Network Working Group
INTERNET-DRAFT
Expires: May 1998
Intended Category: Informational

H. Lachman
Netscape Communications Corp.
November 1997

LDAP-based Routing of SMTP Messages: Approach Used by Netscape <draft-lachman-ldap-mail-routing-00.txt>

Status of this Memo

The previous version of this draft was <draft-ietf-asid-email-routing-ns-00.txt>. Changes relative to that version are in Sections 4.2, 5.1, 5.3, 7, and the appendices.

Abstract

Directory services based on the Lightweight Directory Access Protocol (LDAP) [1] and X.500 [2] provide a general-purpose means to store information about users and other network entities. One of the many possible uses of a directory service is to store information about users' email accounts, such as their email addresses, and how messages addressed to them should be routed. In the interest of interoperability, it is desirable to have a common schema for such email-related information.

This document discusses some of the fundamental questions to be considered in the development of a common schema for LDAP-based routing of SMTP [3] messages, presents an approach that has been implemented and deployed, and discusses possible extensions to that

approach that may serve to make it more complete and general. The intent is to provide information about an existing implementation, and to stimulate discussion about whether and how to standardize the relevant aspects of LDAP-based messaging management.

1. Background and Motivation

LDAP-based directory services are currently being used in many organizations as a repository of information about users and other "network entities" (such as groups of users, network resources, etc.). Some information is stored in the directory for the consumption of persons browsing for information (e.g., telephone numbers, postal addresses, secretary's name), while other information (e.g., login name, password, disk quota) is stored for use by one or more network applications or services. It is the latter kind of information that is of interest in this discussion. In general, it is advantageous for different network applications and services to refer to the directory for user account information, rather than each service keeping its own collection of user account records, which requires the network administrator to separately create or destroy user entities, passwords, etc., in many different systems each time a user joins or leaves the organization. The goals of centralized user management and sharing of information with other service types drove our decision in the design of Netscape Messaging Server (an SMTP-based mail server product) to use LDAP-based directory services as a common repository for user account information.

Now, if a given mail server can refer to the directory for its own users' account information, it follows that that same information is visible to other LDAP-aware mail servers in the same organization, and therefore that information can aid those other mail servers in correctly routing messages to users of the mail server in question. This assumes that there is an agreed-upon set of per-user attributes to support message routing. If this assumption is met, we can obviate other means currently employed to specify per-user message routing (such as the Unix "aliases" database). The benefit of this is to further consolidate per-user system information.

If each vendor's mail server product has its own schema for LDAP-based message routing, then the above benefits can be achieved for single-vendor customers, but customers who have multiple vendors' mail server products would not be well served. They will likely expect interoperability, which will require a common schema to be supported by the various vendors' products. Thus, it is worthwhile to consider how to develop a common schema.

This document does not attempt to define a standard. It does attempt to define what the relevant questions are, and goes on to describe

one vendor's solution plus possible extentions to generalize it. It is hoped that this discussion helps to characterize the issue, and encourages the development of a common solution.

This document considers only intra-enterprise SMTP message routing using LDAP-based directory services. Solutions and issues involving inter-enterprise routing, non-SMTP message handling, non-LDAP directory services, and other messaging management topics not related to message routing, are outside the scope of this document (except that the concepts presented may also be applicable in the case of X.500 directory services).

2. Terminology

In the context of this document, a "mail server" is a Simple Mail Transfer Protocol (SMTP) message transfer agent (MTA). It either includes, or has associated with it, a local message delivery agent (MDA) that performs delivery to accounts that are considered local to the particular mail server. A mail server may offer related services as well, such as providing a means for mail user agents (MUAs) to pick up messages, but that is outside the scope of this discussion.

The term "account" is used to refer to any entity that can receive mail. This includes mail user accounts, as well as mail group accounts (mail distribution lists). A "delivery" is said to have occured when an MTA passes a message to the local MDA, having first ascertained that the message is destined for a particular account that can be delivered to locally. The MDA may then place the message in a local message store, and/or take other actions as specified by the account's attributes.

"Routing" and "forwarding" are distinct actions. "Routing" is said to have occured when an MTA passes a message to a "next-hop" MTA, having ascertained that the addressed entity is not a local account but may exist elsewhere. "Forwarding" is said to have occured when a message has successfully arrived at a particular account and the MDA determines that the message must be resent to one or more new destinations as specified by the account's attributes. "Forwarding" may be configurable by the user, while "routing" is normally configurable only by a network administrator. With this definition, it might also be said that when a message arrives at a mail group account, and the MDA resends the message to all of the individual group members, this is also "forwarding".

3. Questions to Consider

When a message arrives at an MTA, the MTA must determine whether to deliver the message to a local account, route the message to another

MTA, or, in the case of an unresolvable recipient address, take some remedial action such as "bouncing" the message. In the course of making this determination, an MTA may reference information from a variety of sources, including its own local configuration, one or more directory services, and perhaps other sources. This document discusses only per-account routing and addressing information provided by an LDAP-based directory service, and what role that information might play in helping the MTA determine what to do with a message.

The question of how an MTA might use such information can be broken down into three subquestions:

Question (1): For a given recipient address, which LDAP entry does it belong to?

Question (2): For a given LDAP entry, should a message addressed to it be delivered locally or routed?

Question (3): If a message addressed to a given LDAP entry needs to be routed, to where should the message be routed?

In order for these questions to be answerable by the MTA, LDAP entries that represent mail accounts should include attributes that specify addressing and routing information. These attributes should be advertised to (i.e., readable by) the MTAs that are expected to act on them, and there should be a definition of what attributes are involved and how they are to be interpreted. With this definition, an MTA can be implemented or configured to correctly use such information to answer the above questions, and, therefore, correctly handle messages addressed to accounts represented as LDAP entries.

4. Description of Solution Implemented

In the design of Netscape Messaging Server, we defined two new LDAP object classes, 'mailRecipient', which is used to represent a mail user account, and 'mailGroup', which is used to represent a mail group account (mail distribution list). An LDAP entry of either class may have attributes that are of an "account configuration" nature and are used solely by the MDA handling mail for the account, while other attributes are used by the account's "home" MTA as well as other MTAs. It is this latter set of attributes that are of interest in this discussion, since we are concerned with the behavior of MTAs.

Our MTA implementation uses the following attributes:

mail	primary email address

mailAlternateAddress	additional email addresses
mailHost	server hosting mail account
uid	user id (login name)

The 'mail' and 'mailAlternateAddress' attributes are used to specify the email addresses [4] that are considered valid for an account. They must all be complete email addresses (e.g., "joe@example.com", as opposed to "joe" or "joe@mars"). 'mailHost' is the fully-qualified domain name [5] of the mail server that considers the account to be locally deliverable (e.g., "mars.eng.example.com"). 'uid' is the user's login name. A 'mailGroup' is not expected to have a 'uid' attribute, and may or may not have a 'mailHost' attribute, but both attributes should be present for a 'mailRecipient'. For a detailed description of the 'mailRecipient' and 'mailGroup' object classes and associated attributes, refer to Appendices A and B.

Our MTA implementation looks for the above attributes, and uses them to answer the three fundamental questions considered above, as follows.

4.1. Mapping an Address to an LDAP Entry

To resolve Question (1), we take the recipient address from the SMTP "envelope", and see if there is exactly one LDAP entry that advertises that address as one of its valid addresses. Specifically, we search for an LDAP entry that has a 'mail' or 'mailAlternateAddress' attribute whose value is the address in question. The search is performed across all LDAP entries in a given directory subtree, which is configured in the MTA as its "base subtree" of interest.

If the above search fails, we may also perform a fallback search. Specifically, if the above search yields zero matches, we split the address in question at the "@" sign, yielding a "local part" and a "domain part". If the MTA configuration specifies that it is the final authority on messages addressed to the domain part in question (i.e., it has the authority to bounce messages addressed to that domain), then we search for an LDAP entry whose 'uid' attribute equals the local part. If we get exactly one match, then we regard this as a successful match.

In theory, the fallback search may not be required, but since our MTA rewrites envelopes to 'uid'@'mailHost' (as discussed in Section 4.3), it is clearly advantageous for receiving MTAs in this environment to be able to unconditionally recognize an address thusly rewritten.

4.2. Determining Whether or not to Perform Local Delivery

To resolve Question (2), we look up the LDAP entry's 'mailHost' attribute. If the value of this attribute matches the fully-qualified domain name (FQDN) specified in the MTA configuration, then the message is passed to the local MDA.

If the value of the 'mailHost' attribute does not match the MTA's FQDN, then the message is routed.

Absence of routing information in the LDAP entry (i.e., not having a 'mailHost' attribute) is considered invalid and will result in a bounce, unless the MTA determines that it is a 'mailGroup' object, or a 'mailRecipient' object with a 'mailForwardingAddress' attribute. These cases are considered valid so that any MTA can be eligible to perform the resending action on behalf of a mail group or a "forwarding-only" object.

4.3. Determining How to Route the Message

To resolve Question (3), we look up the LDAP entry's 'mailHost' and 'uid' attributes. The 'uid' attribute is normally present for a 'mailRecipient', and is not normally present for a 'mailGroup'. If the 'uid' attribute is present, we rewrite the recipient address in the SMTP envelope to 'uid'@'mailHost', i.e., we combine the respective values of these two attributes and add an "@" sign to formulate a new recipient address. If the 'uid' attribute is not present, we do not rewrite the recipient address.

The message is routed to the destination indicated in the 'mailHost' attribute. This may or may not mean that the MTA will open an SMTP connection to the host identified as the 'mailHost', since the MTA configuration may specify routing rules that prevent this (e.g., in sites that are configured so that all message traffic follows a fixed "star" topology). Also, some sites may use DNS MX records [6] for internal message routing. For example, an MTA "mail.example.com" may receive a message for "joe@example.com", find that the 'mailHost' for this account is "mars.eng.example.com", and then discover that mail for "*.eng.example.com" is MX'ed to "hub.eng.example.com", which will then be the "next hop".

5. Possible Ways to Generalize the Solution Implemented

The following are several ways our approach could be extended to make it more general. None of these suggestions are reflected in our existing implementation as of this writing. We have no specific plans to follow or not follow these suggestions in any subsequent implementation. The intent is to provide ideas as to what a more general approach might look like. Whether or not these ideas should be implemented, or should become the basis for a future standard, are

left as open questions.

5.1. More Flexible Envelope Rewrites

One might argue that it is not really necessary for MTAs to rewrite envelopes when performing intra-enterprise message routing. The argument is as follows. Taking an example from above, suppose Joe's account is on "mars.eng.example.com", and Joe's account advertises "joe@example.com" as one of its valid email addresses. One would expect that Joe's "home" MTA knows what Joe's valid email addresses are. When mail arrives on "mail.example.com" for "joe@example.com", and it finds Joe's LDAP entry that advertises this address, it should be able to route the message without rewriting the envelope under the assumption that Joe's "home" MTA (and other MTAs such as "hub.eng.example.com" that are "closer" to Joe's "home" MTA than "mail.example.com") can also correctly identify the address as belonging to Joe.

However, existing practice in sites that use SMTP-based messaging often includes the rewriting of addresses to be host-specific. In order to avoid going against existing practice, our MTA implementation rewrites envelopes to 'uid'@'mailHost', as explained above. This is a fixed behavior, and some sites may desire more flexibility.

One way to provide more flexibility is to add an attribute, say:

```
mailRoutingAddress      address for internal mail routing
```

This could be added to the 'mailRecipient' and 'mailGroup' object classes as a way to explicitly specify how to rewrite the envelope when routing a message. Then, if the 'mailRoutingAddress' is present, the envelope is rewritten to the indicated address, otherwise, the address is not rewritten. This provides flexibility for site-specific policy governing whether or not envelopes are rewritten, and if so, how they are to be rewritten. It obviates the fixed 'uid'@'mailHost' behavior in our implementation (see Section 4.3), because the same information can then be stored in the 'mailRoutingAddress' attribute.

It should be noted that if the 'mailRoutingAddress' attribute were used as described here, it should contain an address that is recognizable on the destination MTA as being an address of the mail account represented by the LDAP entry in question, to ensure that the search specified in Section 4.1 will succeed on the destination MTA.

Also, the 'uid'@'mailHost' search could be removed from the method specified in Section 4.1, but some sites may still regard this as a

desirable fallback, although in this case the reasons to keep it are more along the lines of the reasoning in Section 5.2.

One might observe that 'mailRoutingAddress' and 'mailHost' may be partially redundant, and, in general, it is desirable to avoid redundancy of information in the directory. Having both attributes would be useful, however, if for some reason a network administrator wanted to separately control "next-hop" determination and envelope rewrites. So if both attributes were present, 'mailHost' would determine where to route the message, and 'mailRoutingAddress' would determine how to rewrite the envelope. If only 'mailRoutingAddress' were present, then the right-hand side (the domain part) of the routing address would determine the next destination. If only 'mailHost' were present, then the envelope would not be rewritten, or be rewritten as dictated by the configuration of the MTA.

5.2. Localpart-only Searches

Our implementation performs searches on email addresses as complete addresses (e.g., "joe@example.com"). We do not split the address at the "@" sign and search on the "local part", except in the case characterized above as a "fallback" search. This approach is probably sufficient for most customers since they can always add more addresses to an account as needed. It also reduces the risk of "namespace crossovers" that could result if customers with multiple distinct domains (e.g., with "joe" being a different person in each domain) did localpart-only searches.

Nevertheless, some sites may desire the flexibility to configure their MTAs to perform "localpart-only" searches, once the MTA has ascertained that the domain part is considered to be "local". They may then want the search to attempt matches against arbitrary attributes, like 'uid', 'cn' (with spaces and other illegal characters matching underscores or dots in the address), or some attribute whose purpose is to contain localpart-only email addresses. Site-specific needs can vary considerably in this area, and the most appropriate solution may be to make this part of an MTA's functionality as configurable as possible.

5.3. New Object Class: 'mailableObject'

As currently implemented, the 'mailRecipient' object class provides (a) attributes used only by the final MTA, and (b) attributes that may be used by non-final MTAs, i.e., attributes relevant to the task of routing messages among the various MTAs within an organization. It may be advantageous to group the attributes in category (b) together as belonging to their own object class, say, 'mailableObject'. Then, the 'mailRecipient' and 'mailGroup' object

classes would include only category (a) attributes. An LDAP entry representing a user would then become a "mailable user" by mixing in the 'mailableObject' and 'mailRecipient' object classes, with their associated attributes. An LDAP entry that represents a group would become a "mailable group" by mixing in the 'mailableObject' and 'mailGroup' object classes.

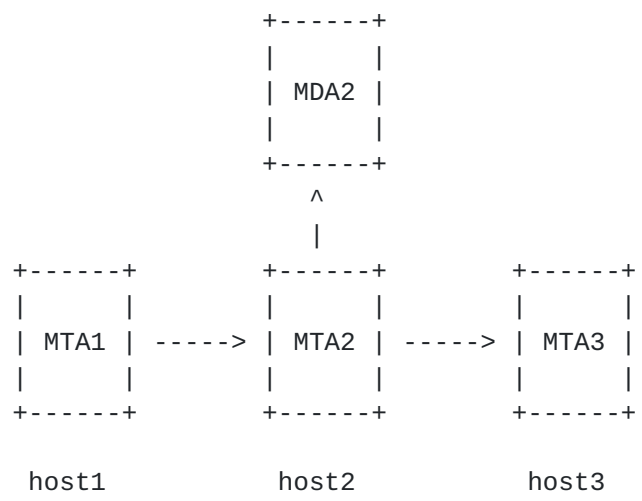
The proposed 'mailableObject' object class is defined as follows, with object identifiers as indicated:

```
Object class:  mailableObject
  (OID: 2.16.840.1.113730.3.2.34)
    Required attribute:
        objectClass
          (OID: 2.5.4.0)
    Allowed attributes:
        cn
          (OID: 2.5.4.3)
        mail
          (OID: 0.9.2342.19200300.100.1.3)
        mailAlternateAddress
          (OID: 2.16.840.1.113730.3.1.13)
        mailHost
          (OID: 2.16.840.1.113730.3.1.18)
        mailRoutingAddress
          (OID: 2.16.840.1.113730.3.1.47)
```

Here is an example of an LDAP entry for a mail user account, using the 'mailableObject' object class:

```
dn: cn=Joe Blow,o=Example Corporation,c=US
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: mailableObject
objectClass: mailRecipient
cn: Joe Blow
sn: Blow
uid: joeblow
userPassword: {crypt}jqYx78jsZxGrI
mail: joeblow@example.com
mailAlternateAddress: jojo@example.com
mailHost: host2.example.com
mailDeliveryOption: mailbox
```

Consider a network with three hosts that run MTAs:



The above illustrates a two-layer mail routing and delivery model. If a message addressed to "joeblow@example.com" arrives on host1, MTA1 need only look in the directory for an LDAP entry that is a 'mailableObject' and has an address "joeblow@example.com", determine whether it can handle mail for that LDAP entry locally, and if not, route the message, in this case, to host2. On host2, MTA2 will follow the same logic, but determine that the message can be handled locally. Thus, only MDA2 need be concerned with the type of mailable object ('mailRecipient' or 'mailGroup') and the attributes particular to that type of object (in this case 'mailDeliveryOption').

This implies that in a network environment containing LDAP-aware MTAs from different vendors, the MTAs can route messages to each others' accounts based on common interpretation of the 'mailableObject' object class and its attendant attributes. Object classes and attributes that represent the upper layer of handling (i.e., delivery to a mailbox, sending a vacation notice, forwarding to a different account, and mail group list expansion) do not require common interpretation as long as a specific host is handling mail for the LDAP entry in question, and so only that host (and that vendor's implementation) must understand the object classes and attributes at that level.

On the other hand, Section 4.2 specifies that in the absence of any routing information, an MTA can decide to handle the message locally in certain cases. This requires that there is either common interpretation of upper-layer attributes across all MTAs in the environment, or the MTAs that do not handle such cases are configured to route the messages to MTAs that do.

In short, the main advantage of the 'mailableObject' object class is to define a single object class that can serve to identify an LDAP entry as an entity to which email can be addressed, and to aggregate

the attributes that can provide interoperability at the MTA level.

Another example of this advantage is the case where LDAP entries have a 'mail' attribute but do not represent any kind of email account, i.e., the 'mail' attribute is being used for contact information only. For example, an LDAP entry representing a conference room might specify the telephone number and email address of the contact person for the room, and therefore may have a 'mail' attribute indicating the email address of the contact person. The contact person also has an LDAP entry, with the same 'mail' attribute, but since this person is a 'mailableObject', and the conference room is not, only the person's LDAP entry, and not the conference room's LDAP entry, is found on an address lookup such as the one specified in Section 4.1.

5.4. More Configurability

In lieu of a standard, mail server vendors could also achieve interoperability by providing a high degree of configurability in their products. For example, each mail server product could provide a means to configure or program its methods of resolving each of Questions (1), (2), and (3); if all of the mail servers in a given site were configured to use the same methods, then they would, in theory, interoperate in terms of LDAP-based SMTP message routing as described in this document. However, this approach to interoperability simply shifts the burden of standardization to the customer, and then there might still be a demand for a "recommended configuration profile" (i.e., a standard) for customers who desire solutions that work "right out of the box".

On the other hand, some level of configurability with regard to the functionality discussed here may be desirable.

6. Security Considerations

As in all cases where account information is stored in an LDAP-based directory service, network administrators must be careful to ensure that their directory service controls users' access to the entries and attributes stored therein, according to site policy (e.g., allowing users to modify, say, their 'mailForwardingAddress' attribute, but not their 'mailHost' attribute). Mail server products and their associated user management tools should help administrators to ensure this, and should also help administrators avoid configurations that would result in misdelivered mail due to "namespace crossovers" and other reasons.

7. References

[1] W. Yeong, T. Howes, S. Kille, "Lightweight Directory Access Protocol", RFC 1777, March 1995.

[2] "Information Processing Systems - Open Systems Interconnection - The Directory: Overview of Concepts, Models and Service", ISO/IEC JTC 1/SC21, International Standard 9594-1, 1988.

[3] J. Postel, "Simple Mail Transfer Protocol", RFC 821, August 1982.

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

[5] P. Mockapetris, "Domain names - concepts and facilities", RFC 1034, November 1987.

[6] C. Partridge, "Mail routing and the domain system", RFC 974, January 1986.

[7] M. Smith, "Definition of the inetOrgPerson Object Class", Internet-Draft (work in progress), <draft-ietf-asid-inetorgperson-01.txt>, July 1997.

[8] B. Steinback, "Using LDAP for SMTP Mailing Lists and Aliases", Internet-Draft (work in progress), <draft-steinback-ldap-mailgroups-00.txt>, September 1997.

8. Author's Address

Hans Lachman
Netscape Communications Corp.
501 East Middlefield Road
Mountain View, CA 94043
USA

Phone: (650) 254-1900
EMail: lachman@netscape.com

Appendix A. mailRecipient Object Class and Attributes

The following is an informal description of the 'mailRecipient' object class and associated attributes. It was designed to be used as a "mix-in" object in combination with a person's LDAP entry (in our implementation an 'inetOrgPerson' entry [7]) to enable a person to be recognized and handled as a mail user.

Object class: mailRecipient
(OID: 2.16.840.1.113730.3.2.3)

```
       Required attribute:
           objectClass
             (OID: 2.5.4.0)
       Allowed attributes:
           cn
             (OID: 2.5.4.3)
                   Common name (person's full name).
           mail
             (OID: 0.9.2342.19200300.100.1.3)
                   "Primary" email address.  This is the address that
                   would likely be displayed by "white-pages" lookup
                   applications.  Must be a complete email address
                   (e.g., "joe@example.com").
           mailAccessDomain
             (OID: 2.16.840.1.113730.3.1.12)
                   Domains and IP addresses from which user may do POP
                   or IMAP login.
           mailAlternateAddress
             (OID: 2.16.840.1.113730.3.1.13)
                   Email addresses that are considered valid for this
                   user in addition to their 'mail' address.  Must be
                   complete email addresses.
           mailAutoReplyMode
             (OID: 2.16.840.1.113730.3.1.14)
                   Auto-reply mode, may be one of:  'vacation' (send
                   reply text to sender, but only once per vacation),
                   'reply' (send reply text unconditionally), or 'echo'
                   (like 'reply' but include original message in the
                   reply).
           mailAutoReplyText
             (OID: 2.16.840.1.113730.3.1.15)
                   Reply text to use with 'mailAutoReplyMode'.
           mailDeliveryOption
             (OID: 2.16.840.1.113730.3.1.16)
                   Mail delivery option, one or more of:  'mailbox'
                   (deliver to user's POP/IMAP mailbox), 'native'
                   (deliver with platform's native delivery method,
                   e.g., "/usr/bin/mail"), or 'program' (perform program
                   delivery).  There must be at least one
                   'mailDeliveryOption' and/or 'mailForwardingAddress',
                   otherwise, mail to this account is undeliverable.
           mailForwardingAddress
             (OID: 2.16.840.1.113730.3.1.17)
                   User-specifiable mail forwarding address(es).
           mailHost
             (OID: 2.16.840.1.113730.3.1.18)
                   Fully-qualified domain name of the MTA that is the
                   final SMTP destination for mail addressed to this
```

account. Used for routing (see Section 4.3), and also used to determine which LDAP entries represent accounts that are to be considered local to a given mail server (see Section 4.2).

mailMessageStore
(OID: 2.16.840.1.113730.3.1.19)
Identifier for the message store containing this user's POP/IMAP mailbox. Contains absolute path of the message store directory (may be some other identifier in the future).

mailProgramDeliveryInfo
(OID: 2.16.840.1.113730.3.1.20)
Command text for program delivery.

mailQuota
(OID: 2.16.840.1.113730.3.1.21)
Quota in bytes for user's POP/IMAP mailbox.

multiLineDescription
(OID: not assigned)
User-specifiable personal description (not really related to email; in the future, this attribute may be removed or defined in some other object class).

uid
(OID: 0.9.2342.19200300.100.1.1)
User's login name.

userPassword
(OID: 2.5.4.35)
User's password.

Appendix B. mailGroup Object Class and Attributes

The 'mailGroup' object class provides attributes that specify the members and configuration of a mail group, and is defined in a separate document [8].