACT

A new electronic mail system has been designed and implemented. It is constructed upon a general mechanism for transporting files through a network for delivery to an arbitrary set of programs. As such, the system is capable of operating as more than an electronic mail system. The mail system permits transparent distribution of mail across host boundaries. In addition the mail system provides for delivery confirmations, hierarchical structuring of distribution lists, full name or title addressing, user profiles, and priority transmission.

## CONTENTS

1.	GOALS.....	1
2.	ARCHITECTURE.....	2
2.1	INnet.....	2
2.2	INmail.....	3
2.2.1	Composition.....	4
2.2.2	Validation.....	4
2.2.3	Disposition.....	4
2.2.4	Retrieval.....	4
3.	IMPLEMENTATION.....	5
3.1	INnet.....	5
3.1.1	An Example.....	5
3.1.2	The QFTP Daemons.....	8
3.1.3	The FTP Daemons.....	9
3.1.4	The Alias Table.....	9
3.1.5	Alias Programs.....	10
3.1.6	Protection Issues.....	11
3.2	INmail.....	11
3.2.1	Mail Transmission.....	11
3.2.2	Mail Reception.....	12
3.2.2.1	Disposition.....	12
3.2.2.2	Retrieval.....	13

## A New Electronic Mail System\*

Interactive System Corporation's IS/1 operating system, based on Unix\*\*, supports an electronic mail system known as 'INmail'. INmail has descended from various ARPAnet mail programs.

INmail has recently been revised and a new set of electronic mail tools has been conceived and implemented. This paper presents the conceptual overview of this new system.

INmail and the underlying file transport system, INnet, are presently in operation under INTERACTIVE's IS/1 and DEC's VMS\*\* operating systems.

### 1. GOALS

INmail and INnet are designed to allow communication between users on geographically separated computers. The interconnection may span the range of technology from low speed asynchronous lines to high-bandwidth local networks or intercontinental satellite links.

INmail and INnet provide users and system managers with a set of mechanisms to adopt message composition and disposition procedures most in keeping with their organizational or personal preferences. INmail allows, for example, a centralized repository of all messages generated by or addressed to a specified class of users (e.g., members of the purchasing department) while all other traffic is delivered into personal mailboxes.

No single message composition or disposition tools can satisfy all users or meet all unforeseeable future needs. INmail and INnet are amenable to local supplementation by additional software.

INnet provides a framework for the construction of transaction systems. Accordingly, INnet treats INmail as a transaction system in which message composition is the generation of a transaction request and message delivery is the invocation of a transaction processing agent.

---

\* Copyright 1981 by INTERACTIVE Systems Corporation. All rights reserved.

\*\* UNIX is a trademark of Bell Laboratories. DEC and VMS are trademarks of the Digital Equipment Corporation.

## 2. ARCHITECTURE

INmail is but one application constructed upon a more general network communication system called INnet.

### 2.1 INnet

In essence, INnet's job is to carry a file across a network, relaying the file as necessary, until the file reaches a destination host computer. At the destination host, INnet delivers the file by invoking a sequence of programs as defined by a table indexed by the addressee name.

The mode of interconnection is premised upon relatively low bandwidth, intermittently operative telecommunications links. However, almost any kind of link may be used. Because of the unpredictability of network delays, INnet exercises flow and error control on a link-by-link rather than end-to-end basis. End-to-end controls are left to higher level protocols.

INnet does not support terminal or interactive traffic. Even with this restriction many highly desirable functions are possible -- network mail, remote file printing, spooling of word processing output to phototypesetters, background file fetching, and so forth.

INnet carries sealed boxes of information through the network. INnet cares nothing about the contents of those boxes. It is possible, thereby, to construct a number of distinct sub-systems (e.g. electronic mail, remote printing, purchase order generation, etc.) sharing the same underlying transmission mechanism but using disjoint sets of INnet addresses. A facility is provided for placing a type notation on each file. Type stamping reduces confusion caused by misaddressed files.

INnet is an abstract machine upon which various subsystems are constructed. Each subsystem, in turn, may be considered an abstract machine in which users perform productive work. This hierarchy of abstract machines forms a layered system of network protocols.

A program using INnet places both the file to be transported and a leader into a transmission queue. The leader contains addresses and other information used by INnet. The INnet transmission queues are protected and may be updated only by programs having appropriate privilege.

Addresses in INnet consist of the tuple: (host, alias name). Host names must be unique. An alias name has meaning only in the context of a given host.

When no direct path exists to the destination, INnet will relay the file through intermediate hosts. When a leader indicates multiple addressees on separate hosts, INnet attempts to avoid replicating the

file into separate units. The heuristics used provide a reasonable, but not always optimal, conservation of bandwidth.

Upon reaching the addressed host, INnet uses a table to map the alias name into a sequence of programs to be invoked to effectuate delivery. Each program in the sequence is executed until the sequence is exhausted or one of the programs terminates abnormally.

Much of the power of INnet lies in this ability to define programs to be used to effectuate delivery. Delivery to alias 'A' may be performed by a standardmail delivery routine, while delivery to alias 'B' may be performed by the invocation of a complex sequence of user-supplied programs. There are relatively few constraints on what programs may be used.

INnet may find it impossible to relay a file, find an applicable sequence of delivery programs, or to completely execute such a sequence. In that case, failure recovery options specified in the leader provide a means to return a notice of failure to the sender.

## 2.2 INmail

INmail is split into a transmission half and a reception half. Transmission consists of composition and validation. Reception consists of disposition and retrieval. Composition and retrieval contain the interfaces visible to users. Validation and disposition are visible only to the system manager.

Figure 1 shows the relationship of INmail to INnet.

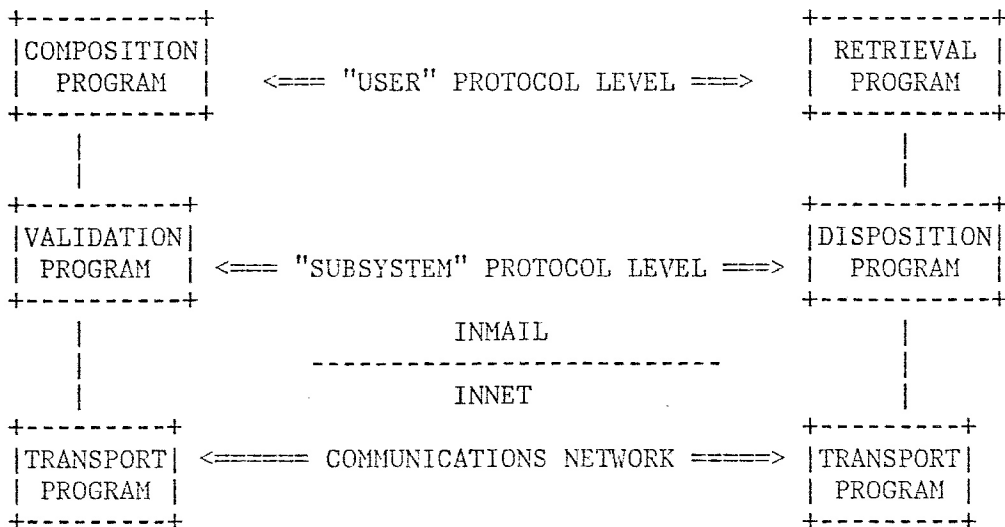


FIGURE 1.

The 'user protocol level' of INmail is the human-to-human communication while the 'subsystem protocol level' is the collection of



agreements and conventions followed by the validation programs to transfer information about the message to the disposition programs. This information includes, for example, requests for confirmations.

#### 2.2.1 Composition

A message is composed through the use of one or more message composition programs. These programs generally consist of a collection of prompters, editors, and formatters. In addition to allowing the user to generate a body of text, composition allows the user to say for whom the text is intended and how it is to be handled.

#### 2.2.2 Validation

Validation checks whether a message meets certain format and content criteria. If the message passes the checks, the validation program constructs a message leader containing (addressee name, addressee host) tuples. The message and leader are placed into a queue for transmission.

The INmail validation program is privileged to write into the transmission queue and is trusted to exercise that power only when appropriate. Composition programs are neither privileged nor trusted and thus may cause a message transfer only indirectly through the protection domain of the validation program.

#### 2.2.3 Disposition

Mail disposition begins once INnet moves a file to the destination computer. INnet maps the alias name (the name of the addressee) into sequential activation of disposition programs. These programs are called with parameters derived from information placed into the file leader by the validation program and from information found in the alias name-to-program map.

INmail provides programs to post a message into a user's mailbox, redistribute a message to other users, or send a message to a printer.

#### 2.2.4 Retrieval

Retrieval consists of those programs used to examine mail after it has been delivered. These programs allow the user to retrieve messages by name or by some selection criteria. Messages may be deleted or aggregated with other messages having similar characteristics or relating to the same matter. The composition tools may be activated to answer or forward a message.

### 3. IMPLEMENTATION

#### 3.1 INnet

INnet implements a protocol called QFTP, Queued File Transfer Protocol. (QFTP).

QFTP consists of a number of queues and a number of ever-running "daemon" programs. The function of the queues is to isolate programs which operate asynchronously from one another. In general, queues serve as interfaces between protocol layers.

QFTP has the job of taking a file from a transmission queue on one machine and relaying that file from one machine to another until that file reaches the target machine. QFTP will then invoke one or more programs to dispose of the file. QFTP uses the services of a file transfer protocol (FTP) to drive the physical transfer of files over individual telecommunications links.

Figure 2 depicts the relationship of QFTP and FTP.

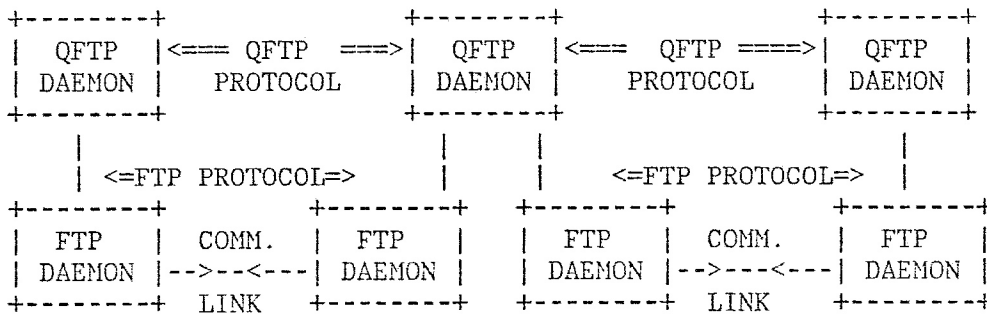


FIGURE 2

##### 3.1.1 An Example

For purposes of illustration, assume that there is a validation program named sendfile on computer C1. Computer C1 is connected to computer C2 and C2, in turn, is connected to C3.

Sendfile constructs a transmission queue entry for the file. The leader of the entry contains the following information:

```

<userid><return addr>
<alias><urgency><address><alias data>
<alias><urgency><address><alias data>
.
:
<alias><urgency><address><alias data>
<failure spec>
<type stamp>
<file ptr>

```

These fields have the following general meaning:

- <userid> The name of the user who called upon the validation program.
- <return addr> The network address (or network routing path) necessary to reach from 'here' to the computer on which the file originated. 'Here' is dynamic, varying as the file moves stepwise from its source to its destination.
- <alias> The name of an alias purported to exist on a destination computer.
- <urgency> A specification of how fast this file should be delivered to the associated alias. The maximum of all the urgency fields in a queue entry file is encoded into the file name making it possible to scan a queue for priority messages without opening the individual files.
- <address> The network path from 'here' to the computer on which the associated alias is purported to exist.
- <alias data> Information which will be available to the various programs invoked under <alias>. This field allows the sender of a file to indicate to the alias programs that there is something out-of-the-ordinary about this file or recipient. For example, mail confirmation parameters are encoded into this field.
- <failure spec> Defines to the QFTP daemons what actions they should take when it becomes impossible to further move the file towards its destination.
- <type stamp> Used to pass information to all of the alias programs which will be called to dispose of the file. Programs having the privilege to place entries in the QFTP transmission queue are trusted to properly identify the type of file they are sending. The disposition programs executed under an alias name are expected to use this type of information to ascertain whether they may accept the file. This facility allows a separation of traffic by type. This may be used to prevent, for example, the posting of anything but files typed as 'mail' into mailboxes.
- <file ptr> A pointer to the actual file which the user intended to transfer.

Transmission queue entries are merely files stored in a known directory.

The QFTP daemon on C1 periodically examines the transmission queue on C1. The daemon will eventually find the queue entry placed there by sendfile. The QFTP daemon sorts the list of target and network addresses. Where the target is local, on computer C1, the QFTP daemon performs a delivery function. Delivery will be described later. For all other targets, the QFTP daemon collects those with a common next-transmission-link, updates <return addr> and <address>, and creates new queue entries in the send queues of the appropriate network FTP daemons.

The file now exists as a queue entry in the send queue for the FTP daemon connecting C1 and C2.

Periodically the FTP daemon examines the send queue. When something is found, a file transfer sequence is initiated over the communication link. This sequence results in an exact image of the queue entry (including the data in the original file) being placed in a queue on C2.

The QFTP daemon on C2 eventually comes across this entry. This daemon examines the entry for local targets and performs necessary sorting functions. A queue entry is made in the send queue of the FTP daemon for the C2-C3 link.

The file will eventually arrive at C3. The C3 QFTP daemon examines the entry and finds that the target is local. The QFTP daemon then looks up the target name in an "alias" table.

The alias table maps target names into a sequence of programs to be invoked. Thus, when the file arrives at C3, one or more programs may be called to process the file. Suppose the alias to which the file is being sent is "Alaska". The alias table might contain the following entries:

<u>alias</u>	<u>name</u>	<u>owner</u>	<u>group</u>	<u>program</u>	<u>parameters</u>
	Abe	lincoln	pres	postmail	/c/lincoln/.mail
		ntl_arcv	cong	redistrib	National Archives at Washington
	Alaska	lincoln	pres	postmail	/c/lincoln/Alaska.mailbox
		lincoln	pres	sendmail	"to: Abe@SM" "Subject: See Alaska box"
		seward	cabinet	redistrib	seward at SM
		lincoln	pres	/bin/print	-fi

The QFTP daemon at C3 sequentially initiates the programs postmail, sendmail, redistrib, and /bin/print with their associated parameter lists. Each program is executed in the protection domain defined by the associated owner and group. The QFTP daemon establishes input

stream for each program such that each program may read the entire portion of the file.

In this example, the file is posted to the mailbox known as `/c/lincoln/Alaska.mailbox`. Sendmail is then invoked to send a notice to something known as `'Abe@SM'`. A copy is forwarded to `'seward at SM'`. Finally, a copy of the file is printed on C3's line printer.

To prevent arbitrary files from being posted to mailboxes, 'postmail' accepts only those files having as their <type stamp> the type `'INmail'`. Files with other type stamps are rejected by the various mail delivery programs. The QFTP daemon will abort a sequence if the completion value of any program indicates an error.

After the sequence of programs is completed, the QFTP daemon removes the file.

### 3.1.2 The QFTP Daemons

When the daemon awakens, it begins a scan of a directory for the first file to process.

After selection, the queue entry is opened, the address information is scanned, and a sort is performed to create clusters of addressees having a common 'next site' in the network path. This minimizes redundant transmission of the message text. Message splitting is deferred until the transmission paths differ.

After the QFTP daemon sorts the recipients by common next link, the maximum <urgency> is ascertained for each link. This information is then passed to the FTP daemon not as part of the file leader but rather as part of the file name used to make the queue entry. Thus an urgent message may have the queue entry for the FTP daemon named `'VAX.5x6'` where the final character specifies the priority.

When an addressee is local the daemon reads the local alias file to find an alias matching the addressee name. Assuming that one is found, the daemon serially invokes the programs indicated by the alias entry with the appropriate user and group identifications and with the standard input being the text of the file which was transferred. The alias file specifies what information is to be passed as parameters.

When a QFTP daemon cannot forward a message for any reason or when no alias can be found to dispose of the message at the target computer, the QFTP daemon makes use of the <failure spec> in the queue entry. This field is set by the validation programs. The options available include:

1. Throw everything away without notification.
2. Generate a short descriptive INmail message addressed to a specified alias (usually the originator).

3. Repackage the file as an INmail message and return it to a specified alias (usually the originator).

### 3.1.3 The FTP Daemons

Each network daemon has two jobs. First to drain the queue of pending outgoing files. Second to receive incoming files and place them into the QFTP daemon's transmission queue. There are different FTP daemons for different types of network links.

FTP daemons have a rather simple service specification:

1. A file placed in a given directory on machine A is moved to a given directory on machine B.
2. File contents are preserved.

This specification may be met by something as primitive as a physical transfer of files on magnetic tape. Indeed, if one desired, low priority files could be spooled to tape and then the tape shipped by an over-night freight carrier.

The existing FTP daemons additionally contain a priority negotiation sub-protocol.

### 3.1.4 The Alias Table

The alias table contains the information necessary to transform an alias name into a sequence of programs activations. The table contains one entry per alias. The entries take the following form:

```
<alias name> <alias owner> <alias group>
<program entry>
<program entry>
:
```

There may be zero or more <program entries>. Each defines a program to be activated. A program entry has one of two forms:

```
[<aliasname>
OR
[<pathname> <owner> <group><directory><qparm><parameters...>
```

The first form is a direct reference to an executable program. <pathame> is simply a file pathname to an executable program load module. <owner> and <group> define the protection domain in which the program is to run. <directory> establishes the working directory. <qparm> indicates whether the QFTP daemon is to generate a set of parameter strings based upon information known to the daemon. <parameters> are character strings to be passed directly to the alias program.

<qparm> information includes all information from the queue entry relevant to the alias being used. This information includes the alias name, <alias data>, the <type stamp>, <urgency>, sender name, sender address, and so forth.

This parameter scheme allows many existing and special purpose programs to be used under aliases.

The second <program entry> form is an indirect reference to another alias table entry. The program sequence defined by that other alias will be used. The current sequence will be resumed upon completion of the indirectly referenced sequence.

A sequence of <program entries> is aborted when a program terminates with an error. An error in a sub-sequence affects only that sub-sequence.

The alias table is maintained as an indexed text file. There is one alias table per network host. Inconsistencies between these files may result in lost, misdelivered, or looping traffic.

A distribution list is an alias file entry containing multiple indirect references or calls to a redistribution program. When a file is addressed to a distribution list, delivery is performed as many times and in as many ways as defined in the distribution list. A distribution list is referenced exactly as any other addressee. Distribution lists may reference other distribution lists, either local or remote. In this manner a hierarchical structure may be constructed paralleling an organizational hierarchy. At INTERACTIVE we have established a corporate distribution list which references a distribution list for each functional organization. Each organization list, in turn, references the lists for its component sub-organizations. Eventually lists are reached which directly reference mailboxes of the various corporate officers and employees.

### 3.1.5 Alias Programs

Any program may serve as an alias program. However, since an alias program is essentially a universally callable program which runs in a protection domain separate from that of its caller, alias creation should not be taken lightly. Large security holes would be created if general purpose programs were declared as alias programs.

Those alias programs supplied with the INmail package use the <sender stamp> field of messages to ensure that they are operating only on files approved by a trusted validation program.

Users can provide protection for their own alias programs by creating trusted validation programs which place their <sender stamp> only upon those files meeting their acceptance criteria and by creating alias programs which perform interpretive access checks.

When an alias program is invoked, its standard input is taken from the message file and its standard output directed to a null device.

Alias programs may double as validation programs. If a relay mechanism is desired (e.g. forwarding of mail to a user temporarily at another address.) the alias program may turn around and generate a new queue entry in the transmission queue.

### 3.1.6 Protection Issues

Protection depends upon the information in the alias table being correct and upon the proper action of the validation programs, the daemons, and the alias programs. System managers who create or modify any of these can expose the entire system's, or at least a single user's, files to improper access.

The alias table (actually alias file) contains considerable protection-related information. The alias file must be protected from improper modification.

## 3.2 INmail

### 3.2.1 Mail Transmission

Mail transmission covers both composition and message validation. The actual transfer operation is performed by the QFTP transport mechanism of INnet.

The to program perform both composition and validation functions. A separate sendmail program performs only validation functions. Sendmail is intended to be used by user created composition programs.

to leads the user through a composition session, validates the message, and places the message in the QFTP transmission queue. The syntax for specifying addressee lists is best described by an example:

To: Abe Lincoln (confirm, urgent) at Illinois

In this example, the sender has indicated that the message is addressed to 'Abe Lincoln'. The sender is to receive a confirmation of delivery and the message is to be given priority treatment during transmission. An explicit host name, 'Illinois', is given. In the absence of a host name, to would have checked whether 'Abe Lincoln' is known in the local alias file.

Addressee names may contain blanks. As a consequence, addressee specifications must be separated by commas.

The to program has a macro facility allowing users to specify private mailing lists. When a user gives the name of a private mailing list, the contents of that list is expanded into the message header. A private mailing list may reference any local or remote addressees or



any other private mailing list. If other private mailing lists are referenced, they are expanded into the message header. Expansion of private mailing lists allows recipients to reply without finding and reading the original private mailing list.

Public mailing lists are maintained as distribution lists within alias files. Public mailing lists are always accessible and may be referenced remotely, obviating the need to expand their contents into message headers.

INmail allows users to select, through a 'profile', a limited set of composition options. These options define the verbosity of the to program, default contents for certain message header fields, choice of editor, and whether automatic file copies are to be made.

File copies, unlike normal copies of messages, contain otherwise hidden information such as the 'blind copies' list. There are a number of automatic file copy options:

1. Don't ask whether to make a copy, and don't make one.
2. Don't ask whether to make a copy, but make one in a defined mailbox.
3. Ask whether to make a copy. If the response is positive, make a copy in a defined mailbox.
4. Ask whether a copy is to be made and where it is to be placed.

Message validation consists of checking the existence of all required fields, stamping the message header with a true date and sender identification, and checking for validity of all local names and distribution lists. The to program will prompt the user to correct errors.

### 3.2.2 Mail Reception

Mail reception covers both delivery disposition and subsequent retrieval.

#### 3.2.2.1 Disposition

INmail provides three alias programs to support mail disposition. These programs check the <sender stamp> placed by the validation program:

<u>alias name</u>	<u>parameters</u>	<u>description</u>
postmail	mailbox-name	Extracts mail header and text from file and posts to mailbox Postmail examines the <alias data> parameter and if necessary generates delivery confirmations.
redistrib	addresses	Redistributes mail to specified addresses. No additional marks are placed on the mail.
printfile	queue name	Uses the IS/1 Queuing system to send the file to a line printer or high quality output device.

#### 3.2.2.2 Retrieval

The msg program performs message retrieval. This program allows a user to select and view messages, to move messages into repository files, and to answer or forward a particular message.

## A New Electronic Mail System\*

### ABSTRACT

A new electronic mail system has been designed and implemented. It is constructed upon a general mechanism for transporting files through a network for delivery to an arbitrary set of programs. As such, the system is capable of operating as more than an electronic mail system. The mail system permits transparent distribution of mail across host boundaries. In addition the mail system provides for delivery confirmations, hierarchical structuring of distribution lists, full name or title addressing, user profiles, and priority transmission.

## CONTENTS

1.	GOALS.....	1
2.	ARCHITECTURE.....	2
2.1	INnet.....	2
2.2	INmail.....	3
2.2.1	Composition.....	4
2.2.2	Validation.....	4
2.2.3	Disposition.....	4
2.2.4	Retrieval.....	4
3.	IMPLEMENTATION.....	5
3.1	INnet.....	5
3.1.1	An Example.....	5
3.1.2	The QFTP Daemons.....	8
3.1.3	The FTP Daemons.....	9
3.1.4	The Alias Table.....	9
3.1.5	Alias Programs.....	10
3.1.6	Protection Issues.....	11
3.2	INmail.....	11
3.2.1	Mail Transmission.....	11
3.2.2	Mail Reception.....	12
3.2.2.1	Disposition.....	12
3.2.2.2	Retrieval.....	13