Title
Achieving interoperability between two domains : connecting the ZOTnet and UUCP computer mail networks

Author
Rose, Marshall

Publication Date
1983-01-26

Peer reviewed

Achieving Interoperability between two Domains – Connecting the ZOTnet and UUCP Computer Mail Networks

Marshall Rose
Department of Information and Computer Science
University of California, Irvine
Wed Jan 26 19:17:47 1983

Computer Mail: mrose%uci@Rand-Relay

Technical Report 201

Introduction

This paper describes the motivation for and implementation of a mode of interconnection achieved between two computer network mail domains: the U.C. Irvine ZOTnet[1] domain and the UUCP domain. It is our intention to examine the problems that occur in establishing interoperability between two domains with very different philosophies, architectures, and underlying protocols. After the issues raised by this problem have been examined, some attention to practical details will be given. Finally, aspects of this problem which have not yet been satisfactorily dealt with shall be discussed.

The organization of this paper is straightforward. Initially, we begin with a terse discussion of the general CBMS model. This is followed with a description of interaction between the two domains prior to interconnection. This will include some background information on the two computer mail networks. In particular, we shall focus on some of the problems caused by a lack of interconnection between the two domains.

After presenting the motivation for the problem, we shall consider how protocols function in a computer mail network. The ISO/OSI model will be used as a framing point to present a slightly different analysis of layers and peers in the computer mail network model. While considering the different protocols, a great emphasis will be put on the user agent to user agent protocol, with a somewhat lesser emphasis on the service agent to service agent protocols.

Next, we shall give much attention to the issues of interconnecting different domains. Traditionally, two methods of interconnecting networks are used: protocol translation and protocol encapsulation. The former technique is used in linking the ZOTnet and UUCP domains. Unfortunately, protocol translation is very difficult, and poorly understood. We shall see how these general issues affect our particular problem, and how the datagram-like nature of the two networks serves to make the task easier.

While examining these issues, we shall also consider how the concept of names, addresses, and routes are used in the two domains, and their effect on the problem. Addresses and routes are handled in completely different ways in the two networks, so the differences shall make a significant impact on the solution.

Once we have seen the motivation of the problem, and examined its basis, one method of its solution will be attempted. Our efforts will be focused primarily on how addressing is handled in the protocol translation. Again, we shall see how the vastly different addressing scheme is a key aspect. Routing mechanisms are also significant, but will be of secondary concern.

[1] The name ZOTnet was derived from the U.C. Irvine slogan. The official mascot is the Anteater, which reportedly makes a ZOT sound. Hence, ZOTnet, for UCI's first computer mail network.

Finally, the success and results of the interconnection will be examined. We shall see how well the solution has been able to meet the problem, and what has been gained in the process. Ways in which the solution is incomplete will be discussed, along with possible extensions that could make the solution more complete.

Background

One model for a Computer Based Message System (CBMS), has been presented by the IFIP Working Group 6.5 (see [NBS81] for one interpretation. [X-MHS1] has a more complicated viewpoint). In this model, entities are given the capability to send messages to another, under a specific set of restrictions. Entities interact with a user agent (UA), instead of engaging in direct communication. The UAs in turn do not directly communicate, and rely on a message transfer system (MTS) to perform the actual transport. The MTS is composed of individual message service agents (MSAs), which serve a given locality (say a host). Collectively, the MSAs and the UAs form a message handling system (MHS) Within the model, certain boundaries of responsibility and possession exist. In particular, when placing a message into the MTS from a UA, the message crosses a posting slot, in which the responsibility for delivery of the message (and possession of the message) transfer from the UA to the local MSA. Similarly, when a message is delivered from the MTS to the UA, the message crosses a delivery slot with the same type of responsibility and possession semantics. Messages within the MTS are divided into two parts: the envelope and the body. The former contains information to help the MTS deliver the message to its intended addressees, the latter is the actual message itself.

Environs of the two Domains

Let us begin by discussing the environs of the two computer mail domains prior to their interconnection. We shall consider the two domains separately, and discuss the services that each offers to its users. Following this, we shall consider how a lack of interconnection affected users of the two domains.

The ZOTnet, a local area mailing network, is described in some detail in [ZOTnet]. The ZOTnet currently serves a research environment for the faculty and graduate students of the Department of Computer Science at the University of California, Irvine. Four machines currently participate in the ZOTnet domain. Two are DECsystem-2020s running release 4 of the TOPS-20 operating system. The other two are VAX-11/750s[1], which are running 4.1bsd UNIX[2].

[ZOTnet] identifies the various protocols employed by the UAs and MSAs in the MHS. Of these, two will be of particular importance in this paper: the first is the application layer protocol, which is used for the formatting of messages, and tends to be understood by both UAs and MSAs, and the second is the message layer protocol, which is used between adjacent MSAs communicating in the MTS. For the application layer, the ZOTnet presently adheres to the Internet standard described in [RFC822], i.e., the Standard for the Format of ARPA Internet Text Messages.

---

[1] VAX is a trademark of Digital Equipment Corporation.

[2] UNIX is a trademark of Bell Laboratories.

The two DECsystem–2020 machines have a single MSA on each, which supports the ZOTnet. One of the VAX–11/750s acts as the ZOTnet gateway. It polls the two DECsystem–2020s and the other VAX frequently, in order to exchange mail. The MSA software used on the gateway is an enhanced version of the University of Delaware's Multi-Channel Memo Distribution Facility (MMDF) system [CSnet–DN3], which forms the basis of the NSF's CSnet PhoneNet project. MMDF may best be viewed as a store–and–forward channel switch, in which a collection of channels interface between given foreign protocol suites and the internal MMDF protocols. The internal MMDF message layer protocol, which is used at one end of the channel to accomplish message transfers, is called the submit protocol, for lack of a better name. Since MMDF can support a wide variety of interconnections to other message transfer layer protocols, it is the natural choice for being used on the ZOTnet gateway.

The UNIX to UNIX Copy (UUCP) network is implemented on computers running the UNIX operating system[UUCP–1]. The UUCP network achieves command execution, file transfer, and mail communication for its users. A technical discussion of the network is presented in [UUCP–2]. What is particularly interesting about the UUCP domain is the fact that it is quite large, and has a topology that is not a connected graph. That is, UUCP exists as several networks, but not all of these networks are connected together.

As the name implies, UUCP is strictly for UNIX systems. Hence, it is easy to see why the two VAX–11/750s participate in the UUCP domain, and why the two DECsystem–2020s are excluded. To compound matters, for quite some time after its installation, the second VAX–11/750 was connected to the ZOTnet gateway by UUCP, but was not in the ZOTnet domain. Hence the ZOTnet gateway had two separate computer network mail systems that did not interoperate! We now have a sufficient framing to see what sort of interaction took place between the ZOTnet and UUCP domains prior to the establishment of interoperability.

Interaction prior to Interconnection

In examining the lack of interconnection, let us examine three problems: The first problem is that messages for the two networks have two completely different formats. As a consequence, each user is required to have two separate maildrops. Because messages from both domains can not coexist in the same maildrop (or mail archive file), each user must now query the status of both. This results in much confusion, particularly when a user is attempting to keep a transcript of all correspondence on a given topic in a single mail file.

The second problem is that each network has a separate UA capable of reading the maildrops. This results in each user being required to deal with two UAs in order to process mail. This is an extraordinary inconvenience, as there is little similarity between any of the UAs for the two domains[1]. In short, users are forced to deal with two mail networks, and the required level of expertise in each is greater than the required expertise one would need to interact in only one of the networks.

Finally, the third problem is that the two networks have no connectivity. This means than when communicating with electronic mail correspondents, the user is required to know not only the address for each correspondent, but the network which serves each correspondent. Further, it is impossible to send the same message to correspondents in different networks. Not

[1] Some UAs for a particular domain are similar to each other, but it is impossible to find a UA in the ZOTnet domain that is similar to one in the UUCP domain.

only does this reduce the value of group communication, but it makes replies to all parties impossible.

These three problems combine to make an excellent case for interconnection between the two networks. Other arguments in favor of interconnection could be raised that have a more philosophical basis, but these three are sufficient to provide the reader with motivation.

Protocols in the Computer Mail Network Model

In [ZOTnet], an argument is presented which views a hierarchy of the computer mail network model as being highly analogous to the model presented by the ISO in their formalization of Open Systems Interconnection. In this section, we shall consider how protocols and protocol translation fit into models of computer network mail.

Although it is not our intention to reproduce the discussion here, we shall summarize one result, which is embodied in Figure 1 (which also can be found in [ZOTnet] as Figure 1). The convention used in drawing this diagram is for vertical lines to represent the services that a layer offers to the one above it, and for horizontal lines to represent the protocol that two peers use to communicate. The term named within a given box is the entity acting as a peer. This convention is followed with the exception that the bottom-most horizontal line is marked with the medium that supports the communication, not the protocol itself.

A detailed explanation and justification of this figure is found in [ZOTnet]. For our perspective, it interesting to note the following:

1. For lack of a better term, natural language is used to denote the protocol used between users of this system as they communicate. This protocol is viewed as being at the application layer.

2. The message format standard is recognized as an important protocol, and is itself placed at the presentation layer. Because the standard specifies the format of structured information, it is easy to see the message format standard as a presentation layer protocol. It is through careful adherence to a single message format standard that different UAs can offer to a wide variety of services to their users.

3. The message transfer protocols reside at the session, transport, and network layers. Individual MSAs must perform tasks found in all three layers in order to relay mail from one site to another.

In view of the roles that protocols take in our model of computer network mail, what can be said of protocol translation in this context? To answer this, consider the information that a message format standard specifies. First is the overall message structure. In most cases, a general memo framework is envisioned. The message is divided into two parts, a set of headers (analogous to a letterhead) and some text. Although no rigid specification is placed on the format of the text part, a very strict syntax and semantics is imposed on the headers part. Traditionally, the headers are viewed as consisting of one or more header-lines, each consisting of a keyword (e.g., "From") and a value. Some of these header-lines have a well-defined structure imposed on their value portions. The contents of these value portions are typically addresses and dates. It is by understanding this protocol that the UA is able to offer its users a set of services. Hence, translating from one domain to another, at the UA protocol level, simply entails translating the message format, in particular the header-lines, so that the same semantic value is present and

```
------------------                          ------------------
| Julian Feldman |     Natural Language     |  David Sheldon  |
|    feldman@UCI |--------------------------| sheldon@UCI-20A |
------------------                          ------------------

      |                                         |
      | User                                    | User
      |  Interface                              |  Interface
      |                                         |

-------------------                         ------------------
|    User Agent   | Internet Message Format |    User Agent   |
|            (MH) |-------------------------|            (MM) |
-------------------   Standard (RFC-822)    ------------------

          |                                         |
          | Posting/                                | Posting/
          |  Receipt                                |  Receipt
          |                                         |

-----------------        -----------  -----------------
|Delivery Service| Submit | Translate| Env. |Delivery Service|
|         (MMDF) |--------| (ZSLAVE) |////////|       (XMAILR) |
-----------------        -----------  -----------------

     |                         |
     | Channel                 |----| Channel
     |  Service                     |  Service
     |                              |

-------------                 -------------
|  Polled   |    PhoneNet     |  Passive   |
| Channel   |-----------------| Channel    |
-------------                 -------------

      |                             |
      | Physical                    | Physical
      |  Circuit                    |  Circuit
      |                             |

-------------                 -------------
| DataSwitch |     RS-232     | DataSwitch |
|            |----------------|            |
-------------                 -------------
```

Figure 1. Interaction in the ZOTnet domain

perceivable in each message (initial and translated) in their respective domains (source and destination). In particular, it is noted that translation of structured value portions (e.g. addresses) will be of great importance, as these are well-defined in most specifications, and well-used in all UAs.

For the application layer (natural language) protocol, which occurs between users, no translation is needed. In most (but not all) cases, it is the text part of the message in which this protocol is used, and since there is (and probably will never be) a formal definition of a user to user (natural language) protocol, translation at this level is highly suspicious at best.

In terms of the message transfer protocol, we are interested primarily in envelope translation. For the life-time of a message in the MTS (i.e., from the time it is accepted through the posting-slot until the time it is accepted through the delivery-slot), each message has associated with it, an envelope. A message envelope[1] contains information about the message which is strictly internal to the MTS. This information usually includes the address of the entity that claims responsibility for the message (e.g., the sender of the message), a time-stamp indicating the time the message entered the current MSA, and a list of addresses that the message is to be delivered to. The responsible entity's address is used to return notifications of failed deliveries to one or more of the envelope's addresses. The time-stamp can be used to indicate when the message should be returned to the responsible entity if it has not been fully processed within a certain time. The other addresses in the envelope are those which have not yet received a copy of this message. Again, it appears that addresses will play an important part in translation.

This important role can not be underestimated. It is vital to understand that addresses *must* always be meaningful in any domain in which they are found (say, by virtue of a message containing those addresses traversing those domains). That is, as a message crosses from the initial to intermediary to final domains, the addresses in that message must make sense in each. This applies to addresses in both the envelope and the headers. Although it is easy to see why the envelope must be manipulated, the reason for changing the headers is not so clear. Recall that the addresses-format valid in the source domain need not be valid in a destination domain, and that the destination domain need not know how to convert addresses from the source domain format to its own format. Hence, each intermediary MSA must ensure that the addresses are correctly formatted.

[1] The difference between the envelope and parts of the headers is critically important. Although some of the information in the envelope can be found in the headers, it is generally impossible to fully reconstruct the envelope solely from information in the headers.

Some Issues in Interconnection

In this section we shall examine some of the major issues involved when two computer mail networks are to be interconnected. In particular we shall focus on protocol translation in a store-and-forward network environment, the relations between names and addresses in different domains, and access control policies.

Translating Protocols

Protocol translation is often a very difficult task. One reason why this is so, is that very often the protocols to be joined have been designed by different parties with no plans of future interconnection. Regardless of the cause, much research remains to be done in this area. Fortunately, in our context, protocol translation between computer mail networks is much simpler than is translation in other contexts. The primary reason for this is the store-and-forward nature of the communication.

The channel-fitting nature of the MMDF system provides a natural setting for developing a channel that can translate between the two protocols. Simply put, a message enters the MMDF queue with an envelope. The envelope identifies a responsible party (usually the sender), a time-stamp, and a number of addressees. For each addressee, MMDF assigns a channel designator based on a table-driven decision. The algorithm used is discussed in [ZOTnet]. This allows us to easily mark messages which have at least one addressee in the UUCP network, and designate those addressees as being served by a UUCP channel. This is sufficient to achieve a method of establishing a link between the two networks at the message transfer layer.

It is not yet sufficient to handle the full protocol translation. In [ZOTnet] the UA protocol is identified as the current ARPA Internet standard. The UUCP network does not have a single official standard for UAs. The standard used is by default[1]. Without an adequate formal specification, such as [RFC822], the translation between the two networks at the UA protocol level can become quite strained. It is indeed fortunate that the by default standard for UUCP UAs is simple and mostly consistent. A more detailed examination of protocol translations will be made when we view names and addresses in the two domains.

[1] That is, whatever UUCP programs generate and interpret forms the standard.

Address Recognition

Now that we have seen how a channel can be used to act as a link between the two networks, let us consider where address recognition should take place. That is, where in the MHS should the detection of an address in a foreign network take place? This question remains a topic for further research.

Some argument might be given that the local MSA should be able to recognize addresses from different networks and internally mark the addresses for processing by the appropriate channel, with the UAs completely isolated from recognizing such addresses. In support of this argument, one should notice that the MSA is the entity that is closest to the interconnection between the networks, and if it can successfully hide the differences between its local network and the foreign network, then both the users and UAs will benefit. Unfortunately, addressing schemes between different networks can be sufficiently different that parsing of foreign addresses will be impossible or ambiguous. As a result, UAs can not easily analyze messages on the user's behalf. One example of this is the reply command found in most UAs. Usually, this command invokes the UA to examine a given message, and construct the initial part of a reply to the parties mentioned in the message by examining and extracting from the contents of certain fields (e.g., "Reply-To:", "To:", and so forth). Because of these problems, it is clear that the MSA can not hide the presence of foreign addresses from UAs or users.

An alternate view would be that the individual UAs should be able to recognize foreign network addresses and specifically tell the MSA that a foreign network is to be addressed. While this does allow UAs to act intelligently when handling foreign addresses, it introduces a new set of problems as UAs on a given host must be customized to handle different networks as they may be interconnected in the future. In any event, the UAs themselves should not be the only entities in the MHS which perform address recognition. This would imply that the UA would post multiple copies of a message with different MSAs, one for each network.

A good solution is that taken by [RFC819]. In short, addresses are viewed as local@domain. The local part is not manipulated by domains outside of the specified domain. This allows each network to use its own special formats within its domain, and still allows UAs and users in the Internet to process mail. In our context, we can see this as a feature with good and bad points — the ZOTnet uses the 822 standard, but the UUCP domain does not adhere (even in spirit) to 822.

Boundaries of Translation

Related to the problems of address recognition, we should consider what kind of translation occurs at the UA protocol level. Recall that each domain views a message as consisting of two parts, the headers and the text. In the ZOTnet domain, fields in the headers part have a very specific syntax and semantics. This discipline does not occur in the UUCP domain.

The term munging is used to denote the translation of the contents of a message's header. It is this munging in the UA layer protocol that we are concerned with. The next section will deal with the specifics of the translation, in this section, we are concerned with other details.

One major problem with the division of a message into a headers and text part is the handling of encrypted messages. Often, in order to ensure privacy of information, users of the UUCP network encode messages using an encryption method. It would be impossible to translate these messages if the headers are encrypted as well as the text. The 822 standard recognizes this problem, and does not support the encryption of the header portion of a message. It is interesting to note that in practice maildrops in the UUCP domain are world readable by default. In short, any user can read the mail of any other user[1]. In contrast, maildrops in the ZOTnet domain are protected so that only the intended user has read access.

Mappings between Domains

When two domains are to be interconnected, one has the circumstance to ask if addresses in different domains should map to the same name on the same machine if a given machine exists in more than one domain. To elaborate on this, consider a problem mentioned above. A user has two maildrops on the same machine, one for each domain. Should the user named Veet Voojagig on a given machine be viewed as having the same name in both domains that the machine participates in? Phrased another way, should the address ucivax!voojagig in one domain be equivalent to Voojagig@UCI in another domain, or should the addresses really refer to separate names?

One can make substantial arguments on either side of this question. Those believing that the addresses should map to the same name can argue — we are referring to the same entity. Those with an opposite viewpoint can easily stress that names, addresses, and domain names are entities unique to each individual domain.

This question is particularly interesting, as it forces us to re-consider our conceptions of names, addresses, and routes, and the relations we see as existing between them. The answer to this question remains a research problem.

Access Control

A final issue to consider is that of access control policies and mechanisms. This issue is both technical and administrative in nature. From the technical side, we can easily see how a channel can be augmented with mechanisms to ensure that only authorized users and sites can use the channel. Since the channel has access to the message's envelope, it should be able to easily identify the party that claims responsibility for the message and the addressees being served by the channel. Based on this information, the channel can decide if access to the channel should be permitted.

From the administrative side, certain domains may wish to restrict other domains from using their MSAs as transport media for messages. This can be particularly true if two domains are attempting to use a third as a intermediary. The current mechanism used in the UUCP channel in ZOTnet is to examine the sender of the message first. If this user is authorized to access the given network, then the channel performs the transmission. Otherwise, if the

---

[1] UNIX even goes so far as to provide a user command that specifically is used to read other people's mail. This policy is greeted with clenched fists, and rightly so, by computer mail users who view the sanctity of the mail as paramount.

destination is authorized to receive mail, then the channel performs the transmission. If neither of these conditions hold, the message is rejected.

Names, Addresses, and Routes

In order to understand how the UA protocol is translated between the ZOTnet to UUCP domains, we must examine the concepts of names, addresses, and routes in both.

Names

In both domains, a name is a reference to an entity. In our context, a name is a user's name, such as Veet Voojagig. Since names are found in both protocols for purely display purposes, no translations of names between the two domains need take place.

Addresses

In the ZOTnet domain, the 822 standard defines the syntax and semantics of an address. In short, an address takes the form local@domain, where local may include information specific to the address' particular domain, and domain is a hierarchical Internet domain specification, often a simple host name. Addresses in the ZOTnet domain parse from right to left, with the left-most sub-component of the domain being the most significant sub-domain, and the right-most sub-component being the most general.

In the UUCP domain, an address is a relative route from the originating system to the destination system. Addresses take the form

host1!host2!host3!...!hostN!user,

where host1 is known to the originating system, host2 is known to host1, and so forth, for a recipient located on hostN. Addresses in the UUCP domain parse left to right, with the left-most sub-component of the address specifying the next host to route the message to.

It is interesting to compare the semantics of the two address forms. In the ZOTnet domain, addresses tend to be completely specified from a universal point, and are reachable from any host in the domain regardless of that host's topological position. In contrast, in the UUCP domain, addresses are strictly source relative, and source-routing is explicitly used. The same address may be completely valid at one host in the UUCP network, and completely unrecognizable at another host. Some of the problems with UUCP domain style addresses are discussed in passing in [RFC819].

Routes

Routing is handled implicitly in the ZOTnet domain by the MTS. The user usually does not specify a route to be taken, and instead relies on the MTS to route messages. Each MSA in the ZOTnet maintains a host table for each channel, which guides the routing process. If the user does wish to specify a message's route in traveling to an addressee, source routing is used in the addressee's address. The form of the local part of such an address is

mailbox%host1%host2%host3%...%hostN

This form of source-routing is similar to the UUCP style of addressing, with the exception that the parse is from right to left: hostN is the next site to route the message to, and host1 is the local host for mailbox.

Clearly one problem that must be addressed during translation from the UUCP domain to the ZOTnet domain is managing to root[1] the UUCP address to a particular MSA so that it can be re-introduced into the UUCP domain as a valid address at a later point. When translating from the ZOTnet domain to the UUCP domain, we should not have this problem, providing that other entry points into the ZOTnet domain exist. If this is not guaranteed, then some sort of rooting should occur when addresses are munged in order to anchor the address semantics to some known basis.

Some suspect that the term source-routing is a pun of sorts for the term source-rooting.

Implementation

It is now time to consider how the address and route translations take place. We shall begin by examining the mapping from the ZOTnet to the UUCP domain, and then examine the dual mapping from the UUCP to the ZOTnet domain. To discuss the address mapping, we will define certain useful symbols. Let <zhome> denote the name of the host which the MSA is running on in the ZOTnet domain, and let <uhome> denote the name of the same host in the UUCP domain. Further, let <from> denote the name of the host on which the message originated in the ZOTnet domain. Finally, let <uucp> denote the name of the UUCP channel for the MSA in the ZOTnet domain which will provide gateway services between the two.

The convention used in drawing Figures 2 and 3, is to represent the translation process in terms of a decomposition chart. In the initial state, one starts with an address. Based on the composition of the address, one of two (or more) paths can be taken. This process is continued until a leaf in the decomposition chart is reached. The corresponding translation is then found directly under that leaf.

ZOTnet to UUCP

The algorithm to map an address from the ZOTnet domain to one in the UUCP domain is summarized in Figure 2.

First, an '@'-sign is checked for in the address. If one is found, the address has a great likelihood of being in the ZOTnet domain. Hence, an '!'-sign to the left of the '@'-sign is checked for. If one is found, the address appears to have a mixed format, and may already have been munged at a previous time. To determine this, we see if the address is of the form

<string1>!<string2>%<uucp>@<zhome>

If so, this is an address which was previously munged from the UUCP domain to the ZOTnet domain, the translation is to reverse the previous translation:

<string1>!<string2>

If the address is not of this source-routed form, then it is unclassifyable, so the action is to translate it to

<uhome>!address

for lack of a better alternative. In short, this form of the address is beyond the algorithm's understanding. It is hoped that another agent may be able to make sense out of the address. Our action therefore is to prepent our UUCP hostname and hope for the best.

```
                              address
                                 |
                 ----------------------------------
                 |                                 |
              string                             s1@s3
                 |                                 |
         -------------             -------------------------------
         |           |            |          |                   |
        s1         s1!s2        s1@s2     s1!s2@s3        s1!s2%<uucp>@<zhome>

<uhome>!s1@<from> s1!s2  <uhome>!s1@s2  <uhome>!s1!s2@s3     s1!s2
```

Figure 2. Translation from the ZOTnet to UUCP domain

```
                             address
                                |
          ---------------------------------------
          |                                      |
        string                                 s1@s3
                                                 |
                             -----------------------------
                             |         |                  |
                           s1@s2    s1!s2@s3      s1!s2%<uucp>@<zhome>

    string%<uucp>@<zhome>    s1@s2     s2@s3      s1!s2%<uucp>@<zhome>
```

Figure 3. Translation from the UUCP to ZOTnet domain

If the address did not contain a '!'-sign to the left of the '@'-sign, then we have an address of the form local@domain, which is clearly in the ZOTnet domain. This address is translated to

<uhome>!local@domain

If the address does not even contain an '@'-sign, a check is made to see if the address contains an '!'-sign? If so, the address is clearly in the UUCP domain, and no translation is done. An address of this sort is bogus. That is, the address is clearly in the destination domain, and does not conform to the current domain's addressing standards.

If the address does not contain an '!'-sign either, the address is in the ZOTnet domain, but lacks a host. The correct translation is to insert the host and translate it as

<uhome>!address@<from>

Several observations can be made regarding this mapping algorithm. In particular, in decomposing addresses from different domains, extreme care must be taken in following the addressing rules for the local domain. The decomposition hierarchy adheres closely to the right-to-left semantics of addresses in the ZOTnet domain. Second, it is critical that addresses that have been previously munged while going the other direction are appropriately translated back to their original form. As the reader will see after examining the algorithm described for mappings from the UUCP to ZOTnet domain, addresses are preserved. Third, an address munger must be able to detect and handle addressing abnormalities. The case in which an address contains both an '@'-sign and an '!'-sign but is not source-routed is such an example. Again, the decomposition rules attempt to perform the correct munging. In this particular case, the address may have been munged at another site in the ZOTnet domain.

UUCP to ZOTnet

The algorithm to map an address from the UUCP domain to one in the ZOTnet domain is summarized in Figure 3. First, an '@'-sign is checked for in the address. If one is found, an '!'-sign to the left of the '@'-sign is checked for. If one is found, then we may have an address which has been previously munged. To determine this, we see if the address is of the form

<string1>!<string2>%<uucp>@<string3>

If this is so, the address is a UUCP address which has been munged to the ZOTnet domain form. As such, no further munging need take place. If the address is not of this source-routed form, then it is a bogus address in the UUCP domain, and the string prior to and including the right-most '!'-sign to the left of the '@'-sign is removed. This is done, as the UUCP style address prior to the ZOTnet style is apparently being used to route the message to an MSA that can connect from the UUCP domain to the ZOTnet domain. In this case, the UUCP style prefix is un-important, as it was included for purposes of routing only.

If no '!'-sign was found, but there was a '@'-sign, the the address is clearly in the ZOTnet domain, and no translation is performed. If no '@'-sign was ever found, then the address is definitely in the UUCP domain and is translated to

address%<uucp>@<zhome>

Note that this decomposition hierarchy has the same qualities as the ZOTnet to UUCP domain algorithm. First, the hierarchy is not inconsistent with UUCP style addressing[1]. Second, previously munged addresses are returned to their original form. Third, the decomposition hierarchy examines the full range of possible addresses and performs the appropriate actions. In this respect, note that unintelligible or bogus addresses are always (somehow) handled.

Other details

Finally, a few other details must be considered. For the ZOTnet domain, a UUCP channel was designed and developed. This channel takes messages in the ZOTnet format, munges them, and then invokes the UUCP server with the appropriate arguments. Address recognition is achieved in two areas: first, the official ZOTnet UA was made to be knowledgeable about UUCP domain style addresses. Second, the source-routed nature of these addresses in the ZOTnet domain causes the UUCP channel to be invoked. The ZOTnet MSA does not however, recognize addresses in the UUCP domain strictly by their pure form,

host1!host2!...!hostN!user

but instead relies on ZOTnet style source-routing translation into

host1!host2!...!hostN!user%<uucp>@<zhome>

to normalize the handling of such addresses. No restriction is placed on mail crossing from the ZOTnet to UUCP domain.

For the UUCP domain, the local UUCP MSA was modified to perform address recognition, and to call the appropriate server to perform the translation and entry into the ZOTnet domain. Note in this case, the UUCP MSA is performing the address recognition and the UUCP UA is not. Since the ZOTnet domain connects to the CSnet and ARPAnet domains, messages from the UUCP domain are strictly monitored. Only bona fide ZOTnet sites are allowed to originate UUCP mail into the ZOTnet domain. However, any UUCP site may send mail to a site that is strictly in the ZOTnet domain. This prevents UUCP sites outside the ZOTnet from using the ZOTnet domain as an unauthorized entry point into the CSnet domain.

Examples of Translation

Let us consider how a given message might have its headers translated. For these examples, note that the UCI host in the ZOTnet domain is the same machine as the ucivax host in the UUCP domain. Also, the name of the UUCP channel in the ZOTnet is uucp. (Hence, source-routed addresses will usually contain a %uucp string).

Suppose that a user in the ZOTnet domain sends a message with the following header fields:

[1] The by default qualities of the UUCP domain work in our favor in this case.

From: Arthur Dent <adent@uci>
Date: 14 Jan 83 18:09 PST (Fri)
To: F.Prefect@Uci-20a
CC: Tricia McMillan <trillian@Uci-20b>,uciunix!beeblebrox
Subject: Test Message

The UA performs address recognition, and munges all addresses to conform to the standards for the UA's domain. The user's UA gives the following message to the MSA:

From: Arthur Dent <adent@uci>
Date: 14 Jan 83 18:09 PST (Fri)
To: F.Prefect@Uci-20a
CC: Tricia McMillan <trillian@uci-20b>,
uciunix!beeblebrox%uucp@Uci
Subject: Test Message

The copy that is delivered to recipients in the ZOTnet domain is not altered. The copy for the recipient in the UUCP domain must pass through the UUCP channel. The UUCP channel gives the UUCP server the following message:

From ucivax!adent@uci Fri Jan 14 18:09:00 1983
To: ucivax!F.Prefect@Uci-20a
CC: ucivax!trillian@Uci-20b,uciunix!beeblebrox
Subject: Test Message

and the UUCP server directs the message's transit through the UUCP domain to the address uciunix!beeblebrox.

One translation made, which has not yet been discussed in this paper, is that between the "From:" and "Date:" fields in the ZOTnet style message and the initial "From" line in the UUCP style message. Translating date formats from one domain to the other is relatively straight-forward and uninteresting in our context, and shall not receive further attention here.

Now let us consider the headers of a reply that a UA in either domain might generate on the user's behalf. For the ZOTnet domain, the headers might be:

From: Ford Prefect <F.Prefect@Uci-20a>
Date: 14 Jan 1983 18:48 PST (Fri)
To: Arthur Dent <adent@Uci>
CC: Tricia McMillan <trillian@Uci-20b>,
uciunix!beeblebrox%uucp@Uci,
F.Prefect@Uci-20a
Subject: Re: Test Message

When this message is processed by the ZOTnet MSA, the addresses in the ZOTnet domain are handled normally. The address, uciunix!beeblebrox%uucp@Uci, in the UUCP domain is first sent to the UCI host. From there, the source–routing in the local part of the address is noticed (%uucp) and the message is re–introduced into the UUCP domain (via the UUCP channel) on the appropriate host in the UUCP domain. As one would expect, the UUCP channel again performs the appropriate munging. The address given to the UUCP server is uciunix!beeblebrox.

For the UUCP domain, the headers of a reply might be:

```
From uciunix!beeblebrox Fri Jan 14 18:48:32 1983
To:     ucivax!adent@uci
CC:     ucivax!F.Prefect@Uci-20a,ucivax!trillian@uci-20b,
        uciunix!beeblebrox
Subject: Re: Test Message
```

The UUCP MSA transmits the message to the UUCP host ucivax, where it's UUCP MSA detects several foreign addresses (e.g., adent@uci, F.Prefect@Uci–20A, and trillian@uci–20b), and invokes the ZOTnet server. The ZOTnet server examines each address, and after verifying that they are strictly in the ZOTnet domain, munges the headers and enters the message into the ZOTnet queue. The munged headers are:

```
From:   uciunix!beeblebrox%uucp@Uci
To:     adent@uci
CC:     F.Prefect@Uci-20a,trillian@uci-20b,
        uciunix!beeblebrox%uucp@Uci
Subject: Re: Test Message
```

This process of replies can continue, with all of the various agents in the MHS handling the interconnection of the two domains for the user. For a detailed schematics which includes all major servers used in the two domains, the interested reader should consult Appendix B.

Results

We can now consider some of the results that came about when interconnection was achieved. In most respects the additional interoperability was a success.

In terms of achieving a successful method of protocol translation, great success was encountered. The demand for munging messages of one format into another led to a general mail filter package being developed that allows users to easily convert mail files between the two systems. A user still has two maildrops, but the UAs for each domain have been modified to automatically filter the contents of the other domain's maildrop. Hence, each user can act as if there is a single, virtual maildrop.

Since the protocol translation is complete between the two UAs, individuals are now presented with a single MHS in which to handle electronic mail. A result of this is that a user can choose a preferred UA and domain to handle all of the user's mail. The importance of this from the user's context can not be underestimated. With additional interoperability, the user's communication environment is easily extended.

Although users must still know the address for a given name, they needn't distinguish between different networks. In fact, most UAs accept either type of address (within limits), and the MSA performs the appropriate actions after the message has been posted. The UAs themselves perform no translation, but the MSA does. It is now easy to see how one can address recipients in both domains in a single message using a given domain's standards, and have each recipient receive a message in a processable format. Again, it is easy to see how additional interoperability greatly enhances the user's communication capabilities.

One major problem which still remains to be resolved was posed under the question of whether addresses from different domains can map to the same name. The current symptom demonstrates itself as the UAs not recognizing the user's maildrop in the other domain as belonging to the user. This usually results in the user receiving two copies of replies that are generated by the UA: the first is sent by virtue of the fact that most UAs tend to add the user's address to the "CC:" list of the user's reply; the second will be sent if the user's address in the other domain is present in the "To:" or "CC:" list of the message being replied to.

Encryption of message headers also remains a problem which has not been solved. One possible partial solution is to have the message envelope contain a duplicate of much of the information contained in the message headers. This scheme is very uncertain however. This remains a research issue.

Finally, the general question of how much address recognition should occur in the MSA remains to be answered. The approach taken in our implementation has been to cater to each: the MSA performs address recognition in the UUCP domain, while the UAs and parts of the MSA perform address recognition in the ZOTnet domain. It is interesting to note that in the current implementation, UA's are allowed to receive mail from foreign domains by two paths: either from the local MSA or from the foreign maildrop. A general explanation and solution of the issues involved in the problem would be welcomed.

Conclusions

This paper has dealt primarily with the interconnection of two widely different computer mail networks, and has shown how interoperability can be achieved. To do so, protocol translation occurs between peers at the application layer protocol which governs the formatting of messages within a given domain. Further, a link is established between the two networks using a store–and–forward channel as the agent of communication.

The interconnection was challenging as the two domains have completely different philosophies regarding message format, addresses, and routing. In addition, the MTS for each domain is entirely different in operation. Despite these problems, a detailed analysis of the addressing and routing mechanisms has yielded an appropriate set of translation rules.

Fortunately, the success of this project has been primarily due to the relative ease with which protocol translation can be achieved in a datagram–like environment. With other applications, this level of success may not be possible. Much research remains to be done in the field of protocol translation, if the more complicated and substantial questions in this field are to be satisfactorily answered.

The practical value of the project has been substantial, users in the ZOTnet have been given a larger scope of interoperability with other computer mail correspondents in the UUCP domain. It is hoped that this additional capability will prove of substantial benefit to the users of the ZOTnet.

Acknowledgements

Doug Kingston at the US Army Ballistics Research Lab, Aberdeen, Maryland developed the first UUCP channel for MMDF in October of 1981. Steve Bellovin at the University of North Carolina, Chapel Hill also performed some development work towards translation.

References

[CSnet-DN1] L. Landweber, and M. Solomon, "Use of Multiple Networks in CSnet", Proceedings of the Compcon Spring 1982, (November, 1981). Also CSnet Design Note DN-1.

[CSnet-DN3] D.H. Crocker, E.S. Szurkowski, and D. Farber, "An Internetwork Memo Distribution Capability — MMDF", Proceedings of the Sixth Data Communications Symposium, (September, 1979). Also CSnet Design Note DN-3.

[IEN048] Vinton G. Cerf, "The Catenet Model for Internetworking", IEN 48, (July, 1978).

[ISO97-16] ISO TC97/SC16 Document Number 227, "Reference Model of Open Systems Interconnection", (June, 1979).

[NBS81] National Bureau of Standards, "Specification of Message Format for Computer Based Message Systems", NBS Report ICST/CBOS-81, National Bureau of Standards, (September, 1981).

[RFC819] Zaw-Sing Su, and, Jonathon B. Postel, "The Domain Naming Convention for Internet User Applications", RFC 819, (August, 1982).

[RFC822] D.H. Crocker, "Standard for the Format of ARPA Internet Text Messages", RFC 822, (August, 1982).

[UUCP-1] D.A. Nowitz, and M.E. Lesk, "A Dial-Up Network of UNIX Systems", Bell Laboratories, (August, 1978).

[UUCP-2] D.A. Nowitz, "UUCP Implementation Description", Bell Laboratories, (October, 1978).

[X-MHS1] International Telegraph and Telephone Consultive Committee, "Message Handling Systems: System Model-Service Elements", CCITT Draft Recommendation X.MHS1, Question 5/VII, (September, 1982).

[ZOTnet] M.T. Rose, "The ZOTnet - a Local Area Mailing Network", Technical Report, Department of Information and Computer Science, University of California, Irvine, (January, 1983).

APPENDIX A
Abbreviations

The following (obscure) abbreviations are used (somewhat) consistently in this paper:

4.1bsd: version 4.1 of the U.C. Berkeley distribution of the UNIX operating system

CBMS: Computer Based Message System

IEN: Internet Experiment Notebook

ISO: International Organization for Standardization

MHS: Message Handling System

MMDF: Multi–Channel Memo Distribution Facility MSA: Message Service Agent

MTS: Message Transfer System

OSI: Open Systems Interconnection

RFC: Request for Comments

UA: User Agent

UUCP: UNIX to UNIX Copy

APPENDIX B
Servers in the two domains

Figure 4 illustrates the relations between the servers in the two domains. No particular hierarchy is enforced, although the flow of this diagram may be viewed as the path that a message might travel.

Following is a terse description of the actual servers:

Servers in the ZOTnet domain

Submit: A process which enters messages into an MMDF queue. The Submit program verifies the authenticity of the entity posting the message, and also validates the message's adherence to the formatting standard. After the message has been entered into the queue, the Submit program may invoke the Deliver program to process the delivery of the message. This decision is based on the channels through which the message must travel.

Deliver: A process which manages the channel-service for a message. For each message in the queue, it sees what channels are required and fires those channels that are active. When a message has been fully processed (i.e., when all addresses have been successfully sent through a channel), the Deliver program removes the message from the queue.

Local Channel: A process which implements the delivery-slot protocol for the recipient of a message. That is, if the recipient has specified a rcvmail hook to process incoming messages, the Local channel invokes that hook; if not, then the Local channel delivers the message to the user's ZOTnet maildrop.

Phone Channel: A process which calls another system and invokes the Slave program on that system in order to transfer message from one MSA to another.

Slave: A process which is invoked by a Phone channel on another system. The Slave program interacts with the Submit program to enter the message into the MMDF queue on that system.

UUCP Channel: A process which interacts with the UUX program to send a message in the UUCP domain. During its execution, the UUCP channel performs protocol translation at the UA level for the entire message, and protocol translation at the message transfer level for the envelope.

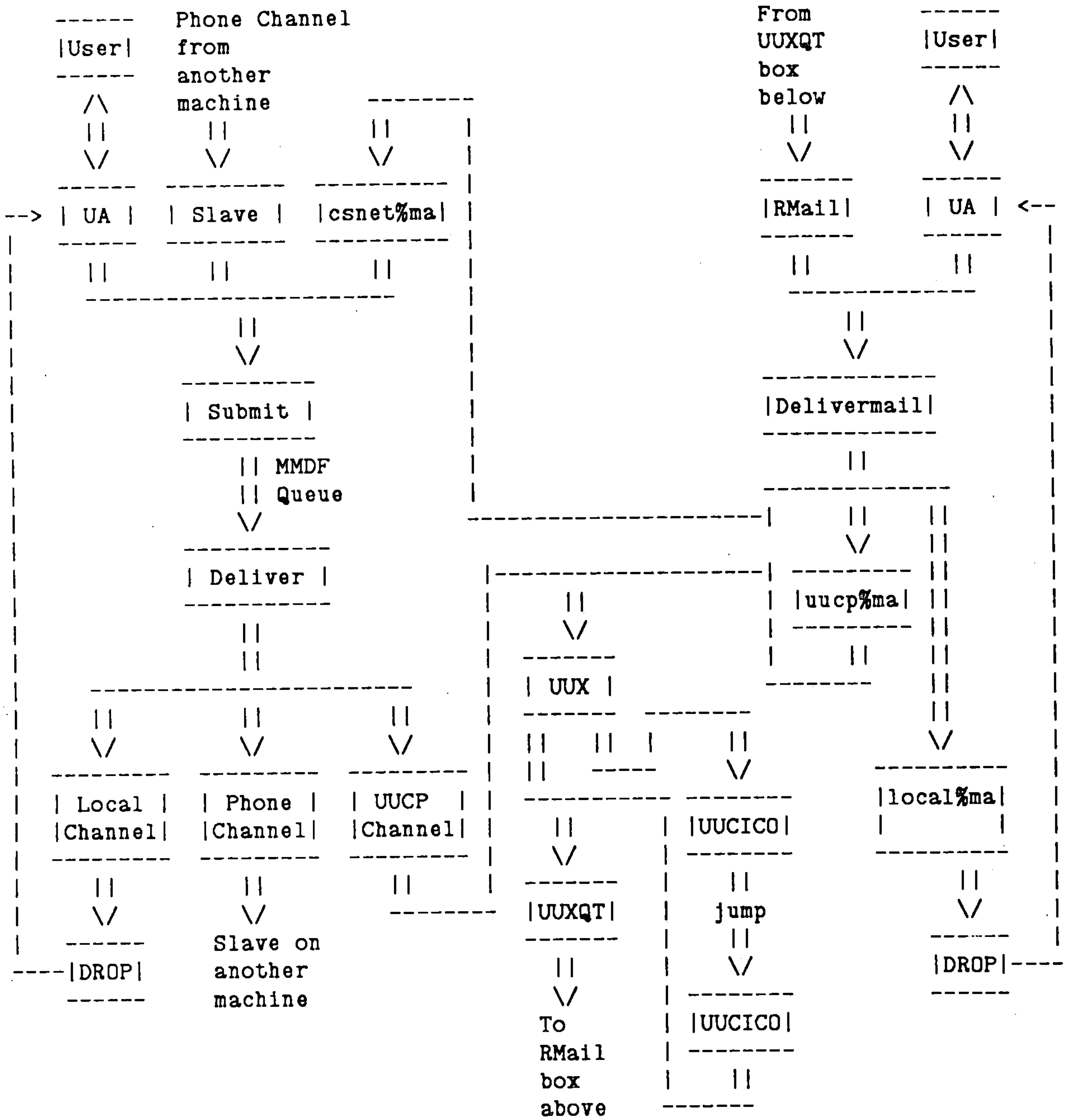

Figure 4. Servers and their relationships

Servers in the UUCP domain

Delivermail: A process which manages message delivery for UNIX mail messages. There is no MSA queue for the messages. The Delivermail program performs address recognition and fires the appropriate mailer to handle individual addressees.

local%mail: A process which implements the delivery–slot protocol for the recipient of a message. In the UUCP domain, this means deliverying the message to the user's maildrop, and notifying the user if the user is logged in and wishes to be so notified.

csnet%mail: A process which interacts with the Submit program to enter a message into the ZOTnet domain. During its execution, this program performs protocol translation at the UA level for the entire message, and protocol translation at the message transfer level for the envelope. This program also enforces access control policies for messages entering the ZOTnet domain from the UUCP domain.

uucp%mail: A process which invokes the UUX program to deliver a message to another UUCP machine.

UUX: A process which enters command and data files into the UUCP queue in preparation for execution of the commands on a remote system. After entering this information into the queue, the UUX program sees if the system on which the commands are to be executed is the local UUCP system. If so, it invokes UUXQT directly; if not, it invokes the UUCICO program.

UUCICO: A process which calls another system and invokes a UUCICO slave on that system. The UUCICO program then transfers the command and data files to the other system. Upon completion of the transfer, the slave UUCICO program invokes the UUXQT program.

UUXQT: A process which executes a command file in the UUCP queue using the data files placed their by UUX or a slave UUCICO. In the mail context, the command file specifies invocation of the RMail program.

RMail: A process which invokes the Delivermail program on a specified message with a set of addressees.