Network Working Group
Request for Comments: 1711
Category: Informational

J. Houttuin
RARE

October 1994

Classifications in E-mail Routing

Status of this Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Abstract

This paper presents a classification for e-mail routing issues. It clearly defines commonly used terminology such as static routing, store-and-forward routing, source routing and others. Real life examples show which routing options are used in existing projects.

The goal is to define all terminology in one reference paper. This will also help relatively new mail system managers to understand the issues and make the right choices. The reader is expected to already have a solid understanding of general networking terminology.

In this paper, the word Message Transfer Agent (MTA) is used to describe a routing entity, which can be an X.400 MTA, a UNIX mailer, or any other piece of software performing mail routing functions. An MTA processes the so called envelope information of a message. The term User Agent (UA) is used to describe a piece of software performing user related mail functions. It processes the contents of a message's envelope, i.e., the header fields and body parts.

Table of Contents

1. Naming, addressing and routing

A name uniquely identifies a network object (without loss of generality, we will assume the ’object’ is a person).

Once a person’s name is known, it can be used as a key to determine his address.

An address uniquely defines where the person is located. It can normally be divided into a domain related part (e.g., the RFC 822 domainpart or in X.400 an ADMD or OU attribute) and a local or user related part (e.g., the RFC 822 localpart or in X.400 a DDA or Surname attribute). The domain related part of an address typically consists of several components, which normally have a certain hierarchical order. These domain levels can be used for routing purposes, as we will see later.

Once a person’s address is known, it can be used as a key to determine a route to that person’s location.

We will use the following definition of an e-mail route:

e-mail route	a path between two leaves in a directed Message Transfer System (MTS) graph that a message travels for one originator/recipient pair. (see Figure 1)

Note that, in this definition, the User Agents (UAs) are not part of the route themselves. Thus if a message is redirected at the UA level, a new route is established from the redirecting UA to the UA the message is redirected to.

The first and last leaves in a mail route are not always UAs. A route may start from a UA, but stop at a certain point because one of the MTAs is unable to take any further routing decisions. If this happens, a warning is generated by the MTA (not by a UA), and sent back to the originator of the undeliverable message. It may even happen that none of the leaves is a UA, for instance if a warning message as discussed above turns out to be undeliverable itself. The cautious reader may have noticed that this is a dangerous situation. Special precautions are needed to avoid loops in such cases (see [1]).

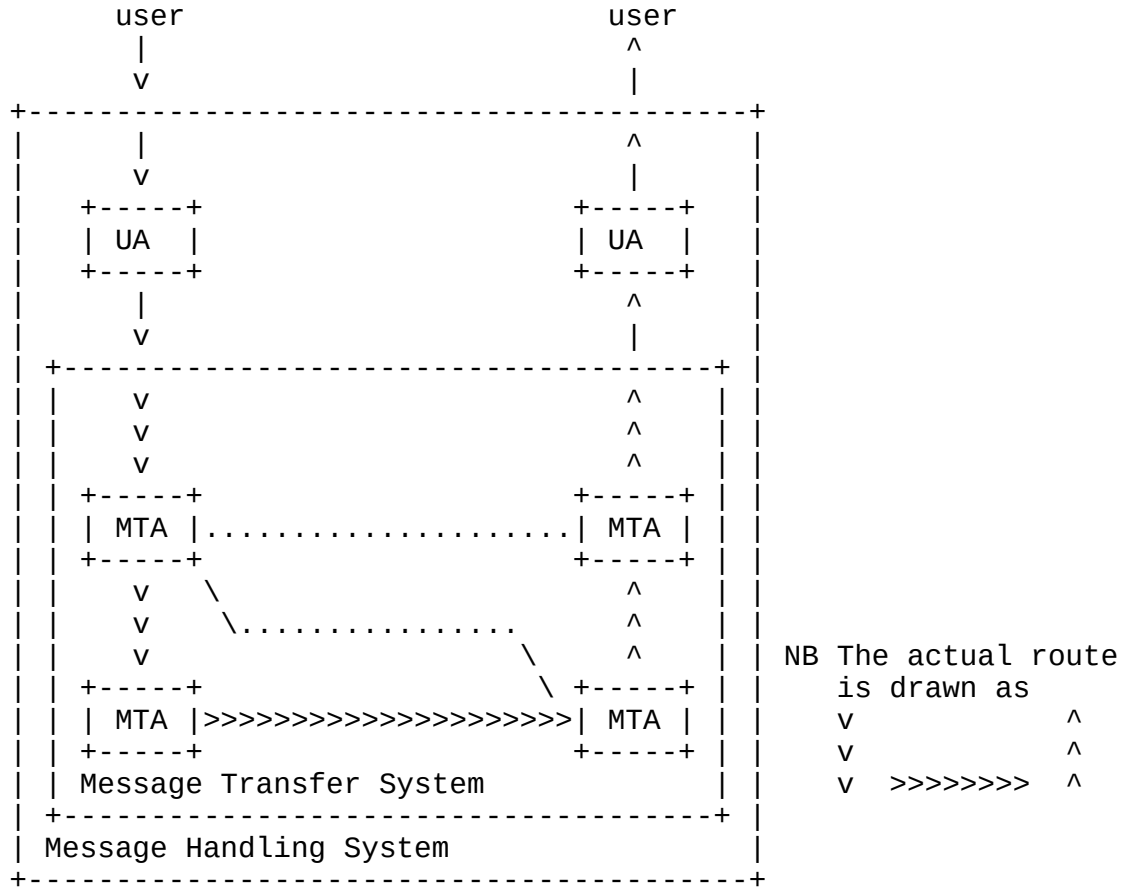

Figure 1. A mail route

It is important that the graph is directed, because routes are not necessarily symmetric. A reply to a message may be sent over a completely different mail route for reasons such as cost, non-symmetric network connectivity, network load, etc.

According to the definition, if a message has two different recipients, there will also be two mail routes. Since the delivery to a UA (not the UA itself) is a part of the route, this definition is still valid if two UAs are connected to the same MTA.

The words '.. for one originator-recipient pair.' in the definition do not imply that this pair provides the MTA with all necessary information to choose one specific route. One originator-recipient pair may give an MTA the possibility to choose from a number of possible routes, the so-called routing indicators (see chapter 2).

Other information (e.g., line load, cost, availability) can then be used to choose one route from the routing indicators.

Routing is defined as the process of establishing routes. Note that this is a distributed process; every intermediate MTA takes its own routing decisions, thus contributing to the establishment of the complete route.

Taking a routing decision is not a purely algorithmic process, otherwise there would hardly be any difference between an address and a route. The address is used as a key to find a route, typically in some sort of rule-based routing database. The possible options for realising this database and algorithms for using it are the subject of the rest of this paper.

2. Static versus dynamic

Dynamic (mail) routing allows a routing decision to be influenced by external factors, such as system availability, network load, etc. In contrast, static (mail) routing is not able to adapt to environmental constraints. Static routing can be viewed as an extremely simple form of dynamic routing, namely where there is only one choice for every routing decision.

Dynamic routing algorithms normally use some kind of distributed databases to store and retrieve routing information, whereas static routing is typically implemented in routing tables.

Note that dynamic routing can occur at different layers: at the mail level, dynamic routing might allow a message to be relayed to a choice of MTAs (the routing indicators). As an example, consider the Internet mechanism of using multiple Mail eXchanger (MX) records, describing MTAs that can serve a domain. If the primary choice MTA is not available, a second choice MTA can be tried. If this second choice MTA is busy, a third one will be tried, etc. On lower layers, there may be more than one presentation address for one MTA, each of which can again have an associated priority and other attributes.

These choices may represent that an MTA prefers to be connected to using one certain stack, e.g., RFC1006/TCP/IP, but is also able to accept incoming calls over another stack, such as TP0/CONS/X.25. We will call this dynamic stack routing. Theoretically, dynamic stack routing should be transparent to the mail routing application, and is thus not a part of dynamic mail routing. It is mentioned here because in existing products, dynamic stack routing is often very well visible at the mail configuration level, so MTA managers should at least be aware of it.

Although static routing is often table based, not all table based routing algorithms are necessarily static in nature. As a counter example, X.400 routing according to RFC 1465 [2] is clearly table based, but at the same time allows a fairly dynamic kind of mail routing (as well as dynamic stack routing, which in this approach is cleanly separated from the dynamic mail routing part). A mail domain can specify a choice of so-called RELAY-MTAs (formerly called WEPs) that will serve it, each with a priority and maximum number of retries.

For reasons of flexibility and reliability, dynamic routing is almost always the preferred method.

3. Direct versus indirect

Direct routing allows the originator’s MTA to contact the recipient’s MTA directly, whereas indirect routing (also known as store-and-forward routing) uses intermediate MTAs to relay the message towards the recipient. It is difficult to clearly distinguish between direct and indirect routing: direct routing assumes the existence of a fully meshed routing topology, whereas indirect routing assumes the existence of a more tree-like hierarchical topology. Mail routing in most existing networks is upto some degree indirect. Networks can be classified as being more or less direct according for the following rule of thumb: larger fan out of the routing tree means more direct routing, greater depth of the tree means less direct routing. Two kinds of indirect routing are presented here: firewalls (downstream) and default routes (upstream).

3.1. Firewalls

A firewall ’attracts’ all messages for a certain set of addresses (the address sub space behind the firewall) from the outside e-mail world to a central relaying MTA (the firewall). This is done by publishing routes to all other MTAs that must relay their messages over this firewall (the attracted community). Note that local knowledge should be used to route messages within the address space behind the firewall. An example for this is presented later on. There

exist many reasons for using firewalls, e.g., security considerations or to concentrate the management for a given domain onto one well managed system.

The Internet mail system would allow all mail hosts connected to the Internet to directly accept mail from any other host, but not all hosts use this possibility. Many domains are hidden behind one or more 'Mail eXchanger' (MX), which offer to relay all incoming mail for those domains. The RELAY-MTAs mentioned earlier can also be considered firewall systems.

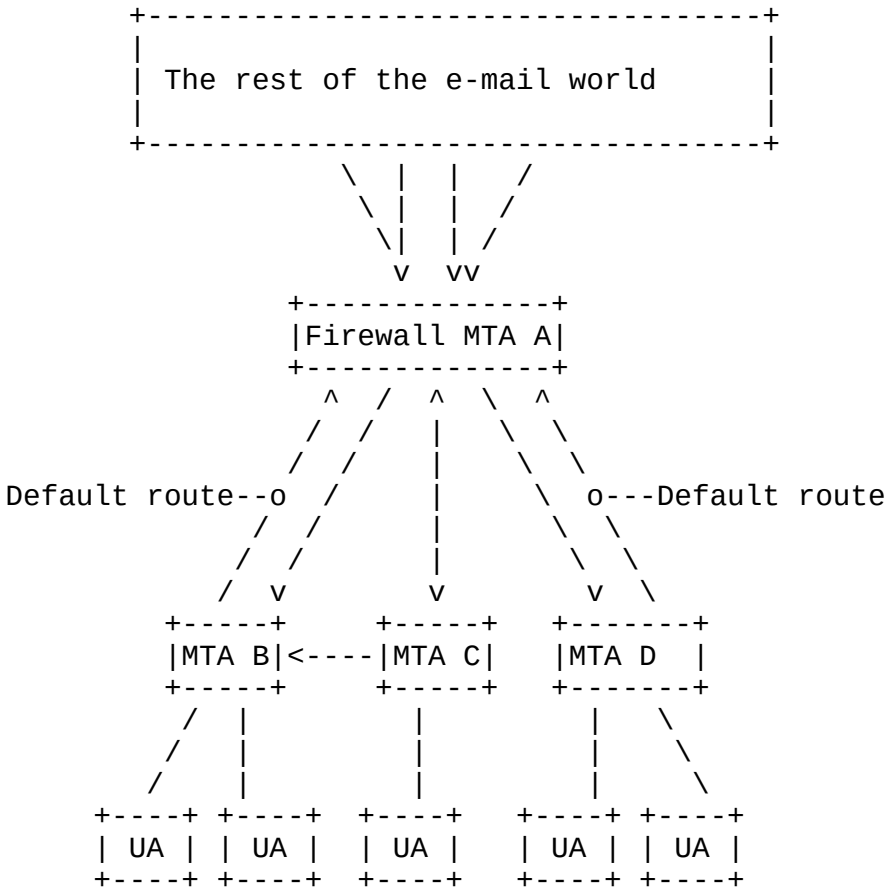

Figure 2. Firewall and default route

3.2. Default versus rule based

Default routing is to outgoing mail what a firewall is for incoming mail, and is thus often used in conjunction with firewalls. It is about the simplest routing algorithm one can think of: route every message to one and the same MTA, which is trusted to take further

care of routing the message towards its destination. Pure default routing is rather useless; it is normally used as a fall back mechanism accompanying a rule based algorithm.

For example, the simplest usable default algorithm is the following: first check if a mail should be delivered to a local UA. If not, perform default routing.

In order to avoid loops, it is not acceptable for all MTAs within a certain routing community (see chapter 9) to use default routing. At least one MTA should be able to access all routing rules for that community. Consider the following example: An X.400 MTA A, which serves the organisation organisational unit OU=orgunA within the organisation O=org, receives a message for the domain O=org; OU=orgunB;. Since MTA B in the same organisation serves all other OUs, A will default route the message to B. Suppose that B would use the same mechanism: first check if the OU is local and if not, default route to A. If OU=orgunC is not served by either A or B, this routing set-up will lead to a loop. The decision that a certain OU does not exist can only be made if at least one of the MTAs has knowledge of all existing OUs under O.

An example of a firewall and two default routes is shown in figure 2. It visualises that a firewall is a downstream and a default route is an upstream indirection. MTA B and D use default routing; they can only route to one other MTA, MTA A.

For more detailed information, please refer to [3], which lists most pros and cons of both approaches. Your choice will depend on many factors that are specific for your messaging environment.

4. Routing at user level

Normally a message is routed down to the deepest level domain, and then delivered to the recipient per default routing. I.e., every user in this domain is considered to have his mailbox uniquely defined within this domain on the same MTA, and every user on that MTA can be distinguished within this domain. Exceptions can occur when the users within a domain have their mailboxes on different MTAs (distributed domain), or when several domains exist on the same MTA (shared MTA).

4.1. Distributed domains

Routing is normally performed down to a certain domain level. Mail to all users that are directly registered under this domain is then delivered per default routing, i.e., delivered locally. Explicit user routing (i.e., rule-based routing on user level attributes according to a fixed table listing all users) may be necessary when not all

users have their UAs connected to the same MTA.

Note that the whole issue of distributed domains is nothing more than a special case of the problems discussed in chapter 3.2: 'Default versus rule-based'. The only reason for mentioning this in a separate chapter is that there are many software products that don't deal with routing based on local address parts in the same way as with routing based on domain related address parts.

As an example, consider an organisation where two mail platforms are available. Some users prefer using platform A, others prefer platform B. Of course, the easiest solution would be to create a subdomain A and a subdomain B, and then route domain A to system A and B to B. Default user routing on both platforms would then do the rest. However, when an organisation wants to present itself to the outside world using only one domain, this scheme cannot be used, at least not without special precautions (see the paragraph about avoiding loops in chapter 3.2).

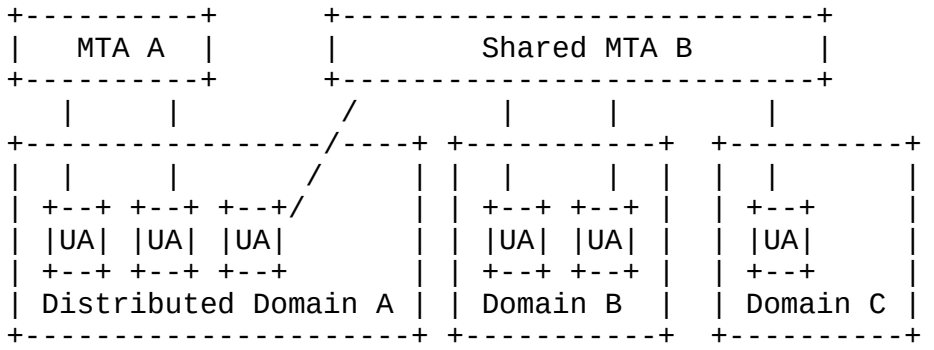

Figure 3. Distributed domains and shared MTAs

Another possibility to have uniform addresses in outgoing e-mail, despite the fact that a domain is distributed, is to make routing decisions on information in the local part of the address, e.g., in X.400 the Surname in exactly the same manner as making routing decisions on any other attributes. Thus products and routing algorithms that are able to route on user related address parts are said to support distributed domains.

4.2. Shared MTA

The opposite of a distributed domain is a shared MTA: several domains are routed locally on the same MTA. These domains sharing one MTA may cause problems when two or more domains have a local user with the same name.

Theoretically, this problem doesn't exist: the address is being routed down to the deepest domain level, and within that level, there will only be one user with that name (let's at least assume this for simplicity). Some products however use only one database of all users locally connected to this MTA instead of one database per domain, so that default user routing at the deepest level can lead to conflicts. It is beyond the scope of this document to describe the tricks needed to avoid these conflicts when using such products.

5. Routing control

Routing control means that routing decisions can be affected by the originator of a message. This normally takes the form of either granting or denying access for a certain user or group of users.

Routing control is often useful in an X.400 ADMD/PRMD environment, where it is either used to grant access only to users who are known to be chargeable, or where ADMDs can refuse messages that were relayed to them over international PRMD connections; a policy that is not allowed in the CCITT version of the standards (as opposed to the ISO version). Of course, the PRMDs can also perform routing control themselves in order to circumvent such problems.

Although there may be good reasons for using routing control, one must be aware that it can make the messaging environment unpredictable for end-users. Where using routing control is unavoidable, the originator whose message has been rejected is likely to appreciate receiving a message, clearly telling him where and why routing of his message was refused, whom to contact, and what options are available to avoid such rejections in the future.

6. Bulk routing

In order to reduce network traffic, intelligent mailers may prefer a message addressed to a group of remote users to be transferred to a remote domain only once, thus postponing the 'explosion' into several copies. This technique, called bulk routing, is especially useful when an MTA hosts large mailing lists.

Several possibilities exist. In a typical hierarchical firewall mail system, bulk routing can be done almost automatically by intelligent MTAs. For instance, in an X.400 community, a large international distribution list can create a message with an envelope containing 1000 recipient addresses, some of which can probably be grouped by the MTA depending on whether they can be routed further to the same remote MTA, according to the normal routing implementation at this MTA. The size and number of these groups will largely depend on how indirect this routing implementation is. In the GO-MHS community, the

number of groups will almost always be less than 50, which provides a rather fair distribution of traffic load over the involved MTAs (that is, fair according to the author's taste, who is not aware of any existing fair mail load distribution formula).

As an extreme example, the simplest way to automatically (i.e., without using special optimisation tools) bulk route mail is to use one default route. Any outgoing message, regardless of the number of recipients, will be routed over the default route only once. The default remote MTA will then have to break up the message (envelope) into several copies and is thus responsible for the actual explosion and distribution. NB. This mechanism can be exploited to shift the cost and overhead of exploding a message towards another domain/MTA. If you ever get a request for a bilateral default route agreement; i.e., the requesting party wants to default route over your MTA, it may be worth to check first if the requesting party is running or planning to run large mailing lists.

In more direct routing environments, such as BITNET, bulk routing will not function as automatically as described above. Without special precautions, an MTA would open a direct connection to every single host that occurs in the message's envelope, regardless of whether some of these hosts are far away from this MTA, but close to each other, measured by underlying network topology. This can clearly lead to a waste of expensive bandwidth. In order to be able to detect such cases, and to act upon it by sending one single copy over an expensive link and have it distributed at some remote hosts, an MTA must have additional knowledge of the relation between mail domains and the underlying network topology.

BITNET uses the distribute protocol [4] for this purpose. A selected set of hosts is published to have the required topology knowledge and to be able to efficiently distribute the mail on behalf of other MTAs, who can explicitly route all bulk mail to one of those hosts. The complete message, including the envelope, is encoded in a message body, which starts with a distribution request to the distribute server. This server will break up the rest of the body into the original envelope and contents and then use it's topology knowledge to efficiently distribute the original message. Note that this protocol violates the conceptual model of the layering of MTA and UA functionality, but it is about the only trick that will work in a very direct routing environment. It is only needed to overrule a non-efficient (for large mailing lists) routing topology.

Bulk routing is an area where mail routing issues start to overlap with the area of distributing netnews (bulletin board services). Several organisations, such as ISO, RARE and the IETF have started initiatives in the direction of harmonising the two worlds. The first

results, be it standards or products, are not expected before 1995 though.

7. Source routing

Source routing was originally intended to allow a user to force a message to take a certain route. The mechanism works as follows: the MTA that the user wants the message to be routed through is integrated into the address. Once the message has arrived at the specified MTA, that MTA strips itself from the source-routed address and routes the remaining address in the usual way. This mechanism is called explicit source routing and can be useful if a user wants to test a routing path or force a message to be routed over a faster, cheaper, more reliable, or otherwise preferred path.

For instance, if the Internet user user@uni-a.edu wants to test the mail connections to and from a remote domain uni-b.edu, he might source route a message to himself over the MTA at uni-b.edu by addressing the mail to: @uni-b.edu:user@uni-a.edu

Source routing need not always be explicit. Source routes can also be generated automatically by a gateway, in which case we speak of address rooting (to that gateway). The gateway will root itself to the message by putting its own domain in the source route mapped address, thus ensuring that any replies to the gatewayed message will be routed back through the same gateway.

Example 1: RFC 1327 left hand side encoding (see [5]) performed by the gateway 'gw.ch':

```
C=zz;A=a;P=p;O=oo;S=plork ->
"/C=zz/A=a/P=p/O=oo/S=plork/"@gw.ch
```

Example 2: RFC 1327 DDA mapping (see [5]) performed by the gateway C=zz;A=a;

```
bush@dole.us ->
DD.RFC-822=bush(a)dole.us;C=zz;A=a;
```

Example 3: the so-called %-hack:

```
user%final.domain@1st.relay
```

When the relaying host '1st.relay' receives the message, it strips its own domain part and interprets the localpart 'user%final.domain': it changes the % to an @ sign and relays the message to the address

```
user@final.domain
```

Example 4: Another example of the already mentioned explicit source routing, this time through two relays:

@1st.relay,@2nd.relay:user@final.domain

In the Internet, use of explicit source routing is strongly discouraged (see [6]), one reason being that not all mail relays will handle such addresses in a consistent manner. Apart from that, the need to use explicit source routing has disappeared over the last decennia. In earlier days, when the RFC 822 world consisted of many sparsely connected 'mail islands', source routing was sometimes needed to make sure that a message was routed through a gateway that was known to be connected to a remote island. Nowadays, the RFC 822 world is almost fully interconnected through the Internet, so the need for end-users to have knowledge of the mail network's topology has become superfluous.

8. Poor man's routing

If we combine static, indirect and source routing, we get what is commonly known as "poor man's routing". The user thus specifies the complete route in the address. A classic example is the old UUCP bang style addressing:

host1!host2!host3!host4!user

Poor man's routing is presented here for historical reasons only. Since, for reasons discussed earlier, most present networks discourage source routing and prefer dynamic over static routing, poor man's routing is not widely deployed anymore.

9. Routing communities

A routing community can be defined as follows:

Routing community:	a set of MTAs X, with the property that for any address a, every MTA in X except a subset Ya will have the option, according to an agreed upon set of routing rules, to directly route that address to at least one MTA in Ya.

Which is a rather formal way of describing that a routing community consists of a set of MTAs (and human operators) that agreed on a common set of rules on how to route messages among each other.

An example of a routing community is the large Internet routing community, in which the agreed rules are implemented in the Domain Name System (DNS). For details, refer to [7]. The subset Ya is in this case the set of MTAs that have an MX record in the DNS for a. MTAs that hide behind fire walls or behind default routes are thus not considered direct members of this community, but normally form their own smaller routing community, with one host (the mail exchanger/default route) belonging to both communities.

Another example is the GO-MHS community, consisting of a set of documented RELAY-MTAs (formerly called WEPs, Well-known Entry Points). Routing communities can be further classified depending on the openness and topology of their routing rules. [3] defines four classes of routing communities:

Local community:	The scope of a single MTA. Contains the MTAs view of the set of bilateral routing agreements, and routing information local to the MTA. Example: any local MTA.
Closed community:	This is like a local community, but involves more than one MTA. The idea is to route messages only within this closed community. A small subset of the involved MTAs can be in another community as well, in order to get the connectivity to the outside world, as described earlier. Example: A set of Private Management Domains (PRMDs) representing the same organisation in multiple countries.
Open community:	All routing information is public and any MTA is invited to use it. Example: the Internet.
Hierarchical community:	A subtree of the O/R address tree. Note that the subtree will in practice often be pruned; sub-sub-trees may form their own routing community. Example: GO-MHS.

This classification cannot always be followed too strictly. For example, completely closed communities are relatively rare. In order for e-mail to be an effective communication tool, an organisation will typically designate at least one of its MTAs as a gateway to

another routing community, for instance to the Internet. The organisation will register an Internet domain, say 'org.net', which points to this gateway, and thus acts as a firewall from the Internet to the domain 'org.net', and as a default route from the closed community to the rest of the Internet. At this stage, the gateway MTA can be regarded as being a member of any of the four types of routing communities. The reader is invited to check this himself.

Especially the distinction between open and closed communities is not always easy. To some extent, most routing communities are open, at least among their own participants. It is just that some routing communities are more open than others. Also, even the most open routing community is not just open to anyone. It is not enough for a community participant to use the community's routing rules and connect to any other MTA in the community. The participant will typically also have to fulfil an agreed upon set of operational requirements, for example the Internet host requirements [6] or the GO-MHS domain requirements [8].

The most open routing community known today is certainly the Internet mail community. As for X.400 routing communities, some problems occur when trying to open a community, the main one being that most X.400 software does not support the so called 'anonymous' connection mode, which allows any remote MTA to connect to it. Most software was designed or configured to use passwords for setting up MTA connections. This, together with the fact that X.400 routing was originally designed to be hierarchical, is one of the main reasons why most X.400 communities today are either closed or hierarchical.

10. Realisations

In this chapter some of the routing classifications described above are assigned to existing mail services and projects.

10.1. Internet mail

RFC 822 mail. An operational service. Co-ordination: distributed. Mostly dynamic routing, although static routing is also possible. DNS based routing rules(*). Mostly direct routing, although indirect is also possible. No dynamic stack routing. Distributed domains possible. Shared MTAs possible, but rare. Routing control not normally used. Bulk routing via SMTP envelope grouping; also possible, but not widely deployed, using the 'distribute protocol' [4]. Source routing supported, but strongly discouraged. No poor man's routing. Open (and hierarchical) routing community.

(*) Sub-communities don't use DNS based routing: The MX records in the DNS are used to "attract" messages from the Internet to the

"border" between the Internet and the sub-community. Thus from the Internet we have dynamic, directory based routing but once the "border" is reached, it is no longer possible to use MX records for mail routing, and thus some form of static routing is generally needed.

10.2. UUCP

RFC 822 style mail. An operational service. Co-ordination: distributed. Mostly static routing, although dynamic routing is also possible. Table based routing rules. Mostly indirect routing. No dynamic stack routing. No distributed domains. Shared MTAs possible, but rare. Routing control not normally used. No bulk routing possible. Source routing (poor man's routing) still widely used by means of 'bang' addressing, but strongly discouraged. Open (and hierarchical) routing community.

10.3. EARN

BITNET mail. An operational service. Co-ordination: The EARN Office, France. Static routing. Table based routing rules, although an X.500 based experiment is running. Mostly direct routing, although indirect is also possible. No dynamic stack routing. No distributed domains. No shared MTAs. Routing control not normally used. Bulk routing possible using the 'distribute protocol' [4]. Source routing not supported. No poor man's routing. Open routing community.

10.4. GO-MHS

X.400 mail. An operational service. Co-ordination: GO-MHS Project Team, Switzerland. Mostly static routing, although dynamic routing is getting more and more deployed since the introduction of RFC 1465 [2]. Table based community-wide routing rules. Indirect routing.

Dynamic stack routing. Distributed domains possible. Shared MTAs. Routing control not normally used, only to avoid routing control problems when routing international traffic to ADMDs. Bulk routing using X.400 'responsibility' envelope flags. Source routing supported for gatewaying to the Internet. No poor man's routing. Hierarchical, but open, routing community.

10.5. ADMD infrastructure

X.400 mail. An operational service. Co-ordination: The joint Administrative Management Domains (ADMDs), typically operated by PTTs. Mostly static routing. Indirect routing. Table based bilateral routing rules. No dynamic stack routing. Distributed domains not supported. Shared MTAs. Routing control used to prohibit routing of

international traffic through PRMDs and to limit access to certain gateways. Bulk routing using X.400 'responsibility' envelope flags. Source routing possible for gatewaying to the Internet. No poor man's routing. Closed hierarchical routing community.

10.6. Long Bud

X.400 mail. A pilot project. Co-ordination: The IETF MHS-DS working group. Dynamic routing. X.500 based routing rules. Mostly indirect routing, although direct is also possible. Dynamic stack routing. Distributed domains. Shared MTAs. No routing control. Bulk routing using X.400 'responsibility' envelope flags. Source routing supported for gatewaying to the Internet. No poor man's routing. Open hierarchical routing community.

10.7. X42D

X.400 mail. An experiment. Co-ordination: INFN, Italy. Dynamic routing. DNS based routing rules as defined in [9]. Mostly indirect routing, although direct is also possible. Dynamic stack routing. No distributed domains. Shared MTAs. No routing control. Bulk routing using X.400 'responsibility' envelope flags. Source routing supported for gatewaying to the Internet. No poor man's routing. Open hierarchical routing community.

11. Conclusion

We have seen several dimensions in which mail routing can be classified. There are many more issues that were not discussed here, such as how exactly the routing databases are implemented, which algorithms to use for making the actual choices in dynamic routing, etc. A follow-up paper is planned to discuss such aspects in more detail.

So far, the author has tried to keep this paper free of opinion, but he would like to conclude by listing his own favourite routing options (without any further explanation or justification; please feel free to disagree):

Static/dynamic:	Dynamic
Direct/indirect:	Every routing community has its own optimum level of indirection
User routing:	Support
Routing control:	Avoid
Bulk routing:	Efficient distribution should be transparent at mail level, but we may need better e-mail models before this becomes possible

Source routing:	Avoid where possible
Poor man's routing:	Avoid

12. Abbreviations

ADMD	Administration Management Domain
CCITT	Comite Consultatif International de Telegraphique et Telephonique
CONS	Connection Oriented Network Service
DDA	Domain Defined Attribute
DNS	Domain Name System
GO-MHS	Global Open MHS
IP	Internet Protocol
ISO	International Organisation for Standardisation
Long Bud	Not an abbreviation
MHS	Message Handling System
MHS-DS	MHS and Directory Services
MTA	Message Transfer Agent
MTS	Message Transfer System
MX	Mail eXchanger
O/R address	Originator/Recipient address
PP	Not an abbreviation
PRMD	Private Management Domain
RARE	Reseaux Associes pour la Recherche Europeenne
RFC	Internet Request for Comments
RTR	RARE Technical Report
SMTP	Simple Mail Transfer Protocol
STD	Internet Standard RFC
TCP	Transfer Control Protocol
TP0	Transport Protocol Class 0
UA	User Agent
UUCP	UNIX to UNIX CoPy
WEP	Well-known Entry Point

13. References

[1] Houttuin, J., "C-BoMBS : A Classification of Breeds Of Mail Based Servers", RARE WG-MSG Work in Progress, April 1994.

[2] Eppenberger, E., "Routing Coordination for X.400 MHS Services Within a Multi Protocol / Multi Network Environment Table Format V3 for Static Routing", RFC 1465, SWITCH, May 1993.

[3] Kille, S., "MHS use of the Directory to support MHS routing", Work in Progress, July 1993.

[4] Thomas, E., "Listserv Distribute Protocol", RFC 1429, Swedish University Network, February 1993.

[5] Kille, S., "Mapping between X.400(1988) / ISO 10021 and RFC 822", RFC 1327, RARE RTR 2, University College London, May 1992.

[6] Braden, R., Editor, "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, USC/ Information Sciences Institute, October 1989.

[7] Partridge, C., "Mail Routing and the Domain System", STD 14, RFC 974, BBN, January 1986.

[8] Hansen, A. and R. Hagens, "Operational Requirements for X.400 Management Domains in the GO-MHS Community", Work in Progress, March 1993.

[9] Allocchio, C., Bonito, A., Cole, B., Giordano, S., and R. Hagens "Using the Internet DNS to Distribute RFC1327 Mail Address Mapping Tables", RFC 1664, GARR-Italy, Cisco Systems Inc, Centro Svizzero Calcolo Scientific, Advanced Network & Services, February 1993.

[10] Houttuin, J., "A Tutorial on Gatewaying between X.400 and Internet Mail", RFC 1506, RTR 6, RARE Secretariat, August 1993.

[11] Postel, J., "Simple Mail Transfer Protocol", STD 10, RFC 821, USC/Information Sciences Institute, August 1982.

[12] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC 822, UDEL, August 1982.

[13] Alvestrand, H.T., et al, "Introducing Project Long Bud Internet Pilot Project for the Deployment of X.500 Directory Information in Support of X.400 Routing", Work in Progress, June 1993.

[14] Kille, S., "A Simple Profile for MHS use of Directory", Work in Progress, July 1993.

[15] Kille, S., "MHS use of the Directory to Support Distribution Lists", Work in Progress, November 1992.

[16] Eppenberger, U., "X.500 directory service usage for X.400 e-mail", Computer Networks for Research in Europe No.1: Computer Networks and ISDN Systems 25, Suppl.1 (1993) S3-8, September 1993.

[17] CCITT Recommendations X.400 - X.430. Data Communication Networks: Message Handling Systems. CCITT Red Book, Vol. VIII - Fasc. VIII.7, Malaga-Torremolinos 1984.

[18] CCITT Recommendations X.400 - X.420. Data Communication Networks: Message Handling Systems. CCITT Blue Book, Vol. VIII - Fasc. VIII.7, Melbourne 1988.

14. Security Considerations

Security issues are discussed in section 3.1.

15. Author's Address

Jeroen Houttuin
RARE Secretariat
Singel 466-468
NL-1017 AW Amsterdam
The Netherlands

Phone: +31 20 639 11 31
Fax: +31 20 639 32 89
EMail: houttuin@rare.nl