Mail Routing using Domain Names: An Informal Tour

Craig Partridge
CSNET Coordination and Information Center
BBN Laboratories Inc.†

Abstract

Four of the major mail networks, the DARPA Internet, UUCP, CSNET and BITNET, tentatively plan to adopt a common naming scheme, based on one adopted by the Internet. Since the naming scheme is new to all the networks, it has required considerable conversion work, particularly in the area of mail routing. A description of the conversion process and some of the more interesting routing problems is presented.

1. Introduction

As some members of the UNIX[1] community are already aware, there have recently been a number of developments in host naming practices in four of the largest mail networks (the DARPA Internet, UUCP, CSNET and BITNET). The most significant change is that the system of appending a network appellation, such as *.uucp* or *.csnet*, to a host name to indicate the network on which the host resides is being replaced. All the networks are committed (in varying degrees of certainty) to using a new uniform naming scheme that will use the domain naming scheme recently adopted as the standard for the Internet.

The creation of a uniform name space is very convenient for mail users, but raises new problems for mail system managers. The major problems involve routing. In the past, presented with a message addressed to a user on *loki.arpa*, a mailer[2] needed only to look at the *.arpa* suffix to know that the destination host was on the Internet. From this point, routing was straightforward: if the mailer was on the Internet, it could deliver the message directly; if the mailer was on another network, it would send the message to a mail gateway on the Internet for final delivery.

Now, however, the mailer is presented with a message addressed to *loki.bbn.com*. The new name intentionally conveys no network information, since underlying the new naming scheme is the philosophy that users should not need to know which network a given host is on to address mail to that host. That is all well and good for users, but what's a poor mailer to do?

---

† The author's mailing address is: c/o BBN Laboratories Inc., 10 Moulton St, Cambridge, Massachusetts, 02238. He can also be reached at the following electronic mail addresses: craig@sh.cs.net, craig@loki.bbn.com, or for the bangist world ..!{harvard,seismo,decvax,ritcv}!bbnccv!craig.

[1] UNIX is a trademark of AT&T Bell Laboratories.

[2] Throughout this paper the term "mailer" is used to describe the collection of one or more programs that make up the mail delivery system on a host, or the MTA for those familiar with the X.400 terminology.

The answer, it turns out, depends upon the network on which the mailer resides. UUCP, BITNET, CSNET, and the Internet will probably all use different algorithms for deciding how to route messages − in large part because their network topologies are different. In the rest of this paper, I discuss how each of the networks plans to handle the problem of mail routing with domain names, insofar as the mechanisms are known, and then discuss some of the more interesting routing problems encountered so far.

2. The Networks

2.1. Internet

The Internet is the name usually given to those hosts which are connected, either directly or indirectly, via other networks to the ARPAnet and MILnet. Access to and use of the network is controlled by the Defense Communications Agency and is usually limited to organizations doing DOD-sponsored work.

The Internet is unique among the four networks in that it does not rely primarily on store-and-forward mail systems for transmitting messages. Because the Internet is a system of interconnected local-area and wide-area networks using the TCP/IP protocols, every host can connect directly to any other host on the Internet. As a result, until recently, the method for delivering mail on the Internet was to open a connection to the destination host and pass the message on using the Simple Mail Transfer Protocol (SMTP).[3] Any more complicated routing of a message was usually done by manipulating the address in the message. For example, if you wanted to send a message to *craig@loki.arpa* via *brl.arpa*, you would send the message to *craig%loki.arpa@brl.arpa*. For hosts off the network, for example *friend@oxbridge.csnet*, mailers were usually specially configured to rewrite addresses to route via mail gateways. The mail gateway for all of CSNET has been *csnet-relay.arpa* (now *relay.cs.net*), so a mailer would have rewritten the address as *friend%oxbridge.csnet@csnet-relay.arpa*. The CSNET relay ‘‘knows’’ how to reach *oxbridge.csnet* via PhoneNet. Mailing to a UUCP site was handled similarly, by delivering the message to one of the UUCP gateways, which would then use the *pathalias* program to determine a route (for example, a message to *user@site.uucp*, might be converted at the UUCP gateway to *host1!host2!site!user*).

However, when the Internet decided to convert to domain names the rules for delivering mail changed. To explain why this is so requires a bit of background.

For some years a plain text database listing information about every host on the Internet (addresses, services supported, host name and aliases, etc.) has been kept on a host at the Internet Network Information Center (the NIC). Every host on the network would periodically retrieve a copy of this table to learn the names and locations of the other Internet hosts. Recently it has been felt that the burden of maintaining and redistributing this database at a central point has become unreasonably high, and that the host information should instead be stored in a distributed database using domain names. Such a database has now been implemented.

The distributed database stores all of its information in entries known as *resource records* (RRs). Each RR matches a domain name with a particular piece of information about that name. For example, for *sh.cs.net*, there is a separate RR for each of its Internet addresses, its hardware, and operating system type. To get the desired information about a host, a query to the database is issued for the domain name and the type of RR of interest (e.g. RRs which list addresses for

[3] Described in Postel [9].

sh.cs.net).

The query is processed by a *resolver*, which does a ‘‘walk’’ through the distributed database using the domain name as a guide. The database is a tree, rooted by a server at the NIC. To resolve *sh.cs.net*, the resolver first queries the server at the NIC for information on *sh.cs.net*. The root server resolves as much of the name as it can and then tells the resolver where to go for additional information (or returns an error if based on its own information, it knows the name is bad). In the example *sh.cs.net*, the server would reply that the name appears valid but that to get a full answer, the resolver must consult the server for the subdomain *cs.net*, which is at the CSNET CIC. The resolver then proceeds to query the *cs.net* server, which has *sh.cs.net* as a leaf. The server returns the addresses to the resolver.[4]

One of the things that the domain database stores is resource records for mail transmitters (known as MX RRs). An MX RR associates a domain name with a list of hosts which will relay or accept mail for it. A domain name may have more than one MX RR associated with it. MX RRs for a given name are ordered. MXs earlier in the list are preferred over later ones. Mailers are required to try to deliver a message to each MX in succession until a successful delivery has been made, the message has been authoritatively rejected, or all of the MXs have been tried.[5] MXs are required to pass messages on to their final destination, either through another MX or directly.

One may wonder why a new, and more complex, routing system was devised to replace the existing system. One reason is simply the desire to move routing information out of mail headers. The complexity of mail addresses in messages traversing the Internet is astonishing, and is widely perceived as unnecessary.[6] Since much of the complexity comes from the need to do routing in message headers, moving much of the routing information out of the headers is a useful step. Another reason for the complexity of the new system is that less flexible routing systems actually turned out to be more difficult to implement.[7] Finally, this system offers several advantages over the old:

- extremely flexible routing. Sites which have intermittent links to the Internet (such as networks connected by satellite or low speed phone links) can set up a system of MXs which increase the chance that, when the link is up, queued mail will get passed through.

---

[4] The system may sound cumbersome to some people. In fact it often works faster than doing a linear search of the plain text database. In addition, there is a local caching mechanism that ensures that frequently used names will usually be known to the local resolver. The entire system is defined and described by Mockapetris [4], [5] and [6].

[5] Actually, mailers need only try a reasonable subset of the MXs, where a reasonable subset is felt to be much larger than one MX. The precise interpretation of MX RRs is described by Partridge [8].

[6] This is not a criticism of the standard for the format of Internet mail headers (see Crocker [2]), but rather a commentary on the willingness of mail systems to make unnecessary use of the rich range of formats available.

[7] Indeed, the original plan for the domain system called for simply classifying mail information into two RR types, one for mail forwarders, which passed the messages on to the final destination, and one type for destinations, which were the host to which final delivery was to be made. This system proved to have at least two major problems. First, it wasn’t quite flexible enough. Several sites wanted to have more than one mail forwarder, and had preferences about which ones to try first; the simple two type mechanism didn’t support this. Second, dividing the world into two RR types turns out to cause partial information problems (a mailer could accidentally get only the destination or the list of forwarders, and make a bad, and potentially fatal, routing decision).

- dynamic re-routing of in-transit mail. Mailers are encouraged to do routing at delivery time (i.e. not when the message is originally submitted or received). As a result, if a host goes down with a severe hardware failure, a change to the database can cause in-transit mail to be automatically re-routed to a backup host.
- mail relays are transparent. If *oxbridge.edu* is actually on CSNET, an MX RR that forwards mail for *oxbridge.edu* to the CSNET relay will do so ‘‘quietly’’. (No more *user%oxbridge.csnet@csnet-relay.arpa*.)

Currently, the Internet is in the midst of the transition to the domain name scheme. Almost all Internet mailers can interpret domain names, but only a few know how to interpret MX RRs. (The rest simply use the old system of mapping name to destination network address but use the new domain names to get the address).

2.2. CSNET

The Computer Science Network (CSNET) was established in 1981 to provide network services, primarily electronic mail access to the Internet, to institutions engaged in computer science research. Recently its charter has been expanded to include any organization doing computer-related research in the sciences and engineering (physics, chemistry, etc.). It is managed by the University Corporation for Atmospheric Research under contract to the National Science Foundation and is administered by the CSNET Coordination and Information Center (CIC) at BBN Laboratories.

CSNET member sites fall into one of two general categories: Internet sites and PhoneNet sites. Internet sites are connected to the Internet either through local links, or through use of CSNET-provided software which allows them to reach the Internet via commercial X.25 networks. Sites already connected to the Internet often choose to join CSNET because the rules for use of the Internet by CSNET members are more liberal than those for DOD-sponsored Internet users. For the purposes of mail routing, these sites behave like regular Internet sites.

However, the vast majority of CSNET members (130 of 174 sites) get mail-only service through PhoneNet. PhoneNet is a store-and-forward star network, with the central node at *relay.cs.net* (formerly known as *csnet-relay.arpa*). Sites either call, or are called, on a regular basis by the relay to send and receive messages. Data is transmitted over 1200 and 2400 baud lines using protocols developed at the University of Delaware. All messages for PhoneNet sites must go through the relay for delivery.

Because CSNET is tied so closely both technically and culturally with the Internet, it is firmly committed to converting to domain names. Internet sites will have to use domains as a requirement for future Internet use, and the CSNET CIC is strongly encouraging PhoneNet sites to convert to domain names to maintain compatibility with the Internet.[8]

The key to making domains work for PhoneNet is making them work at the center of the star network, on *relay.cs.net*. The situation is made a bit more complex by the fact that the relay is also connected to the Internet, and must therefore coexist in both networks. Indeed, converting *relay.cs.net* to handle domains led to the discovery of some of the domain routing problems mentioned below. Happily, most of the domain conversion has been accomplished. The CSNET relay knows how to handle domain names with MX RR’s on the Internet, recognizes domain names

[8] The NIC has kindly agreed to register names for non-Internet sites such as CSNET PhoneNet and UUCP sites. This avoids the headache of different (or worse, colliding) domain names in different networks.

corresponding to PhoneNet sites the relay serves, and is capable of doing some limited rewriting of domain names to make mail easier to accept for sites having trouble converting their local software to use domain names. As soon as all CSNET sites have registered domain names, CSNET will have converted to domain names.

2.3. UUCP

The UUCP-based network is the largest of the three store-and-forward networks (containing approximately 6,000 hosts) and is probably best described as a loosely federated collection of hosts. Hosts usually communicate via regular phone calls, during which messages are transferred using the UUCP protocols. Links that use X.25, TCP, and other transport protocols also exist. Connectivity is variable, with some hosts communicating via dedicated lines, and others working through a single intermittent phone link to one other host. A certain amount of network coordination work is performed by the UUCP Project.

Mail routing in the network has traditionally been explicit in the address. To reach a given host on the network, a user needed to explicitly list the route from the current host to the destination (for example, *ihnp4!wjh12!harvard!bbncca!craig*). Recently the task of routing mail has become simpler with the advent of the *pathalias* program, which can take a name and, using a database, find a route through the network to the named host.

Because of the nature of the UUCP network, conversion to domain names will be voluntary. Sites which do not wish to participate need not do so. It is expected that most sites will convert to using domain names because it is convenient.

The need to retain compatibility with hosts not supporting domains has created interesting problems for authors of new UUCP mail systems. Very recently software has appeared, most notably *smail* from the UUCP Project, which can support both types of hosts. The key to the software is that while the texts of messages will use addresses of the form *user@domain*, the actual routing by mailers will continue to use the old notation based on exclamation points.[9] Domain names will be resolved incrementally, with each host forwarding messages to another host which understands more of the domain name until the message reaches its destination.

2.4. BITNET

BITNET, a store-and-forward network serving the academic community, communicates using RSCS, an IBM protocol, over dedicated 9600 baud lines. Each BITNET site is required to have connections to two other sites, thus providing the necessary connectivity and reliability. Sites relay messages to their final destination. Network support is provided by EDUCOM and the City University of New York.

Currently,[10] BITNET still has not decided whether to use Internet domain names. A task force met in March of 1986 and, among other things, recommended that BITNET adopt the use of domain names, but this recommendation has not yet been officially adopted by the BITNET Executive Committee.

---

[9] This system is described by Horton [3].

[10] April 25th, 1986.

3. Interesting Problems

Readers will probably not be surprised that there are some problems posed by converting to the use of domain names. The more difficult problems observed so far do not occur when routing mail within a given network but when mail must cross network boundaries (e.g., mail sent from the Internet to a host on UUCP). Since the four networks are interconnected (which is why a consistent naming scheme was appealing in the first place), and a large amount of mail flows between them, the problems have some very practical consequences. Here are some of the more interesting problems encountered:

3.1. Naming Problems

The domain naming scheme assumes that a host's name has some correspondence with the organization operating the host. Or to put this another way, all the hosts within an organization are normally expected to have names in the same domain. However, currently it is not uncommon, particularly in university settings, for different divisions of an organization to be members of different networks and have no connections with each other. For example, almost every university computer science department in the United States is a member of CSNET. At many universities there is also a general purpose campus computing center which is often connected to BITNET. In some cases, there is no network link between the computer science department and the campus center. In the past, mail for the two sites could be distinguished by the *.csnet* and *.bitnet* suffixes.[11] If, for example, both are part of ''Oxbridge University'', and now in *oxbridge.edu*, how do we make sure that mail sent to some host in the Oxbridge University campus center doesn't accidentally get sent to the computer science center and die there?

One obvious option is to simply keep lists of exceptional cases. Where a site is served by more than one network, each network would keep track of which networks served the various hosts at the site. The level of inter-network coordination, however, not to mention the probable size of such lists, probably makes this solution unwieldly. One might also note that on the Internet, where every host routes messages itself and there are no internal relays or backbone sites, it is likely that every host on the network would have to store these lists, which is clearly unworkable.

A better option is to extend the domain names to show internal organizational divisions. Domain names can actually contain up to 63 labels (where a label is a sequence of characters delimited by dots, such as *sh* in *sh.cs.net*) and have a total length of 255 characters. Thus, using the example of Oxbridge University, a host in the computer science department could have a name ending in *cs.oxbridge.edu*, and the campus computing center machines could have names ending in *ccc.oxbridge.edu*. The networks would only need to know that mail to *<host>.cs.oxbridge.edu* should be sent via CSNET, and that mail to *<host>.ccc.oxbridge.edu* should be sent to BITNET.[12]

The divisional naming trick works well for most situations, but does have a few problems. One problem is philosophical. A stated purpose for using domains has been to spare users from having to remember on which network a given host within Oxbridge University resides. Divisional naming is analogous to inserting *csnet* (in place of *cs*) or *bitnet* (in place of *ccc*) into the

[11] Actually people tend to distinguish both by network suffix and also in the name. Sites which used the same name with different suffixes on two networks have already encountered occasional problems because mailers get confused or ignore the suffixes, causing mail to be mis-delivered. Clearly domain naming makes the potential confusion even greater.

[12] Somewhere within BITNET and CSNET, information has to be stored about how exactly to find the given host, but this information presumably exists already.

domain name, thereby bending the spirit of the naming scheme. We may clothe our nakedness in the justification that divisional naming is more mnemonic or conveys additional information about the host, but in our hearts we probably know better.

The other problem with divisional naming is practical. It is quite possible to have two hosts within a division which are unable to communicate. A good example is an IBM machine, which will probably happily communicate with BITNET but may have little concept of UUCP, PhoneNet or TCP/IP. If it shares a computer room with a UNIX machine on the Internet, there is not much that a network-independent naming scheme can do to distinguish between the two hosts, except to list routes for them individually.

Obviously the best solution to this problem is for all hosts at a site to be somehow interconnected, but we are a long way from implementing this solution for all environments.

3.2. Gateway Problems

A tougher class of problems exist for hosts which reside on more than one network. The problems exist for any host on more than one network but are most acute at mail gateways, hosts which relay mail from one network to another (for example, *relay.cs.net* between CSNET and the Internet, and *wiscvm.wisc.edu*, the BITNET-to-Internet gateway).

Gateway problems appear when choosing between various routing options provided by the different networks. There are a variety of different possible scenarios; here are some of the more interesting ones discovered so far:

Imagine a message on *relay.cs.net* addressed to a host in *att.com*. Hosts in *att.com* can be reached via both UUCP and CSNET. Suppose, for a moment, that AT&T has decided that it would generally like mail from the Internet routed via UUCP instead of CSNET, and thus has listed a UUCP gateway as its preferred route in the Internet database.[13] Because *relay.cs.net* is on the Internet, it is capable of querying the domain database and learning that the preferred Internet routing for *att.com* is via a UUCP gateway. The question is, should *relay.cs.net* make that query? Does CSNET have an obligation to consult another network about how to route messages that it can deliver internally?

One might instinctively answer that, given that we know AT&T's preferences, *relay.cs.net* should heed the expressed desires of AT&T and send the message via the Internet to the UUCP gateway instead of delivering the message within CSNET. But there is good reason not to consult the other networks.

What if AT&T wants to use CSNET as a backup for UUCP service? After all, they have both services, why not use them? Then AT&T presumably has listed *relay.cs.net* as the second choice mail relay for hosts in *att.com* (after the UUCP gateway). It is now likely that our hypothetical message on *relay.cs.net* was sent to the relay because some Internet mailer couldn't reach the UUCP gateway and has successfully used the second choice. In this situation, if *relay.cs.net* consults the Internet for routing information it will undo AT&T's attempt to use CSNET as a backup.

Of course the message for *att.com* may have originated within CSNET, in which case we arguably still want to consult the Internet for routing information. Many mailers have at least a vague notion of how they received a message, so conceivably a routing mechanism which took into account the network from which the message came could be devised: if the message is from

[13] It should be emphasized that the example routes are all purely hypothetical, and are not intended to reflect on the quality of the mail delivery systems on the different networks.

CSNET, try routing via the Internet first; otherwise try routing via CSNET first. However, recall that AT&T still wants the message sent via CSNET if we can't reach the UUCP gateway. Therefore what the mailer must do (if the message came from CSNET), is get the Internet routing information, confirm that it can reach the UUCP gateway (e.g. try to deliver the message) and if that fails, try via CSNET. Unfortunately this suggests that the relay's mailer must do both routing and delivery at once, while the general trend has been to separate these two functions into separate modules to reduce mailer complexity.

The reader should also note that the preceding strawman examples were carefully chosen not to raise issues about how information from the different routing systems might be compared. Because the Internet database stores a ranked list of MX RRs, it can tell us that AT&T prefers to have its mail routed via UUCP. If a gateway between CSNET and UUCP existed, the routing decisions would become much trickier because the routing systems for the two networks are not compatible. On the CSNET side, the mailer knows that it can reach any host by sending the message on to *relay.cs.net*. On the UUCP side, *pathalias* can tell the mailer how many intermediate host hops are required to reach the destination. If the number of UUCP hops is greater than 1 (i.e. the mailer doesn't talk to the destination directly), there is no easy way to figure out which route is faster.[14]

It appears that the best solutions to the problems of mail gateways (without requiring large amounts of research work) are the simple ones. None of the three mechanisms described below is perfect. Indeed, none of them even attempts to determine the ''best'' route. They are designed to be relatively simple, easy to understand, and to have a low chance of causing mail loops. Presumably, a mail gateway can choose one of these algorithms to obtain reasonable routing performance for its particular configuration.

One solution is to establish the rule that a gateway should refuse to move mail between networks unless it must. This method works quite well if there is only one gateway between two networks. All mail destined to cross networks will be directed to the gateway. When trying to route the message, the gateway will discover that on one network it is supposed to mail to itself, while on the other network it can send the message on to the destination. The decision process is simple, requiring only that a mailer realize that a route to itself implies it should try another network. Unfortunately, the system breaks down if there is more than one gateway available to reach a given destination. There is some chance that two gateways will end up in a loop, sending mail for some destinations between each other indefinitely.[15] Another problem is that hosts on one network have only a single way to get to a host on another network. If mail to *att.com* from the Internet reaches *relay.cs.net*, the relay will pass the message on to a UUCP gateway instead of delivering via CSNET, because the message does not appear to need to leave the Internet to be delivered.

A second solution is for a gateway to have a preference about which network to use. This reduces the chance of mail looping between gateways (this can only happen between gateways

[14] Even if the destination host is only one hop away, it may be faster to send via CSNET if the UUCP link is down.

[15] One can, for example, imagine a UUCP gateway and *relay.cs.net* shuttling a message for *att.com* between each other because both resist letting the message out of the Internet. This scenario is, in fact, impossible because the Internet standards forbid MX hosts from sending to MX hosts less preferred than themselves. So the UUCP gateway would be forbidden to send to *relay.cs.net*. But the problem may be encountered on other networks with different routing mechanisms.

within the network that the gateway prefers to use, not on both networks), and the choice of networks to a destination can be exercised. If *relay.cs.net* prefers to send via CSNET, as it presumably would, mail to *att.com* that reached the relay would be sent within CSNET. The flip side of this solution is that mail will tend to stay within the gateway's preferred network. Explicit routing, such as *@uucp-gateway:user@domain*, would be required to force the mail deliverable within the preferred network to cross the network boundary.

Finally, one can use an entirely different solution, and cross network boundaries whenever possible. If a message comes in on one network, the gateway tries to send it out on the other. This strategy actually seems to have the least chance for causing mail to loop. Unfortunately, the scheme runs into other problems. It is the strategy least likely to honor network access restrictions, such as those for the Internet, because it encourages messages to go through other networks. It also does not work as well as the ''choose a network'' strategy for star networks such as CSNET's PhoneNet, where mail at the central relay that can be delivered within the network is always only one hop from its destination, and thus is probably best delivered internally.

3.3. X.400

Now that we may envision several interconnected networks using the Internet domain naming standard, how well do we fit in with the the rest of the mail world? In particular, how compatible are these networks likely to be with the emerging X.400 standard? The answer seems to be that while the standards are not compatible, messages can be translated from one format to the other, and that mail gateways to X.400 networks will begin to appear in the forseeable future.

In addition, an Internet standard will shortly be issued (if it has not been released already) describing how to convert from messages in Internet standard RFC-822 format to X.400 format and vice-versa.[16] Since CSNET and most of the UUCP community also use 822 messages (or something very similar), we may hope that software implementing this standard could work on all of these networks, and that this software will appear soon.

The way in which BITNET will interface with X.400 is less well known (at least to this author), but their needs may be more pressing than that of the other networks. BITNET communicates with 200 European sites, and European countries are moving swiftly towards adopting X.400 as the mail standard. As a result, in the near future BITNET will be obliged to interface with X.400 systems, and thus it seems likely that some of the first X.400 gateways will appear on BITNET.

4. Conclusion

As the title of this paper implies, its purpose has been more informational than research oriented. The hope is that USENIX attendees will have a better sense of how the mail systems with which they come in contact will likely evolve over the next few years, and how these systems in turn, will deal with the larger outside world (e.g. X.400).

On the technical side I've only looked at the most basic question: Can mail be successfully routed between two points in the cooperating networks? Or to put the problem more bluntly, can using domain names be made to work? Happily the answer is yes. A tougher question is whether we can route mail optimally, so that it takes a minimum amount of time to get to its destination. It turns out that we don't know the answer to this question and may not for some time. It depends

[16] The author would like to thank Steve Kille for allowing him to read his draft of this standard.

on a variety of factors, most notably how easily routing information from different networks can compared. Stay tuned.

References

[1] Comer, D., The Computer Science Research Network CSNET: A History and Status Report, *Communications of ACM*, October 1983, 747-753.

[2] Crocker, D.H., *Standard for the Format of ARPA Internet Text Messages*, RFC-822, University of Delaware, August 1982.

[3] Horton, M., *UUCP Mail Interchange Format Standard*, RFC-976, Bell Laboratories, February 1986.

[4] Mockapetris, P., *Domain Names − Concepts and Facilities*, RFC-882, USC Information Sciences Institute, November 1983.

[5] Mockapetris, P., *Domain Names − Implementation and Specification*, RFC-883, USC Information Sciences Institute, November 1983.

[6] Mockapetris, P., *Domain System Changes and Observations*, RFC-973, USC Information Sciences Institute, January 1986.

[7] Nowitz, D.A., and Lesk, M.E., *A Dial-Up Network of UNIX Systems*, Bell Laboratories, August 1978.

[8] Partridge, C., *Mail Routing and the Domain System*, RFC-974, CSNET CIC, BBN Laboratories, January 1986.

[9] Postel, J., *Simple Mail Transfer Protocol, RFC-821, USC Information Sciences Institute, August 1982*.

[10] Postel, J. and Reynolds, J., *Domain Requirements*, RFC-920, USC Information Sciences Institute, October 1984.