INTERNET DRAFT
RARE WG-MSG
Rev. 2.1

Jeroen Houttuin
RARE Secretariat
18 February 1993

RFC 1327 tutorial

Abstract

This tutorial was produced to help RFC 1327 novices, especially new gateway managers, to find their way into this complicated subject. End-users are encouraged to read the COSINE MHS pocket user guide [pug] instead.

The introduction is general enough to be read not only by gateway managers, but also by those who are new to e-mail in general. Parts of this introduction can be skipped as needed. To a certain extent, this document can also be used as a reference guide to X.400 <-> RFC 822 gatewaying. Wherever there is a lack of detail in the tutorial, it will at least refer to the corresponding chapters in other documents. As such, it shields the RFC 1327 novice from too much detail.

Status of this Memo

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a "working draft" or "work in progress."

Please check the I-D abstract listing contained in each Internet Draft directory to learn the current status of this or any other Internet Draft.

Distribution of this memo is unlimited.

Acknowledgements

This tutorial was originally produced by SWITCH within the context of the COSINE MHS contract. It is heavily based on other papers and books, such as [JH-92], [HTA-faq], [822], [1280], [1310], and [1327], from which large parts of text were reproduced (slightly edited) by kind permission from the authors.

Such reproduced paragraphs are labelled as : <[source]...... Text ......[source]>

Disclaimer

This document is not everywhere exact and or complete in describing the involved standards. Irrelevant details are left out and some concepts are simplified for the ease of understanding. For reference purposes, always use the original standards.

Format

This Internet Draft is available in ASCII as well as in Postscript format. Important features for a tutorial that are only available in the Postscript version include:

- Pictures
- Index
- Type styles

Please use the Postscript version whenever possible.

Contents

1. Introduction

This chapter describes the history, status, future, and contents of the involved standards.

<[JH-92] There is a major difference between mail systems used in the USA and Europe. Mail systems originated mainly in the USA, where their explosive growth started as early as in the seventies. Different company-specific mail systems were developed simultaneously, which, of course, led to a high degree of incompatibility. The Defense Advanced Research Projects Agency (DARPA) which had to use machines of many different manufacturers, triggered the development of the Internet and the TCP/IP protocol suite, which was later accepted as a standard by the US Department of Defense (DoD). The Internet mail format is defined in RFC 822 and the protocol used for exchanging mail is known as the simple mail transfer protocol (SMTP). Together with UUCP and the BITNET protocol NJE, SMTP has become one of the main de facto mail standards in the US.

Unfortunately, all these protocols were incompatible, which explains the need to come to an acceptable global mail standard. CCITT and ISO began working on a norm and their work converged in what is now known as the X.400 Series Recommendations. One of the objectives was to define a super set of the existing systems, allowing for easier integration later on. Some typical positive features of X.400 are the store-and-forward mechanism, the hierarchical address space and the possibility of combining different types of body parts into one message body.

In Europe, the mail system boom came later. Since there was not much equipment in place yet, it made sense to use X.400 as much as possible right from the beginning. A strong X.400 lobby existed, especially in West-Germany (DFN). In the R&D world, mostly EAN was used because it was the only public domain X.400 product at that time.

At the moment, the two worlds of X.400 and SMTP are moving towards each other. On the one hand, the American Department of Defense, one of the main forces behind the Internet, has decided that future networking should be based on ISO standards, implying a migration from SMTP to X.400. On the other hand X.400 users in Europe have a need to communicate with the Internet. Due to the large traffic volume between the two nets it is not enough interconnecting them with a single international gateway. The load on such a gateway would be too heavy. Direct access using local gateways is more feasible. A striking example of the opening-up of the ISO oriented world is the reorganisation of the RARE working groups in June 1992. The new working groups are now also discussing non ISO protocols, such as RFC 822.

Although the expected success of X.400 has been a bit disappointing (mainly because no good products were available), the future of e-mail systems must still be seen in the context of this standard.

And although in the long run X.400 is likely to take over the world of e-mail systems, SMTP cannot be neglected over the next few years. Especially the simple installation procedure and the high degree of connectivity will contribute to a growing number of RFC 822 installations in Europe in the near future.

1.1. What is X.400

In October 1984, the Plenary Assembly of the CCITT accepted a standard to facilitate international message exchange between subscribers to computer based store-and-forward message services. This standard is known as the CCITT X.400 series recommendations ([CCITT 84], from now on called X.400(84)) and happens to be the first CCITT recommendation that relates to the OSI application layer. It should be noted that X.400(84) is based on work done in the IFIP Working Group 6.5, and that ISO at the same time was proceeding towards a compatible document. However, the standardisation efforts of CCITT and ISO did not converge in time, to allow the publication of a common text.

X.400(84) triggered the development of software implementing (parts of) the standard in the laboratories of almost all major computer vendors and many software houses. Similarly, public carriers in many countries started to plan X.400(84) based message systems that would be offered to the users as value added services. Early implementations appeared shortly after first drafts of the standard were published and a considerable number of commercial systems are available nowadays.

X.400(84) describes a functional model for a Message Handling System (MHS) and associates services and protocols. The model illustrated in the Figure 2.1. defines the components of a distributed messaging system:

Users in the MHS environment are provided with the capability of sending and receiving messages. Users in the context of an MHS may be humans or application processes. The User Agent (UA) is a process that makes the services of the MTS available to the user. A UA may be implemented as a computer program that provides utilities to create, send, receive and perhaps archive messages. Each UA, and thus each user, is identified by a name (each user has its own UA).

The Message Transfer system (MTS) transfers messages from an originating UA to a recipient UA. As implied by the figure shown above, data sent from UA to UA may be stored temporarily in several intermediate Message Transfer Agents (MTA), i.e. a store-and-forward mechanism is being used. An MTA forwards received messages to a next MTA or to the recipient UA.

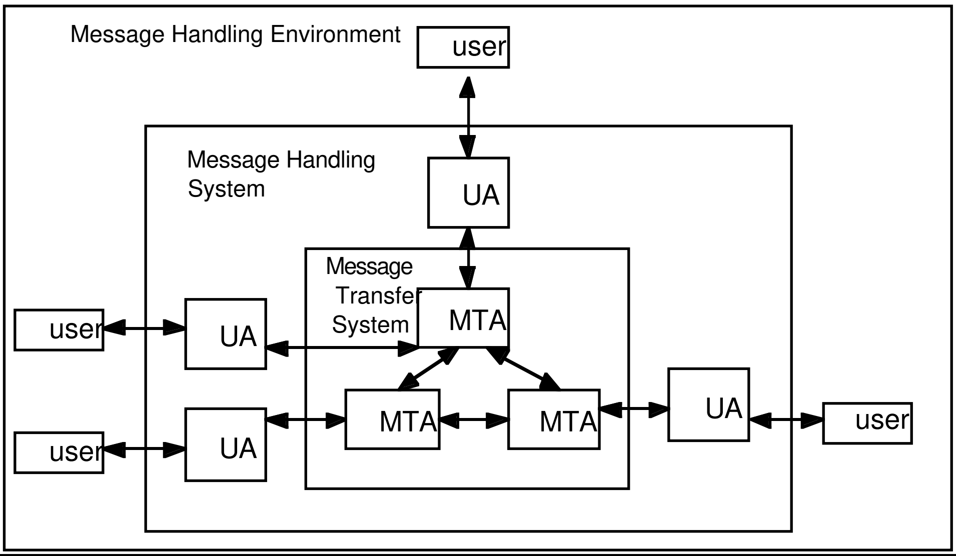

Fig. 2.1.

X.400(84) divides layer 7 of the OSI Reference Model into 2 sublayers, the User Agent Layer (UAL) and the Message Transfer Layer (MTL) as shown in the following figure:

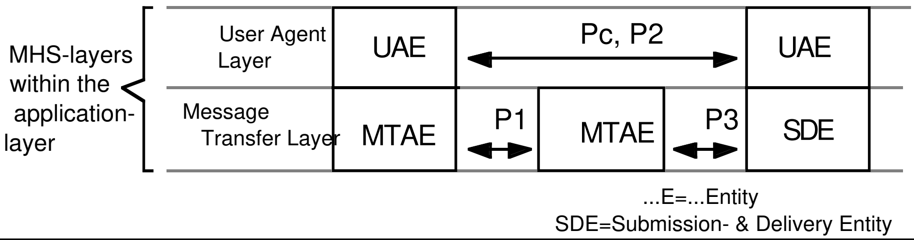

Fig. 2.2.

The MTL is involved in the transport of messages from UA to UA, using one or several MTAs as intermediaries. By consequence, routing issues are entirely dealt with in the MTL. The MTL in fact corresponds to the postal service that forwards letters consisting of an envelope and a content. Two protocols, P1 and P3, are used between the MTL entities (MTA Entity (MTAE), and Submission and Delivery Entity (SDE)) to reliably transport messages. The UAL embodies peer UA Entities (UAE), which interpret the content of a message and offer specific services to the application process. Depending on the application to be supported on top of the MTL, one of several end-to-end protocols (Pc) is used between UAEs. For electronic mail, X.400(84) defines the protocol P2 as part of the InterPersonal Messaging Service (IPMS). Conceivably other UAL protocols may be defined, e.g. a protocol to support the exchange of electronic business documents.

The structure of an InterPersonal Message (IPM) can be visualised as follows (Note that the envelope is not a part of the IPM; it is generated by the MTL):

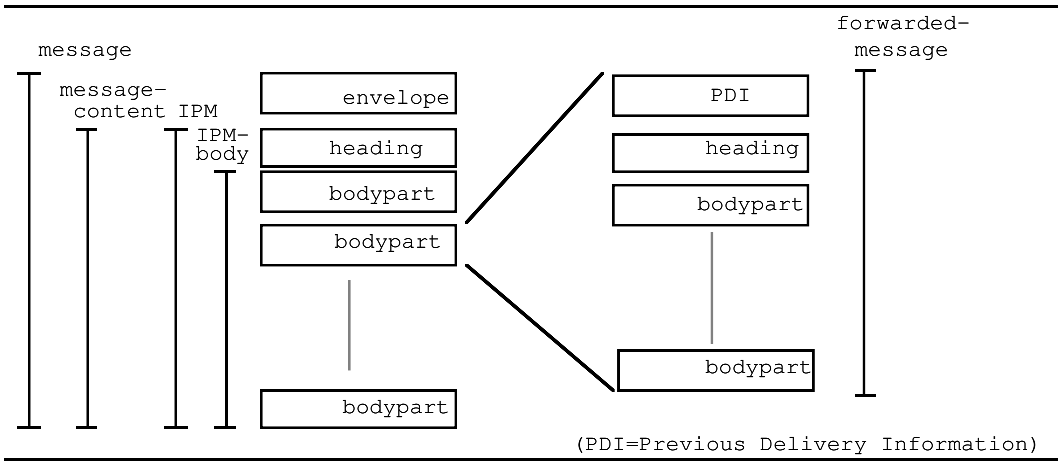

Fig. 2.3.

An IPM heading contains information that is specific for an interpersonal message like 'originator', 'subject', etc. Each bodypart can contain one information type, text, voice or as a special case, a forwarded message. A forwarded message consists of the original message together with Previous Delivery Information (PDI), which is drawn from the original delivery envelope.

Early experience with X.400(84) showed that the standard had various shortcomings. Therefore CCITT, in parallel with ISO, corrected and extended the specification during its 1984 to 1988 study period and produced a revised standard ([CCITT/ISO 88]), which was accepted at the 1988 CCITT Plenary Meeting ([BP-88]). Amongst others, X.400(88) differs from X.400(84) in that it defines a Message Store (MS), which can be seen as a kind of database for messages. An MS enables the end-user to run a UA locally, e.g. on a PC, whilst the messages are stored in the MS, which is co-located with the MTA. The MTA can thus always deliver incoming messages to the MS instead of to the UA. The MS can even automatically file incoming messages according to certain criteria. Other enhancements in the 88 version affect security and distribution lists. [JH-92]>

As for its relevance, X.400 is the only non-proprietary standard for interchange of electronic mail that has the sanction of an official standards body. The status of the 2 main flavours is:

- X.400(84): This is what most implementations today in fact run.

- X.400(88): Although this version has clear advantages over X.400(84), the number of systems implementing it has been rather disappointing. X.400(88) is also an International Standard (called 'MOTIS') by ISO.

1.2. What is an RFC

<[1310] The Internet, a loosely-organised international collaboration of autonomous, interconnected networks, supports host-to-host communication through voluntary adherence to open protocols and procedures defined by Internet Standards. There are also many isolated internets, i.e., sets of interconnected networks, that are not connected to the Internet but use the Internet Standards. The architecture and technical specifications of the Internet are the result of numerous research and development activities conducted over a period of two decades, performed by the network R&D community, by service and equipment vendors, and by government agencies around the world.

In general, an Internet Standard is a specification that is stable and well-understood, is technically competent, has multiple, independent, and interoperable implementations with operational experience, enjoys significant public support, and is recognisably useful in some or all parts of the Internet.

The principal set of Internet Standards is commonly known as the "TCP/IP protocol suite". As the Internet evolves, new protocols and services, in particular those for Open Systems Interconnection (OSI), have been and will be deployed in traditional TCP/IP environments, leading to an Internet that supports multiple protocol suites.

The Internet Activities Board (IAB) is the primary co-ordinating committee for Internet design, engineering, and management [1]. The IAB has delegated to its Internet Engineering Task Force (IETF) the primary responsibility for the development and review of potential Internet Standards from all sources. The IETF forms Working Groups to pursue specific technical issues, frequently resulting in the development of one or more specifications that are proposed for adoption as Internet Standards.

Final decisions on Internet standardisation are made by the IAB, based upon recommendations from the Internet Engineering Steering Group (IESG), the leadership body of the IETF. IETF Working Groups are organised into areas, and each area is co-ordinated by an Area Director. The Area Directors and the IETF Chairman are included in the IESG. [1310]>

Any individual or group (e.g. an IETF working group) can submit a document as a so-called Internet Draft. After at least half a year, if the document, being well discussed, looks stable, the IESG may propose to the IAB to turn the Internet-Draft into a 'Requests For Comments' (RFC). RFCs cover a wide range of topics, from early discussion of new research concepts to status memos about the Internet. All Internet Standards are published as RFCs, but not all RFCs specify standards.

As an example, this tutorial is also an Internet Draft that is to become an Informational RFC later on.

Once a document is assigned an RFC number and published, that RFC is never revised or re-issued with the same number.

1.3. What is RFC 822

<[822] RFC 822 defines a standard for the format of Internet text messages. Messages consist of lines of text. No special provisions are made for encoding drawings, facsimile, speech, or structured text. No significant consideration has been given to questions of data compression or to transmission and storage efficiency, and the standard tends to be free with the number of bits consumed. For example, field names are specified as free text, rather than special terse codes.

A general "memo" framework is used. That is, a message consists of some information in a rigid format (the 'headers'), followed by the main part of the message (the 'body'), with a format that is not specified in RFC 822. It does define the syntax of several fields of the headers section; some of these fields must be included in all messages. [822]>

<[1327] RFC 822 is used in conjunction with a number of different message transfer protocol environments (822-MTSs).

- SMTP Networks: On the Internet and other TCP/IP networks, RFC 822 is used in conjunction with two other standards: RFC 821, also known as Simple Mail Transfer Protocol (SMTP) [821], and RFC 920 which is a Specification for domains and a distributed name service [920].

- UUCP Networks: UUCP is the UNIX to UNIX CoPy protocol, which is usually used over dialup telephone networks to provide a simple message transfer mechanism.

- BITNET: Some parts of Bitnet and related networks use RFC 822 related protocols, with EBCDIC encoding.

- JNT Mail Networks: A number of X.25 networks, particularly those associated with the UK Academic Community, use the JNT (Joint Network Team) Mail Protocol, also known as Greybook.

RFC 822 is based on the assumption that there is an underlying service, which in RFC 1327 is called the 822-MTS service. The 822-MTS service provides three basic functions:

1. Identification of a list of recipients.
2. Identification of an error return address.
3. Transfer of an RFC 822 message.

It is possible to achieve 2) within the RFC 822 header. Some 822-MTS protocols, in particular SMTP, can provide additional functionality, but as these are neither mandatory in SMTP, nor available in other 822-MTS protocols, they are not considered here. Details of aspects specific to two 822-MTS protocols are given in Appendices B and C of RFC 1327. An RFC 822 message consists of a header, and content which is uninterpreted ASCII text. The header is divided into fields, which are the protocol elements. Most of these fields are analogous to P2 heading fields, although some are analogous to MTS Service Elements. [1327]>

1.4. What is RFC 1327

Before describing RFC 1327 in more detail, it is useful to quickly compare RFC 822 with X.400 <[HTA-faq]:

RFC 822 has got:

- Simplicity
- Wide acceptance
- Large user base
- Public domain and commercial implementations
- Public domain and commercial user interfaces
- History

X.400 has got:

- Acceptance in the standards communities
- Commercial vendors of service
- Defined ways to transfer things other than ASCII text (but only a few implementations have implemented it)
- Standard notifications of delivery to user's mailbox and notification of a message being read by the user (these ones are often implemented, too!)
- Future [HTA-faq]>

<[1327] There is a large community using RFC 822 based protocols for mail services, who will wish to communicate with users of the InterPersonal Messaging Service (IPMS) provided by X.400 systems, and the other way around. This will also be a requirement in cases where RFC 822 communities intend to make a transition to use X.400, as conversion will be needed to ensure a smooth service transition. It is expected that there will be more than one gateway, and RFC 1327 will enable them to behave in a consistent manner. Note that the term gateway is used to describe a component performing the protocol mappings between RFC 822 and X.400. This is standard usage amongst mail implementors, but should be noted carefully by transport and network service implementors.

RFC 1327 describes a set of mappings that will enable interworking between systems operating X.400(both 84 and 88) and systems using RFC 822, or protocols derived from RFC 822. The approach of RFC 1327 aims to maximise the services offered across the boundary, whilst not requiring unduly complex mappings. The mappings should not require any changes to end systems. [1327]>

Some words about the history of RFC 1327: It started out in June 1986, when RFC 987 defined for X.400(84) what RFC 1327 defines for X.400(84 and 88). RFC 1026 added a number of additions and corrections to RFC 987. In December 1989, RFC 1138, which had a very short lifetime, was the first one to deal with X.400(88). It was obsoleted by RFC 1148 in March 1990. Finally, in May 1992, RFC 1327 obsoleted all of its ancestors.

RFC 1327 describes all mappings in term of X.400(88). It describes how these mappings should be applied to X.400(84) systems in its Appendix G.

2. What must be mapped

Both RFC 822 and X.400 messages consist of certain service elements (such as 'originator', 'subject'). As long as a message stays within its own world, the behaviour of such service elements is well defined. An important goal for a gateway is to provide the highest possible service level when a message crosses the boundary between the two mail worlds.

RFC 1327 defines mappings between the different service elements. Some of these mappings are quite straight-forward, such as '822.Subject:' <-> 'IPMS.Subject' (IPMS = Inter-Personal Messaging Service), but there are also more complicated cases. Especially when certain service elements exist only in one of the two worlds (e.g. interpersonal notifications), or when service elements exist in both worlds, but with slightly different interpretations, some tricks may be needed to provide the service over the gateway border.

Apart from mapping between the service elements, a gateway must also map the types and values assigned to these service elements. Again, this may in certain cases be very simple, e.g. 'IA5 -> ASCII'. The most complicated example is mapping address spaces. The problem is that address spaces are not something static that can be defined within RFC 1327. Address spaces change continuously, and they are defined by certain addressing authorities, which are not always parallel in the RFC 822 and the X.400 world. A valid mapping between two addresses assumes however that there is 'administrative equivalence' between the two domains in which the addresses are (see also [MSG-93]).

<[1327] The following basic mappings are defined in RFC 1327. When going from RFC 822 to X.400, an RFC 822 message and the associated 822-MTS information is always mapped into an IPM (MTA, MTS, and IPMS Services). Going from X.400 to RFC 822, an RFC 822 message and the associated 822-MTS information may be derived from:

- A Report (MTA, and MTS Services)

- An InterPersonal Notification (IPN) (MTA, MTS, and IPMS services)

- An InterPersonal Message (IPM) (MTA, MTS, and IPMS services) [1327]>

Probes (MTA Service) have no equivalent in RFC 821 or RFC 822 and are thus handled by the gateway. The gateways Probe confirmation should be interpreted as if the gateway were the final MTA to which the Probe was sent. Optionally, if the gateway uses RFC 821 as an 822-MTS, it may use the results of the 'VRFY' command to test whether it would be able to deliver (or forward) mail to the mailbox under probe.

MTS Messages containing Content Types other than those defined by the IPMS are not mapped by the gateway, and should be rejected at the gateway.

Some basic examples of mappings between service elements are listed below.

Service elements:

RFC 822	X.400
Reply-To:	IPMS.Heading.reply-recipients
Subject:	IPMS.Heading.subject
In-Reply-To:	IPMS.Heading.replied-to-ipm
References:	IPMS.Heading.related-IPMs
To:	IPMS.Heading.primary-recipients
Cc:	IPMS.Heading.copy-recipients

Service element types:

RFC 822	X.400
ASCII	PrintableString
Boolean	Boolean

Service element values:

RFC 822	X.400
oh_dear	oh(u)dear
False	00000000

There are some mappings between service elements that are rather tricky and enough important to mention in this tutorial. These are the mappings of origination-related headers and some envelope fields:

RFC 822 -> X.400:

- If Sender: is present, Sender: is mapped to IPMS.Heading.originator, and From: is mapped to

IPMS.Heading.authorizing-users. If not, From: is mapped to IPMS.Heading.originator.

X.400 -> RFC 822

- If IPMS.Heading.authorizing-users is present, IPMS.Heading.originator is mapped to Sender:, and IPMS.Heading.authorizing-users is mapped to From: . If not, IPMS.Heading.originator is mapped to "From:".

Envelope attributes

- RFC 1327 doesn't define how to map the MTS.OriginatorName and the MTS.RecipientName (often referred to as the P1.originator and P1.recipient), since this depends on which underlying 822-MTS is used. In the very common case that RFC 821 (SMTP) is used for this purpose, the mapping is normally as follows:

```
MTS.Originator-name     <->   MAIL FROM:
MTS.Recipient-name      <->   RCPT TO:
```

  This explains why there are no mapped equivalents for those envelope attributes visible in the heading of an RFC 822 message.

For more details, refer to RFC 1327, chapters 2.2 and 2.3.

3. Address mapping

As address mapping is often considered the most complicated part of mapping between service element values, this subject is given a separate chapter.

Both RFC 822 and X.400 have their own specific address formats. RFC 822 addresses are text strings (e.g. "plork@tlec.nl"), whereas X.400 addresses are binary (ASN.1) encoded sets of attributes with values. Such binary addresses can be made readable for a human user by a number of notations; for instance:

```
C=zz
ADMD=ade
PRMD=fhbo
O=a bank
S=plork
G=mary
```

The rest of this chapter deals with addressing issues and mappings between the two address forms in more detail.

3.1. X.400 addresses

As already stated above, an X.400 address is modelled as a set of attributes. Some of these attributes are mandatory, others are optional. Each attribute has a type and a value, e.g. the Surname attribute has type IA5text, and an instance of this attribute could have the value 'Kille'. Attributes are divided in Standard Attributes (SAs) and Domain Defined Attributes (DDAs).

X.400 defines four basic forms of addresses ([X.402(88), 18.5), of which the 'Mnemonic O/R Address' is the form that is most used, and is the only form that is dealt with in this tutorial. This is roughly the same address format as what in the 84 version was known as 'form 1, variant 1' ([X.400(84)] 3.3.2).

Standard Attributes

Standard Attributes (SAs) are attributes that all X.400 installations are supposed to 'understand' (i.e. use for routing), for example: 'country name', 'given name' or 'organizational unit'. The most commonly used SAs in X.400(84) are:

```
surName (S)
givenName (G)
initials (I*)
generationQualifyer (GQ)
OrganizationalUnits (OU*)
OrganizationName (O)
PrivateDomainName (PRMD)
AdministrationDomainName (ADMD)
CountryName (C)
```

The combination of S, G, I* and GQ is often referred to as the PersonalName (PN).

Although there is no hierarchy (of addressing authorities) defined by the standards, the following hierarchy is considered natural:

```
PersonalName < OU < OU <...< O < P < A < C
```

In addition to the SAs listed above, X.400(88) defines some extra attributes, the most important of which is

```
Common Name (CN)
```

CN can be used instead of or even together with PN. The problem in X.400(84) was that PN (S G I* GQ) was well suited to represent persons, but not roles and abstract objects, such as distribution lists. Even though postmaster clearly is a role, not someone's real surname, it is quite usual in X.400(84) to address a postmaster with S=postmaster. In X.400(88), the same postmaster would be addressed with CN=postmaster .

The attributes C and ADMD are mandatory (to be present), and may not be empty. At least one of the attributes PRMD, O, OU, PN and CN must be present.

PRMD and ADMD are often felt to be routing attributes that don't really belong in addresses. As an example of how such address attributes can be used for the purpose of routing, consider two special values for ADMD:

- ADMD=0; (zero) should be interpreted as 'the PRMD in this address is not connected to any ADMD'

- ADMD= ; (single SPACE) should be interpreted as 'the PRMD in this address is reachable via any ADMD in this country'. It is expected that ISO will express this 'any' value by means of a missing ADMD attribute in future versions of MOTIS. This representation can uniquely identify the meaning 'any', as a missing or empty ADMD field as such is not allowed.

Addresses are defined in X.400 using the Abstract Syntax Notation One (ASN.1). X.409 defines how definitions in ASN.1 should be encoded into binary format. Note that the meaning, and thus the ASN.1 encoding, of a missing attribute is not the same as that of an empty attribute. In addressing, this difference is often represented as follows:

- PRMD=; means that this attribute is present in the address, but its value is empty. Since this is not very useful, it's hardly ever being used. The only examples the author knows of were caused by mail managers who should have had this tutorial before they started defining their addresses :-)

- PRMD=@; means that this attribute is not present in the address.
  {NB. This is only necessary if an address notation (see below) requires that every single attribute in the hierarchy is somehow listed. Otherwise, a missing attribute can of course be represented by simply not mentioning it. This means that this syntax is mostly used in mapping rules, not by end users.}

Addresses that only contain SAs are often referred to as Standard Attribute Addresses (SAAs).

Domain Defined Attributes

Domain Defined Attributes (DDAs) were meant to have a meaning only within a certain context (originally this was intended to be the context of a certain management domain), such as a company context (for example: DDA type=internal-phone-nr value=9571). Such DDAs are often used along with the PN or CN attributes.

A bit tricky is the use of DDAs to encode service element types or values that are only available on one side of a service gateway. The most important examples of such usage are:

RFC 1327 (e.g. DDA type=RFC-822 value=u(u)ser(a)isode.com)

RFC 1328 (e.g. DDA type=CommonName value=mhs-discussion-list)

The RFC 1327 example will get more than enough attention later on in this tutorial. As for the second example: RFC 1328 defines the downgrading from X.400(88) to X.400(84), and the DDA of type CommonName is a straightforward trick to represent the attribute CN in an X.400(84) address.

In the context of RFC 1327 and RFC 1328, DDAs are normally used _instead of_ PN and CN, whose equivalents are implicitly encoded within the DDA (localpart=u_ser; CN=mhs-discussion-list).

Addresses that contain both SAs and DDAs are often referred to as DDA addresses.

X.400 address notation

X.400 only prescribes the binary encoding of addresses, it doesn't standardise how such addresses should be written on paper or what they should look like in a user interface on a computer screen. There exist a number of recommendations for X.400 address representation though.

- JTC proposes an annex to CCITT Rec. F.401 and ISO/IEC 10021-2, called 'representation of o/r addresses for human usage'. According to this proposal, an X.400 address would look as follows:

  G=jo; S=plork; O=a bank; OU1=owe; OU2=you; P=fhbo; A=ade; C=zz

  Note that in this format, the hierarchy of O and OUs is exactly the opposite of what one would expect intuitively (the hierarchy is increasing from left to right, except for the O and OUs, where it's right to left).

- Following what was originally used in the DFN-EAN software, most EAN versions today use an address representation similar to the JTC proposal, with a few differences:

  - natural ordering for O and OUs
  - no numbering of OUs.
  - allows writing ADMD and PRMD instead of A and P

The address in the example above could, in EAN, be represented as:

G=jo; S=plork; OU=you; OU=owe; O=a bank; P=fhbo; A=ade; C=zz

This DFN-EAN format is still often referred to as _the_ 'readable format'.

- RARE Working Group 1 (WG1) has made a recommendation that is very similar to the DFN-EAN format, but with the hierarchy reversed. Further ADMD and PRMD are used instead of A and P. This results in the address above to be represented as:

  C=zz; ADMD=ade; PRMD=fhbo; O=a bank; OU=owe; OU=you; S=plork; G=jo

  This format is recognised by most versions of the EAN software. In the R&D community, this is also the prevalent address representation for business cards, letter heads, etc. This is also the format that will be used for the examples in this tutorial.

- RFC 1327 defines a slash separated address representation:

  /G=jo/S=plork/OU=you/OU=owe/O=a bank/P=fhbo/A=ade/C=zz/

  Not only is this format used by the PP software, it is also widespread for business cards and letter heads in the R&D community.

- RFC 1327 finally defines yet another format for X.400 _domains_ (not for human users):

  OU$you.OU$owe.O$a bank.P$fhbo.A$ade.C$zz

  the main advantage of this format is that it is better machine-parseble than the others. This immediately implies its main disadvantage: it is barely readable for humans. Every attribute within the hierarchy should be listed, thus a missing attribute must be represented by the '@' sign (e.g. $a bank.P$@.A$ade.C$zz).

- Paul-André Pays (INRIA) has proposed a format that combines the readability of the JTC format with the parsebility of the RFC 1327 domain format. Although a number of operational tools within the GO-MHS community are already based on (variants of) this proposal, its future is still uncertain.

3.2. RFC 822 addresses

An RFC 822 address takes the form of an ASCII string of the following form:

localpart@domainpart

"domainpart" is sub-divided into

domainpart = sdom(n).sdom(n-1)....sdom(2).sdom(1).dom

"sdom" stands for "subdomain", "dom" stands for "top-level-domain".

"localpart" is normally a login name, and thus typically is a surname or an abbreviation for this. It can also be the address of a local distribution list or an alias that will allow redirecting mail (e.g. mary.plork@tlec.nl might be an alias for plork@tlec.nl). The localpart in the latter address may again be a surname alias for Mary's login name pl, so that the mail will eventually be delivered to pl@tlec.nl)

The hierarchy (of addressing authorities) in an RFC 822 address is as follows:

localpart < sdom(n) < sdom(n-1) <...< dom

Some virtual real-life examples:

joemp@tlec.nl
tsjaka.kahn@walhalla.diku.dk
a13_vk@cs.rochester.edu

In the above examples, 'nl', 'dk', and 'edu' are valid, registered, top level domains. Note that some networks that have their own addressing schemes are also reachable by way of 'RFC 822-like' addressing. Consider the following addresses:

```
oops!user           (a UUCP address)
V13ENZACC@CZKETH5A  (a BITNET address)
```

These addresses can be expressed in RFC 822 format:

user@oops.uucp
V13ENZACC@CZKETH5A.BITNET

Although the domains '.uucp', '.bitnet', and '.earn' are not officially registered, they are used in the Internet to express that the mail should be routed to a gateway.

As for mapping such addresses to X.400, there is no direct mapping defined between X.400 on the one hand and UUCP and BITNET on the other, so they are normally mapped to RFC 822 style first, and then to X.400 if needed.

3.3. RFC 1327 address mapping

Despite the difference in address formats, the address spaces defined by RFC 822 and X.400 are quite similar. The most important parallels are:

- both address spaces are hierarchical
- top level domains and country codes are often the same
- localparts and surnames are often the same

This similarity can of course be exploited in address mapping algorithms. This is also done in RFC 1327 (NB only in the exception mapping algorithm. See chapter 3.3.2).

Note that the actual mapping algorithm is much more complicated than shown below. For details, see RFC 1327, chapter 4.

3.3.1. Default mapping

The default RFC 1327 address mapping can be visualised as a function with input and output parameters:

```
      address information of the gateway performing the mapping
                                |
                                v
                       +-----------------+
   RFC 822 address <--->| address mapping | <---> X.400 address
                       +-----------------+
```

I.e. to map an address from X.400 to RFC 822 or vice versa, the only extra input needed is the address information of the local gateway.

3.3.1.1. X.400 -> RFC 822

There are two kinds of default address mapping from X.400 to RFC 822: one to map a real X.400 address to RFC 822, and another to decode an RFC 822 address that was mapped to X.400 (i.e. to reverse the default RFC 822 -> X.400 mapping).

To map a real X.400 address to RFC 822, the slash separated notation of the X.400 address (see chapter 3.1.) is mapped to 'localpart', and the local RFC 822 domain of the gateway that performs the mapping is used as the domain part. As an example, the gateway 'gw.switch.ch' would perform the following mappings:

```
C=zz; ADMD=ade; PRMD=fhbo; O=tlec; S=plork;   ->
/C=zz/ADMD=ade/PRMD=fhbo/O=tlec/S=plork/@gw.switch.ch
```

```
C=zz; ADMD=ade; PRMD=fhbo; O=a bank; S=plork  ->
"/C=zz/ADMD=ade/PRMD=fhbo/O=a bank/S=plork/"@gw.switch.ch
```

The quotes in the second example are mandatory if the X.400 address contains spaces, otherwise the syntax rules for the RFC 822 localpart would be violated.

This default mapping algorithm is generally referred to as 'left-hand-side encoding'.

To reverse the default RFC 822 -> X.400 mapping (see chapter 3.3.1.2): if the X.400 address contains a DDA of the type RFC-822, the SAs can be discarded, and the value of this DDA is the desired RFC 822 address (NB. Some characters in the DDA value must be decoded first. See chapter 3.3.1.2.). For example, the gateway 'GW.tlec.nl' would perform the following mapping:

```
DD.RFC-822=bush(a)dole.us; C=nl; ADMD=tlec; PRMD=GW
->
bush@dole.us
```

3.3.1.2. RFC 822 -> X.400

There are also two kinds of default address mapping from RFC 822 to X.400: one to map a real RFC 822 address to X.400, and another to decode an X.400 address that was mapped to RFC 822 (i.e. to reverse the default X.400 -> RFC 822 mapping).

To map a real RFC 822 address to X.400, the RFC 822 address is encoded in a DDA of type RFC-822 , and the SAs of the local gateway performing the mapping are added to form the complete X.400 address. This mapping is generally referred to as 'DDA mapping'. As an example, the gateway 'C=nl; ADMD=tlec; PRMD=GW' would perform the following mapping:

```
bush@dole.us    ->
DD.RFC-822=bush(a)dole.us; C=nl; ADMD=tlec; PRMD=GW
```

As for the encoding/decoding of RFC 822 addresses in DDAs, it is noted that RFC 822 addresses may contain characters (@ ! % etc.) that cannot directly be represented in a DDA. DDAs are of the (not so rich) type 'PrintableString', so these special characters need a special encoding. For details, refer to RFC 1327, chapter 3.4. Some examples:

```
100%name@address      ->    DD.RFC-822=100(p)name(a)address
u_ser!name@address    ->    DD.RFC-822=u(u)ser(b)(a)address
```

To decode an X.400 address that was mapped to RFC 822: if the RFC 822 address has a slash separated representation of a complete X.400 mnemonic O/R address in its localpart, that address is the result of the mapping. As an example, the gateway 'gw.switch.ch' would perform the following mapping:

```
/C=zz/ADMD=ade/PRMD=fhbo/O=tlec/S=plork/G=mary/@gw.switch.ch
->
C=zz; ADMD=ade; PRMD=fhbo; O=tlec; S=plork; G=mary
```

3.3.2. Exception mapping according to mapping tables

Chapter 3.3.1. showed that it is theoretically possible to use RFC 1327 with default mapping only. Although this provides a very simple, straightforward, way to map addresses, there are quite some good reasons not to use RFC 1327 this way:

- RFC 822 users are used to writing simple addresses of the form 'localpart@domainpart'. They often consider X.400 addresses, and thus also the left-hand-side encoded equivalents, as unnecessarily long and complicated. They would rather be able to address an X.400 user as if she had a 'normal' RFC 822 address. For example take the mapping

  ```
  C=zz; ADMD=ade; PRMD=fhbo; O=tlec; S=plork;     ->
  /C=zz/ADMD=ade/PRMD=fhbo/O=tlec/S=plork/@gw.switch.ch
  ```

  from chapter 3.3.1.1. RFC 822 users would find it much more 'natural' if this address could be expressed in RFC 822 as:

  ```
  plork@tlec.fhbo.ade.nl
  ```

- X.400 users are used to using X.400 addresses with SAs only. They often consider DDA addresses as complicated, especially if they have to encode the special characters, @ % ! etc, manually. They would rather be able to address an RFC 822 user as if he had a 'normal' X.400 address. For example take the mapping

  ```
  bush@dole.us
  ->
  DD.RFC-822=bush(a)dole.us;
  C=nl; ADMD= ; PRMD=tlec; O=gateway
  ```

  from chapter 3.3.1.2. X.400 users would find it much more 'natural' if this address could be expressed in X.400 as:

  ```
  C=us; ADMD=dole; S=bush
  ```

- Many organisations are using both RFC 822 and X.400 internally, and still want all their users to have a simple, unique address in both mail worlds. Note that in the default mapping, the mapped form of an address completely depends on which gateway performed the mapping. This also results in a complication of more technical nature:

- The tricky 'third party problem'. This problem must not necessarily be understood to read the rest of this chapter.

If it looks too complicated, please feel free to skip it until you are more familiar with the basics.

The third party problem is a routing problem caused by mapping. As an example for DDA mappings (the example holds just as well for left-hand-side encoding), consider the following situation (see Fig. 3.1.): RFC 822 user X in country A sends a message to two recipients: RFC 822 user Y, and X.400 user Z, both in country B:

```
From: X@A
To:   Y@B
      /C=B/.../S=Z/@GW.A
```

Since the gateway in country A maps all addresses in the message, Z will see both X's and Y's address as DDA-encoded RFC 822 addresses, with the SAs of the gateway in country A:

```
From: DD.RFC-822=X(a)A; C=A;....;O=GW
To:   DD.RFC-822=Y(a)B; C=A;....;O=GW
      C=B;...;S=Z
```

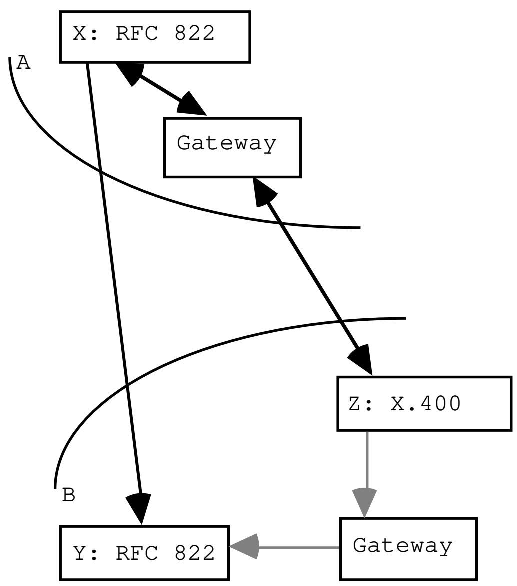

Fig. 3.1 The third party problem

Now if Z wants to 'group reply' to both X and Y, his reply to Y will be routed over the gateway in country A, even though Y is located in the same country:

```
From: C=B;...;S=Z
To:   DD.RFC-822=Y(a)B; C=A;....;O=GW
      DD.RFC-822=X(a)A; C=A;....;O=GW
```

The best way to travel for a message from Z to Y would of course have been over the gateway in country B:

```
From: C=B;...;S=Z
To:   DD.RFC-822=Y(a)B; C=B;....;O=GW
      DD.RFC-822=X(a)A; C=A;....;O=GW
```

The third party problem is caused by the fact that local gateway address information is mapped into addresses.

Ideally, the third party problem shouldn't exist. After all, address mapping affects addresses, and an address is not a route.... The reality is different however. For instance, very few X.400 products are capable to route messages on the contents of a DDA (actually, only RFC 1327 gateways will be able to interpret this type of DDA, and who says that the reply will pass a local gateway on its route back?). The same limitations hold for RFC 822 based mailers: most are not capable to make routing decisions on the content of a left-hand-side encoded X.400 address. So in practice, addressing (and thus also mapping) will very well affect routing.

To make mapping between the addresses more user friendly, and to avoid the problems shown above, RFC 1327 allows for overruling the default left-hand-side encoding and DDA mapping algorithms. This is done by specifying associations (mapping rules) between certain domainparts and X.400 domains. An X.400 domain consists of the domain-related SAs of Mnemonic O/R address (i.e. All SAs except PN and CN). The idea is to use the similarities between both address spaces, and directly map similar address parts onto each other. If, for the domain in the address to be mapped, an explicit mapping rule can be found, the mapping is performed between:

```
localpart      <->   PersonalName
domainpart     <->   X.400 domain
```

Only if no mapping rule can be found - i.e. the address mapping must fall back to its default algorithm - is the address information of the gateway performing the mapping used as an input parameter.

The complete mapping function can thus be visualised as follows:

```
      address information of the gateway performing the mapping
                                   |
                                   v
                          +------------------+
   RFC 822 address <--->| address mapping | <---> X.400 address
                          +------------------+
                                   ^
                                   |
                 domain associations (mapping rules)
```

3.3.2.1. PersonalName and localpart mapping

Since the mapping between these address parts is independent of the mapping rules that are used, and because it follows a simple, two-way algorithmic approach, this subject is discussed in a separate sub-chapter first.

The X.400 PersonalName consists of givenName, initials, and surName. RFC 1327 assumes that generationQualifyer is not used.

To map a localpart to an X.400 PN, the localpart is scanned for dots, which are considered delimiters between the components of PN, and also between single initials. In order not to put too much detail in this tutorial, only a few examples are shown here. For the detailed algorithm, see RFC 1327, chapter 4.2.1.

```
Marshall.Rose            <->   G=Marshall;S=Rose
M.T.Rose                 <->   I=MT;S=Rose
Marshall.M.T.Rose        <->   G=Marshall;I=MT;S=Rose
```

To map an X.400 PN to an RFC 822 localpart, take the non-empty PN attributes, put them into their hierarchical order (G I* S), and connect them with periods.

Some exceptions are caused by the fact that left-hand-side encoding can also be mixed with exception mapping. This is shown in more detail in the following sub-chapters.

3.3.2.2. Mapping between RFC 822 and X.400 domains

A mapping rule associates two domains: an X.400 domain and an RFC 822 domain. The X.400 domain is written in the RFC 1327 domain notation, so that both domains have the same hierarchical order. The domains are written on one line, separated by a '#' sign. For instance:

```
arcom.ch#ADMD$arcom.C$ch#
PRMD$tlec.ADMD$ade.C$nl#tlec.nl#
```

A mapping rule must at least contain a top level domain and a country code. If an address must be mapped, a mapping rule with the longest domain match is sought. The associated domain in the mapping rule is used as the domain of the mapped address. The remaining domains are mapped one by one following the natural hierarchy. Concrete examples are shown in the following sub-chapters.

3.3.2.2.1. X.400 -> RFC 822

As an example, consider the mapping rule:

```
PRMD$tlec.ADMD$ade.C$nl#tlec.nl#
```

Then the address C=nl; ADMD=ade; PRMD=tlec; O=you; OU=owe; S=plork

```
S      OU  O  PRMD  ADMD  Country
|      |   |  |     |     |
plork owe you tlec  ade   nl
```

would be mapped as follows. The Surname 'plork' is mapped to the localpart 'plork', see chapter 3.3.2.1. The domain 'PRMD$tlec.ADMD$ade.C$nl' is mapped according to the mapping rule:

```
localpart
   |  sdom3
   |    | sdom2
   |    |   |  sdom1
   |    |   |   |  top-level-domain
   |    |   |   |   |
plork@         tlec.nl
```

The remaining SAs (O and one OU) are mapped one by one following the natural hierarchy: O is mapped to sdom2, OU is mapped to sdom3:

```
localpart
   | sdom3
   |  | sdom2
   |  |   |  sdom1
   |  |   |   |  top-level-domain
   |  |   |   |    |
plork@owe.you.tlec.nl
```

Thus the mapped address is:

```
plork@owe.you.tlec.nl
```

The name of the file containing the listing of all such mapping rules, which is distributed to all gateways world-wide, is widely known under the following names:

```
'mapping 1'
'mapping table 1'
'map1'
'table 1'
'X2R'
```

As already announced, there is an exceptional case were localpart and PN are not directly mapped onto each other: sometimes it is necessary to use the localpart for other purposes. If the X.400 address contains attributes that would not allow for the simple mapping:

```
localpart       <->   PersonalName
domainpart      <->   X.400 domain
```

(e.g. spaces are not allowed in an RFC 822 domain, GQ and CN cannot be directly mapped into localpart, DDAs of another type than RFC-822), such attributes, together with the PN, are left-hand-side encoded. The domainpart must still be mapped according to the mapping rule as far as possible. This probably needs some examples:

```
C=nl; ADMD=ade; PRMD=tlec; O=owe; OU=you; S=plork; GQ=jr
->
/S=plork/GQ=jr/@you.owe.tlec.nl

C=nl; ADMD=ade; PRMD=tlec; O=o; OU=spc ctr; OU=u; S=plork
->
"/S=plork/OU=u/OU=spc ctr/"@o.tlec.nl
```

Note that in the second example, 'O=o' is still mapped to a subdomain following the natural hierarchy. The problems start with the space in 'OU=spc ctr'.

3.3.2.2.2. RFC 822 -> X.400

As an example, consider the mapping rule:

```
tlec.nl#PRMD$tlec.ADMD$ade.C$nl#
```

Then the address 'plork@owe.you.tlec.nl' :

```
localpart
   |  sdom3
   |    | sdom2
   |    |   |  sdom1
   |    |   |   |  top-level-domain
   |    |   |   |   |
plork@owe.you.tlec.nl
```

would be mapped as follows.

The localpart 'plork' is mapped to 'S=plork', see chapter 3.3.2.1.

The domain 'tlec.nl' is mapped according to the mapping rule:

```
S     OU  OU  O  PRMD  ADMD  Country
|                |     |    |
plork            tlec  ade  nl
```

The remaining domains (owe.you) are mapped one by one following the natural hierarchy: sdom2 is mapped to O, sdom3 is mapped to OU:

```
S     OU  OU  O  PRMD  ADMD  Country
|         |   |  |     |     |
plork     |   |  tlec  ade   nl
         owe you
```

Thus the mapped address is (in a readable notation):

C=nl; ADMD=ade; PRMD=tlec; O=you; OU=owe; S=plork

Had there been any left-hand-side encoded SAs in the localpart that didn't represent a complete mnemonic O/R address, the localpart would be mapped to those SAs. E.g.

```
"/S=plork/GQ=jr/OU=u/OU=spc ctr/"@o.tlec.nl
->
C=nl; ADMD=ade; PRMD=tlec; O=o; OU=space ctr;
OU=u; S=plork; GQ=jr
```

This is necessary to reverse the special use of localpart to left-hand-side encode certain attributes. See 3.3.2.2.1.

You might ask yourself by now why such rules are needed at all. Why don't we just use map1 in the other direction? The problem is that a symmetric mapping function (a bijection) would indeed be ideal, but it's not feasible. Asymmetric mappings exist for a number of reasons:

- To make sure that uucp addresses etc. get routed over local gateways.

- Preferring certain address forms, while still not forbidding others to use another form. Examples of such reasons are:

  - Fading out old address forms.

  - If an RFC 822 address is mapped to ADMD= ; it means that the X.400 mail can be routed over any ADMD in that country. One single ADMD may of course send out an address containing: ADMD=ade; . It must also be possible to map such an address back.

So we do need mapping rules from RFC 822 to X.400 too. The name of the file containing the listing of all such mapping rules, which is distributed to all gateways world-wide, is widely known under the following names:

```
'mapping 2'
'mapping table 2'
'map2'
'table 2'
'R2X'
```

If the RFC 822 localpart and/or domainpart contain characters that would not immediately fit in the value of a PN attribute (! % _), the mapping algorithm falls back to DDA mapping. In this case, the SAs that will be used are still determined by mapping the domainpart according to the mapping rule. In our case:

```
100%user@work.tlec.nl
->
DD.RFC-822=100(p)user(a)address.tlec.nl;
C=nl; ADMD=ade; PRMD=tlec; O=work
```

If no map2 rule can be found, a third table of rules is scanned: the gateway table. This table has the same syntax as mapping table 2, but its semantics are different. First of all, a domain that only has an entry in the gateway table is always mapped into an RFC 822 DDA. For a domain that is purely RFC 822 based, but whose mail may be relayed over an X.400 network, the gateway table associates with such a domain the SAs of the gateway to which the X.400 message should be routed. That gateway will then be responsible for gatewaying the message back into the RFC 822 world. E.g. if we have the gateway table entry:

```
gov#PRMD$gateway.ADMD$Internet.C$us#
```

(and we assume that no overruling map2 rule for the top level domain 'gov' exists), this would force all gateways to perform the following mapping:

```
bush@dole.gov
->
DD.RFC-822=bush(a)dole.gov;
C=us; ADMD=Internet; PRMD=gateway
```

This is very similar to the default DDA mapping, except the SAs are those of a gateway that has declared to be responsible for a certain RFC 822 domain, not those of the local gateway. And thus, this mechanism helps avoid the third party problem discussed in chapter 3.2.2.

The name of the table containing these gateway mapping rules, which is distributed to all gateways world-wide, is widely known under the following names:

```
'gate table'
'gateway table'
'GW'
```

3.4. Table co-ordination

As already stated, the use of mapping tables will only function smoothly if all gateways in the world use the same tables. On the global level, the collection and distribution of RFC 1327 address mapping tables is co-ordinated by the MHS Co-ordination Service:

SWITCH Head Office
MHS Co-ordination Service
Limmatquai 138
CH-8001 Zurich, Europe
Tel. +41 1 261 8112
Fax. +41 1 261 8133

RFC 822: project-team@switch.ch
X.400: C=ch;ADMD=arcom;PRMD=switch;O=switch;S=project-team;

The procedures for collection and distribution of mapping rules can be found on the MHS Co-ordination server, nic.switch.ch: in the directory /procedures . The server is available per FTP:

```
username: cosine
password: <your RFC 822 address>
```

If you want to define mapping rules for your own local domain, you can find the right contact person in your country or network (the gateway manager) on the same server, in the directory /mhs-services .

3.5. Local additions

Since certain networks want to define rules that should only be used within their networks, such rules should not be distributed world-wide. Consider two networks that both want to reach the top-level-domain 'arpa' over their local gateway. They would both like to use a mapping 2 rule for this purpose:

```
TLec in NL:     arpa#PRMD$gateway.ADMD$tlec.C$nl#

SWITCH in CH:   arpa#PRMD$gateway.ADMD$switch.C$ch#
```

(You may have noticed correctly that they should have defined such rules in the gateway table, but for the sake of the example, we assume they defined it in mapping table 2. This was the way things were done in the days of RFC 987, and many networks are still doing it this way these days.)

Since a mapping table cannot contain two mapping rules with the same domain on the left hand side, such 'local mappings' are not distributed globally. There exists a RARE draft proposal ([MSG-93]) which defines a mechanism for allowing and automatically dealing with conflicting mapping rules, but this mechanism has not been implemented as to date. After having received the global mapping tables from the MHS Co-ordination Service, many networks add 'local' rules to map2 and the gateway table before installing them on their gateways. Note that the reverse mapping 2 rules for such local mappings _are_ globally unique, and can thus be distributed world-wide. This is even necessary, because addresses that were mapped with a local mapping rule may leak out to other networks (here comes the third party problem again...). Such other networks should at least be given the possibility to map the addresses back. So the global mapping table 1 would in this case contain the two rules:

```
PRMD$gateway.ADMD$tlec.C$nl#arpa#
PRMD$gateway.ADMD$switch.C$ch#arpa#
```

Note that if such rules would have been defined as as local gate table entries instead of gate2 entries, there would have been no need to distribute the reverse mappings world-wide (the reverse mapping of a DDA encoded RFC 822 address is simply done by stripping the SAs, see 3.3.1.1.).

3.6. Product specific formats

Not all software uses the RFC 1327 format of the mapping tables internally. Almost all formats allow comments on a line starting with a # sign. Some examples of different formats:

RFC 1327

```
# This is pure RFC 1327 format
# table 1: X.400 -> RFC 822
#
PRMD$tlec.ADMD$ade.C$nl#tlec.nl#
# etc.

# table 2: RFC 822 -> X.400
#
arcom.ch#ADMD$arcom.C$ch#
# etc.
```

EAN

```
# This is EAN format
# It uses the readable format for X.400 domains and TABs
# to make a 'readable mapping table format'.
# table 1: X.400 -> RFC 822
#
P=tlec; A=ade; C=nl;        # tlec.nl
# etc.
```

```
# table 2: RFC 822 -> X.400
#
arcom.ch                    # A=arcom; C=ch;
# etc.
```

PP

```
# This is PP format
# table 1: X.400 -> RFC 822
#
PRMD$tlec.ADMD$ade.C$nl:tlec.nl
# etc.

# table 2: RFC 822 -> X.400
#
arcom.ch:ADMD$arcom.C$ch
# etc.
```

Most R&D networks have tools to automatically generate these formats from the original RFC 1327 tables, some even distribute the tables within their networks in several formats. If you need mapping tables in a specific format, please contact your national or R&D network's gateway manager. See chapter 3.4 .

3.7. Guidelines for mapping rule definition

Beware that defining mapping rules without knowing what you are doing can be disastrous not only for your network, but also for others. You should be rather save if you follow at least these rules:

- First of all, read this tutorial.

- Don't use local mappings. (see chapter 3.5)

- Make sure any domain you map to can also be mapped back.

- Aim for symmetry.

- Don't define a gateway table entry if the same domain already has a map2 entry. Such a rule would be redundant.

- Map to ADMD=0; if you will not be connected to any ADMD for the time being.

- Only map to ADMD= ; if you are indeed reachable though _any_ ADMD in your country.

- Mind the difference between PRMD=; and PRMD=@; and make sure which one you need.

- Don't define mappings for domains over which you have no naming authority.

- Before defining a mapping rule, make sure you have the permission from the naming authority of the domain you want to map to. Normally, this should be the same organisation as the mapping authority of the domain in the left hand side of the mapping rule. This principle is called 'administrative equivalence'.

- Avoid redundant mappings. E.g. is all domains under 'tlec.nl' are in your control, don't define:

```
first.tlec.nl#O$first.PRMD$tlec.ADMD$ade.C$nl#
last.tlec.nl#O$last.PRMD$tlec.ADMD$ade.C$nl#
always.tlec.nl#O$always.PRMD$tlec.ADMD$ade.C$nl#
```

  but rather have only one mapping rule:

```
tlec.nl#PRMD$tlec.ADMD$ade.C$nl#
```

- Before introducing a new mapped version of a domain, make sure the world can route to that mapped domain.

  E.g. If you are operating a PRMD: C=zz; ADMD=ade; PRMD=ergo; and you want to define the mapping rules:

```
map1: PRMD$ergo.ADMD$ade.C$zz#ergo.zz#
map2: ergo.zz#PRMD$ergo.ADMD$ade.C$zz#
```

  Make sure that ergo.zz is DNS routeable (has an A or an MX record) and will be routed to a gateway that will route the mails from the Internet to you over X.400.

  In the other direction, if you are operating the Internet domain cs.woodstock.edu, and you want to define a mapping for that domain:

```
map2: cs.woodstock.edu#O$cs.PRMD$woodstock.ADMD$ .C$us#
map1: O$cs.PRMD$woodstock.ADMD$ .C$us#cs.woodstock.edu#
```

  Make sure that C=us; ADMD= ; PRMD=woodstock; O=cs; is routeable in the X.400 world and will be routed to a gateway that will route the mails from X.400 to your RFC 822 domain over SMTP. Within the GO-MHS community, this would be done by registering a line in a so-called domain document, which will state to which mail relay this domain should be routed.

  Co-ordinate any such actions with your national or MHS' gateway manager. See chapter 3.4.

4. Conclusion

Mail gatewaying remains a complicated subject. If after reading this tutorial, you have the feeling you understand the basics, try solving some problems for an end-user. This will often convince you that you didn't understand a bit after all. Even after having worked with it for many years, you can still make amazing discoveries every other week. So at least this is a rewarding area to work in :-) That is, if you have a patient nature..... Don't give up!

Appendix A. References

[821] RFC 821; Jonathan B. Postel; SIMPLE MAIL TRANSFER PROTOCOL; University of Southern California; August 1982

[822] RFC 822; Crocker, D.; Standard of the Format of ARPA Internet Text Messages; University of Delaware, August 1982

[987] RFC 987; Steve Kille; Mapping between X.400 and RFC 822; UK Academic Community Report (MG.19), June 1986

[1280] RFC 1280; Jon Postel; IAB OFFICIAL PROTOCOL STANDARDS; USC/Information Sciences Institute, March 1992

[1310] RFC 1310; Lyman Chapin; The Internet Standards Process; BBN Communications Corporation, March 1992

[1327] RFC 1327; Steve Hardcastle-Kille; Mapping between X.400(1988) / ISO 10021 and RFC 822; University College London, May 1992

[1328] RFC 1328; Steve Hardcastle-Kille; X.400 1988 to 1984 downgrading; University College London, May 1992

[BP-88] Bernhard Plattner, Hannes Lubich; Electronic Mail Systems and Protocols Overview and Case Study; Proceedings of the IFIP WG 6.5 International working conference on message handling systems and distributed applications; Costa Mesa 1988; North-Holland, 1989

[JH-92] Jeroen Houttuin; @route:100%name@address, a practical guide to MHS configuration; Top-Level EC 1992 (not yet published)

[HTA-faq] Harald Tveit Alvestrand; Frequently asked questions on X.400. Regularly posted on USEnet in newsgroup comp.protocols.iso.x400

[MSG-93] Jeroen Houttuin, Klaus Hansen, Serge Aumont; RFC 1327 address mapping authorities. RARE WG-MSG working document

[pug] COSINE MHS Pocket User Guide. COSINE MHS Project Team 1992. Also available in several languages from the MHS Co-ordination server, nic.switch.ch:/public/user-guides . See chapter 3.4.

[RG-87] Rudiger Grimm, Steinar Haug; A minimum Profile for RFC 987; GMD, November 1987; RARE MHS Project Team; July 1990. Also available from nic.switch.ch:/procedures/min-rfc987-profile . See chapter 3.4.

[X.4xx(84)] CCITT Recommendations X.400 - X.430. Data Communication Networks: Message Handling Systems. CCITT Red Book, Vol. VIII - Fasc. VIII.7, Malaga-Torremolinos 1984

[X.4xx(88)] CCITT Recommendations X.400 - X.420. Data Communication Networks: Message Handling Systems. CCITT Blue Book, Vol. VIII - Fasc. VIII.7, Melbourne 1988

Appendix B. Index

Appendix C. Abbreviations

ADMD	Administration Management Domain
ASCII	American Standard Code for Information Exchange
ASN.1	Abstract Syntax Notation One
BCD	Binary-Coded Decimal
BITNET	Because It's Time NETwork
CCITT	Comite Consultatif International de Telegraphique et Telephonique
COSINE	Co-operation for OSI networking in Europe
DARPA	Defense Advanced Research Projects Agency
DFN	Deutsches Forschungsnetz
DL	Distribution List
DNS	Domain Name System
DoD	Department of Defense
EBCDIC	Extended BCD Interchange Code
IAB	Internet Activities Board
IEC	International Electrotechnical Commission
IESG	Internet Engineering Steering Group
IETF	Internet Engineering Task Force
IP	Internet Protocol.
IPM	Inter-Personal Message
IPMS	Inter-Personal Messaging Service
IPN	Inter-Personal Notification
ISO	International Organisation for Standardisation
ISODE	ISO Development Environment
JNT	Joint Network Team (UK)
JTC	Joint Technical Committee (ISO/IEC)
MHS	Message Handling System
MOTIS	Message-Oriented Text Interchange Systems
MTA	Message Transfer Agent
MTL	Message Transfer Layer
MTS	Message Transfer System
MX	Mail eXchanger
OSI	Open Systems Interconnection
OU(s)	Organizational Unit(s)
PP	Mail gatewaying software (not an abbreviation)
PRMD	Private Management Domain
RARE	Reseaux Associes pour la Recherche Europeenne
RFC	Request for comments
SMTP	simple mail transfer protocol
TCP	Transmission Control Protocol
UUCP	Unix to Unix CoPy

Appendix D. Author's address

Jeroen Houttuin
RARE Secretariat
Singel 466-468
NL-1017 AW Amsterdam, Europe

Tel. +31 20 6391131
Fax. +31 20 6393289
RFC 822: houttuin@rare.nl
X.400: C=nl;ADMD=400net;PRMD=surf;O=rare;S=houttuin