

# 8

## OPEN SYSTEMS MESSAGING: ELECTRONIC MAIL

---

“E-mail”—the ability to send and receive the electronic equivalent of written correspondence typically delivered through a postal agency—is the most popular and powerful distributed application in use today. It is used for personal correspondence; with the creation of mailing lists, it is used for electronic “conferencing” and it can be used to post large documents to mailing lists for review. In many cases, it is a convenient alternative to file transfer. For those individuals who have no other file-transfer mechanisms available—i.e., those who may have only terminal access to a network—this form of *messaging* is not only convenient, it is an essential means of acquiring electronic documentation.

The postal service is the obvious paradigm for electronic mail: one composes mail, places it in an envelope, addresses the envelope, and passes it to a postal handler—a carrier, a mailbox, or a postal worker stationed at a service window in a post office. The postal handler is one of many “handlers” in the postal delivery system who will attempt to deliver the mail to the addressee identified on the envelope. Both OSI and Internet mail follow this basic model, although the mechanisms, protocols, and message formats of the two mail systems are different.

What is “mailed” through a postal service is not limited to personal correspondence; in addition to letters, people mail bills, invoices, and other business “forms”; photographs; books and catalogs—in short, all sorts of “stuff.” Similarly, electronic mail has evolved from the basic transfer of textual mail messages to encompass a variety of electronically encoded messages, including facsimile, graphic images, office documents and forms, digitized voice, telex, and potentially even more. The OSI Message Handling System and the Internet’s Simple Mail Transfer

Protocol (SMTP) with recently defined extensions make electronic mail a powerful medium as well.

---

## OSI Message Handling System (X.400 MHS, MOTIS)

The OSI Message Handling System is defined in the CCITT X.400 series recommendations; the ISO standards reference MHS by the less familiar name *Message-Oriented Text-Interchange System* (MOTIS) (ISO/IEC 10021: 1990, in many parts). They describe essentially the same functional model: a distributed system that provides end users with the ability to send and receive electronic messages.<sup>1</sup>

The distributed system that comprises the OSI MHS has the following functional entities:

- An *end user*, identified by an originator/recipient name (O/R name). The end user employs the message handling service to compose, send, and receive messages.
- A *user agent* (UA), an entity that provides an end user with the ability to compose and send messages and also delivers messages to an end user. User agents typically offer some form of local message “management” as well—that is, an end user is customarily provided with the ability to store copies of messages sent/received locally (i.e., in folders, or directories, for subsequent retrieval) and to receive notifications that mail has arrived (the notorious `biff` y in UNIX).
- *Message transfer agents* (MTAs), which forward messages from the originator to the recipient UAs. Where circumstances prevent the immediate delivery of a message to a recipient, MTAs often provide temporary storage of messages and will repeatedly attempt to deliver a message for some predetermined period of time. (Thereafter, the message will be discarded, and the mail system will attempt to notify the originator of the failure.) The conceptual model for “store-and-forward” messaging is illustrated in Figure 8.1.
- MTAs combine to form the *message transfer system* (MTS), and the distributed system composed of the UAs and MTAs is the Message Handling System (MHS). (See Figure 8.2).

---

1. ISO/IEC 10021 and the 1988 CCITT X.400 Recommendations are nearly identical. The earlier model for message handling services X.400-1984 is now obsolete; however, as there remain many implementations of the 1984 version in use today, both are described here.

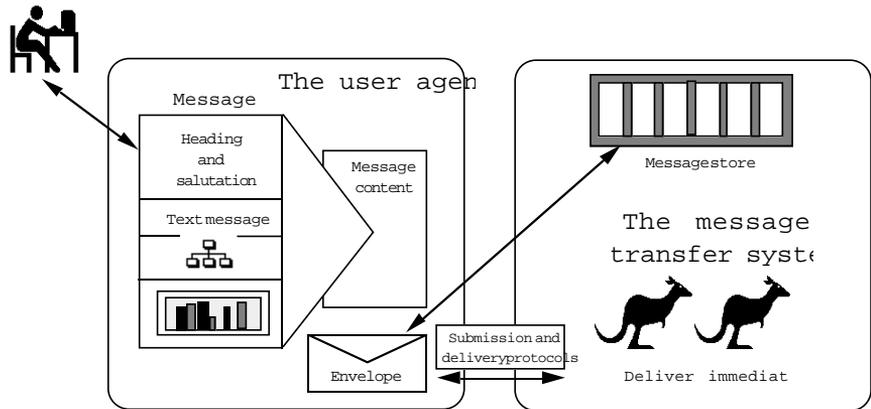
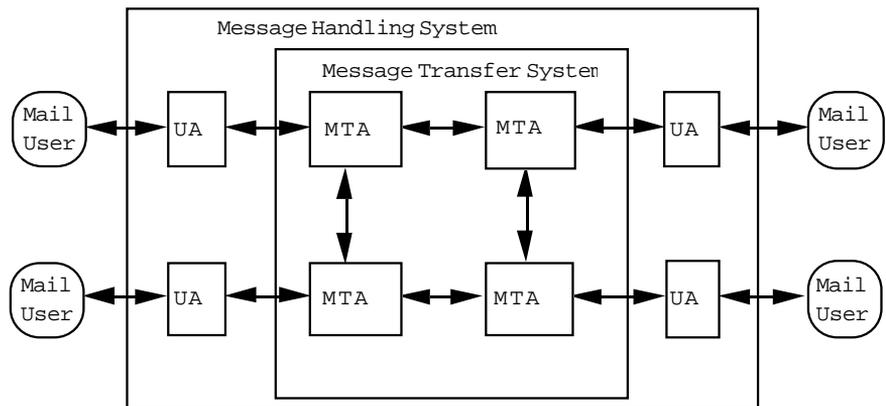


FIGURE 8.1 Store-and-Forward Messaging

In the early X.400 (1984) model of an MHS, the user agents operate at a sublayer above the MTAs, as follows. A mail *originator* (mail user X in Figure 8.3) calls upon a user agent (application) to compose a mail message to mail user Z. The UA application provides mail user X with prompts, menus, etc., that enable X to compose a message, in the process providing the UA with information essential for the preparation of an interpersonal message (IPM); i.e., X provides both heading information (to, from, subject, carbon copy, blind carbon copy) and body information (an ASN.1-encoded text message, accompanied perhaps by an ASN.1-encoded facsimile).



Source: Data Communication Networks Message Handling Systems: X.400 (1984)

FIGURE 8.2 MHS Model

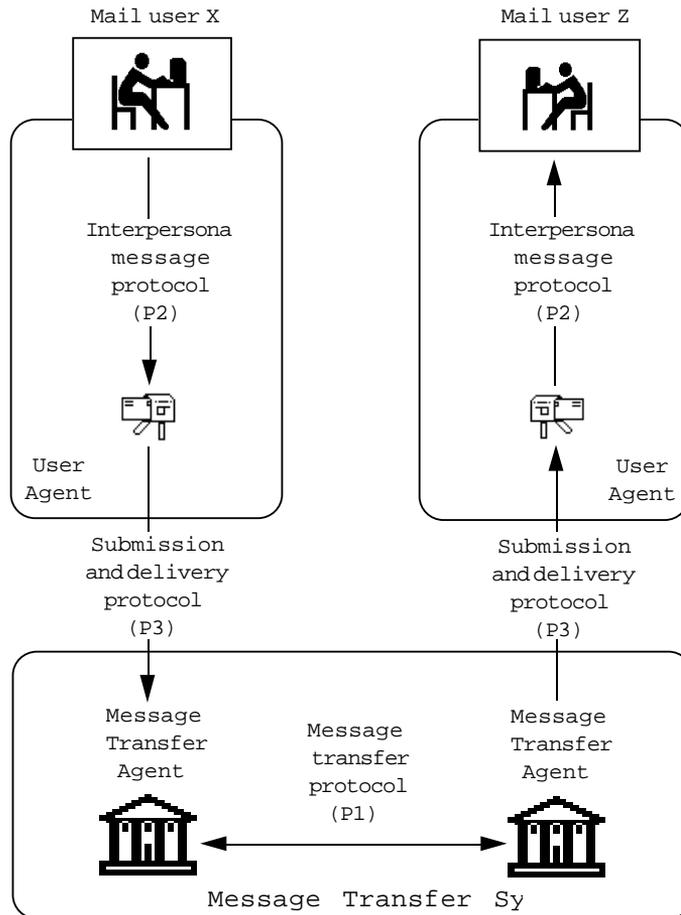


FIGURE 8.3 MHS Protocol Architecture (1984 version)

The message is communicated from the UA invoked by X to the UA that provides mail service to mail user Y through the use of the *interpersonal messaging protocol* (P2) (CCITT Recommendation X.420 [1984], [see ISO/IEC 10021: 1990]). The ASN.1-encoded header of this message (see Figure 8.4) is very nearly self-explanatory (comments in addition to those found in the X.420 recommendation have been added for clarification where needed). The body of an IPM may have multiple parts (see Figure 8.5), each distinguished by an indication of its type. Since the body types represent different electronic media, the OSI Message Handling System is often referred to as providing “multimedia” mail capabilities.

```

Heading ::= SET {
IPMessageId,           -- a unique and unambiguous message identifier, contains a
                        -- PrintableString and may include an O/R name
originator              [0] IMPLICIT ORDescriptor OPTIONAL,
                        -- the name of the message sender; e.g., a mail user X's O/R name
AuthorizingUsers       [1] IMPLICIT SEQUENCE OF ORDescriptor OPTIONAL,
                        -- only if not the originator
primaryRecipients      [2] IMPLICIT SEQUENCE OF Recipient OPTIONAL,
copyRecipients         [3] IMPLICIT SEQUENCE OF Recipient OPTIONAL,
                        -- a list of other folks who are to receive this IPM (more O/R
names)
blindCopyRecipients   [4] IMPLICIT SEQUENCE OF Recipient OPTIONAL,
                        -- a list of other folks who are to receive this IPM (more O/R
names)
                        -- but whose names should not appear in the heading delivered to
                        -- other recipients
inReplyTo              [5] IMPLICIT IPMessageId OPTIONAL,
                        -- the message identification of the message to which this IPM
refers
obsoletes              [6] IMPLICIT SEQUENCE OF IPMessageId OPTIONAL,
                        -- the message identifiers of any messages this IPM renders
                        -- obsolete
crossReferences        [7] IMPLICIT SEQUENCE OF IPMessageId OPTIONAL,
                        -- the message identifiers of any messages this IPM references
subject               [8] CHOICE {T61String} OPTIONAL,
expiryDate            [9] IMPLICIT time OPTIONAL,
                        -- represented as UTCTime, the date/timestamp beyond which the
                        -- delivery of this message becomes meaningless
replyBy               [10] IMPLICIT time OPTIONAL,
                        -- represented as UTCTime, the date/timestamp beyond which a
                        -- reply to this message is meaningless
replyToUsers          [11] IMPLICIT SEQUENCE OF ORDescriptor OPTIONAL,
                        -- a list of folks who should be included in any reply to this
message
importance            [12] IMPLICIT INTEGER { low(0), normal(1), high(2) }
                        DEFAULT normal,
sensitivity           [13] IMPLICIT INTEGER { personal(0), private(1),
companyConfidential(2) } OPTIONAL,
autoforwarded        [14] IMPLICIT BOOLEAN DEFAULT FALSE
                        -- an indication that this message has been forwarded automatical-
ly
                        -- by the MHS to the addressee "delegated" to receive this mail
}

```

(Source: Adapted from X.420 (1984), *Interpersonal Messaging User Agent Layer*)

FIGURE 8.4 ASN.1 Encoding of an Interpersonal Message

```

BodyPart ::= CHOICE {
    [0] IMPLICIT IA5Text,
        -- vanilla ASCII strings of characters
    [1] IMPLICIT TLX,
        -- 5-bit code assignments of ITA2 for conveying telex info
    [2] IMPLICIT Voice,
        -- bit string representing digitized voice
    [3] IMPLICIT G3Fax,
        -- a page count, followed by a sequence of bits, each representing a
        -- page of a group-3 facsimile
}

```

```

[4] IMPLICIT TIF0,
-- a document encoded according to T.73 text-interchange format
-- (used by group-4 fax class-1 terminals)
[5] IMPLICIT TTX,
-- body part is a teletex document (sequence of T61 charstrings)
[6] IMPLICIT Videotex,
-- body part is a videotex document (T.100- or T.101-encoded)
[7] NationallyDefined,
[8] IMPLICIT Encrypted,
-- a body part that has been subjected to encryption
[9] IMPLICIT ForwardedIPMessage,
-- a body part where an IPM has been subsumed within an IPM
[10] IMPLICIT SFD,
-- character-encoded information organized as a sequence of
-- paragraphs, the details of which are specified in CCITT
-- Recommendation X.420, called a "simple formatted document"
[11] IMPLICIT TIF1
-- a document encoded according to T.73 text-interchange format
-- (used by group-4 fax class-2 and -3 terminals)
}

```

(Source: Adapted from X.420 (1984), *Interpersonal Messaging User Agent Layer*)

FIGURE 8.5 IPM Body Types

Continuing with the example in Figure 8.3, the UA invoked by mail user X logs onto a message transfer agent and submits the interpersonal message through a series of service requests. Using the abstract LOGON service, the user agent provides the MTA with the name of the mail originator and a password to validate the UA. (In some electronic-mail applications, the end user provides log-on information once, at application start-up.) Once validated, the user agent invokes the message-submission service (SUBMIT) to transfer messages to one or more recipients. In practice, the detailed operation of the message-submission service is hidden from the end user. Although user interfaces differ across E-mail applications, many offer a menu of mail-creation operations (create new message, forward message, reply to message, attach file); once a message is composed, the user may queue it for submission or send it immediately, often by the simple act of hitting a "send" key or clicking on an equivalent "button." The E-mail application composes the information needed to submit the message to the message transfer system for the mail user (recipient, message content and content type, options such as deferred delivery time, priority, and delivery notice). If the MTA is remote from the UA—i.e., the UA cannot communicate with the MTA via a local (Interprocess communication) interface—the interpersonal message is forwarded to the MTA in a `submissionEnvelope`, one of a set of message protocol data units

2. At the time of publication of the X.400 (1984) Recommendations, the application-

(MPDUs) of the *submission and delivery protocol* (P3). The P3 protocol uses a remote operations service and optionally a reliable transfer service to submit messages (interpersonal as well as operations messages) to the MTA (CCITT Recommendation X.410-1984)<sup>2</sup>; note that embedding this functionality in the MHS elements was quite controversial and is inconsistent with the current structure of the OSI upper layers (see Chapter 10).

The submission and delivery protocol provides the MTA with the addressing and message-processing information it needs to forward (or store and later forward) the IPM through the MTS. Thus, either by decomposing the protocol or by directly parsing the parameters of the SUBMIT.request, an MTA will have acquired the addressing and processing information it needs to forward (or store) the IPM. The MTA composes a user MPDU, consisting of an *envelope* and *content*, to carry the interpersonal message toward its destination.<sup>3</sup> (See Figure 8.6.)

```
UserMPDU ::= SEQUENCE { UMPDUEnvelope, UMPDUContent }
UMPDUEnvelope ::= SET {
    MPDUIdentifier,
    -- a global domain identifier-country name, administrative
    -- domain, and optionally, a private domain identifier, plus a
    -- printable ASCII string-uniquely identifies this UMPDU
    originator ORName,
    originalEncodedInformationTypes OPTIONAL,
    ContentType,
    -- the class of UA used to create the content-e.g., a value of
    --IPM or P2
    UAContentID OPTIONAL,
    -- a printable string
    Priority DEFAULT normal,
    -- can be nonUrgent, normal, or urgent
    PerMessageFlag DEFAULT { } ,
    -- handling directives: disclose recipients, conversion prohib-
ited,
    -- alternate recipients allowed, content return request
    deferredDelivery [0] IMPLICIT Time OPTIONAL,
    -- UTCTime
    [1] IMPLICIT SEQUENCE OF PerDomainBilateralInfo
    OPTIONAL, [2] IMPLICIT SEQUENCE OF RecipientInfo,
    TraceInformation }
```

layer structure described in Chapter 10 was not complete. Remote operations and reliable transfer mechanisms were incorporated into the MHS model to support the submission and delivery protocol and message transfer protocol. The term *server* was used to describe these remote operations and reliable transfer mechanisms. The structure of the application layer for the MHS was revised and aligned with the OSI application-layer structure following the publication of the CCITT Red Books in 1985; the “servers” were removed, and the MHS model now makes use of the remote operations and reliable transfer service elements.

3. This discussion of X.400-1984 focuses primarily on the submission and delivery of messages. The X.400-1984 Recommendations describe the operational behavior of UAs and MTAs and, in particular, describe mechanisms and protocols that MHS elements may use to perform status inquiries, detect routing loops, etc. Discussion of these aspects is beyond what the authors hope to cover here.

UMPDUContent ::= OCTETSTRING

(Source: Adapted from X.411 (1984), Message Transfer Layer)

FIGURE 8.6 ASN.1 Encoding of the UserMPDU of the P1 Protocol

Conceptually, an IPM with a multipart body, encapsulated in the P1 protocol, looks like Figure 8.7.

When the user MPDU of the P1 protocol arrives at the message transfer agent that provides delivery service to mail user Z's user agent, notification and delivery of the interpersonal message are accomplished

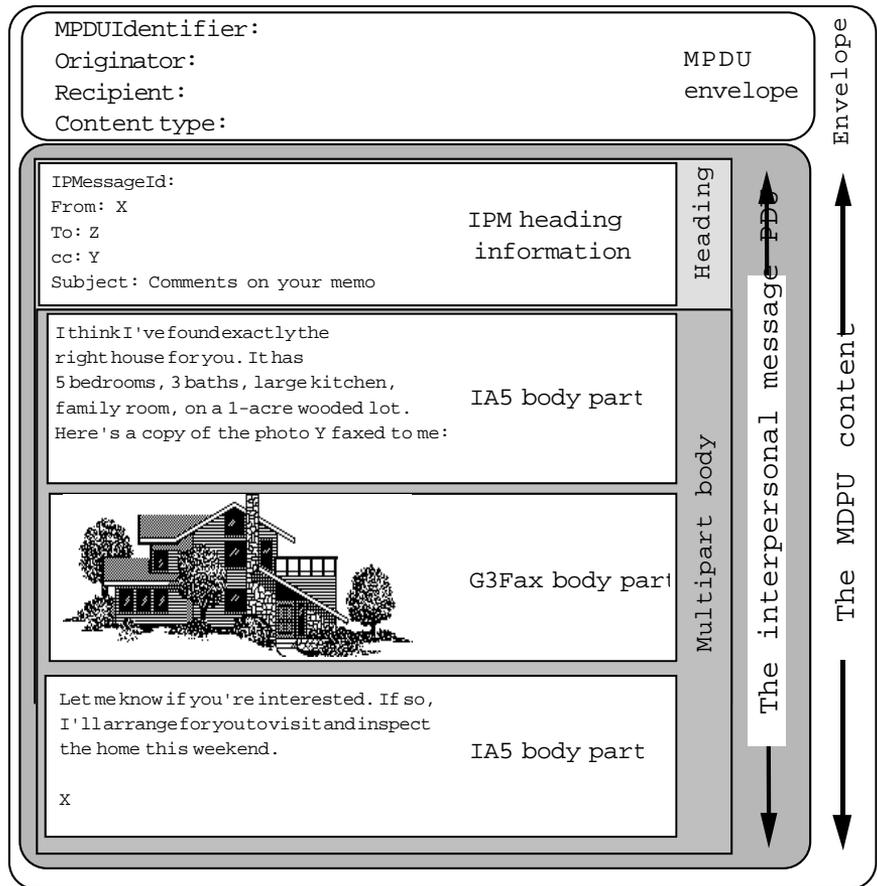


FIGURE 8.7 Conceptual Message Structure

---

## Organization Mapping— Administration of Message Handling Systems

through the use of the message-delivery service (DELIVER). If the MTA is remote from the recipient's user agent, the `deliverEnvelope` MPDU of the submission and delivery protocol (P3) is used to deliver the message from the MTA to mail user Z's user agent. Z's user agent extracts the interpersonal message from the envelope and makes it available to Z. As with message submission, the details of this operation are hidden from the mail user. (An audible or visible notice of mail arrival is common to e-mail applications. The mail user then "opens" his or her mailbox to view the new mail that has arrived.)

Like telephony, electronic mail is most powerful if it crosses organizational, national, and international boundaries. And like the global telephone network, an infrastructure must be associated with this power to see that it is operated effectively and responsibly. OSI MHS describes a hierarchy that enables public and private administrations to cooperate in providing message-handling services.

A collection of message transfer and user agents is said to constitute a *management domain* (MD). MDs may provide both message-transfer and interpersonal message services. Publicly administered MDs—i.e., those operated by a PTT or regulated telecommunications carrier—are called *administration management domains* (ADMDs, or AMDs); MDs operated by a company or noncommercial organization are called *private management domains* (PRMDs or PMDs). Both may offer user agent services to their subscribers. For reasons both political and economic, a PRMD is considered to operate wholly within a single country; hence, a multinational company will have multiple PRMDs. According to the letter of the X.400-1984 Recommendations, private management domains may connect to multiple administrative management domains (e.g., a PRMD in the United States may connect to message handling services offered by several local, independent, and interexchange companies, if they were all indeed permitted to offer such *information services*). PRMDs are not allowed to forward mail between ADMDs (e.g., act as a mail gateway between countries), and of course, a private management domain in the United States, for example, must forward messages to a PRMD in the United Kingdom through ADMDs. The relationship between administrative and private management domains, and correct methods of interconnection are shown in Figure 8.8.

---

## Names and Addresses in MHS



Although such heretical notions are not codified in CCITT Recommendations, PRMDs within a country may connect their MTAs without an intervening ADMD in that country. In the United

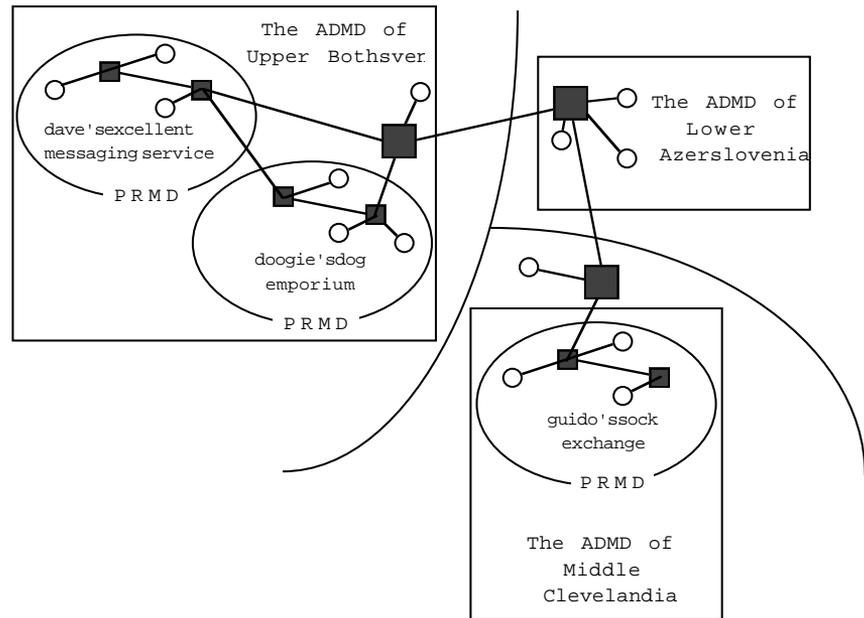


FIGURE 8.8 Administration and Private Management Domains

*States, we call this dreaded notion "bypass," or simply "enterprise mail."*

Every subscriber to the MHS is potentially an originator and recipient of messages; hence, the term *originator/recipient name*, or *O/R name*, is used to describe an MHS user. An O/R name is supposed to be *descriptive*; i.e., it uniquely and unambiguously identifies an MHS user. O/R names are composed of attributes that provide sufficient specific information to distinguish every mail user from every other mail user. For example, suppose there is a Dr. Aorta who is a hematologist on staff in the Temple University Hospital, Philadelphia, Pennsylvania, U.S.A. (see Figure 7.5 in Chapter 7). This attribute list uniquely distinguishes Dr. Aorta from Dr. Ventricle, who is also a hematologist on staff in the hospital of Temple University, Philadelphia, Pennsylvania, U.S.A. However, if Dr. Bert Aorta and Dr. Ernie Aorta are both on staff in the same hematology department, then the attribute list is not sufficient to distinguish Bert from Ernie; a given-name attribute is needed.

The same logic is supposed to be applied to O/R names, and the organizational mapping of the OSI Message Handling System provides insight into the construction of O/R names. Management domains are responsible for ensuring uniqueness of O/R names within the MD. The

standard attributes of an O/R name are:

- *Personal*: nominally a personal name, perhaps composed of surname and given name, initial, and generational qualifier.
- *Organizational*: the name and unit of the organization (company or noncommercial enterprise).
- *Architectural*: ADMD or PRMD name, an X.121 *public data network number*, or a unique UA identifier.
- *Geographic*: nominally a country name; may also include street name and number, town, and region.

For compatibility with telematic services—in the 1984 version of the OSI MHS, O/R names are sometimes more address than name. From the attribute lists, MDs may create O/R names of several forms. These are illustrated in Table 8.1 (consistent with X.400, the optional attributes are distinguished by the use of square brackets).

An example of one of the more commonly encountered forms of O/R name is C=US/ADMD = ATTMAIL/PRMD = DNA6L/ORG = UNISYS/PN = JudyGertz.

O/R names identify users; to forward messages, however, user agents must provide the message transfer system with the address of the destination UA so that the MTS can select the route the message must take to arrive at the destination UA. In X.400 (1984), O/R names describe elements of the MHS architecture, and some variants go so far as to embed network addressing information in the attribute list. In a kinder, gentler world, O/R names would not have such routing information; names would be independent from addressing entirely, and the bindings

TABLE 8.1 Forms and Variants of O/R Names

<i>Form 1, Variant 1</i>	<i>Form 1, Variant 2</i>	<i>Form 1, Variant 3</i>	<i>Form 2</i>
Country name	Country name	Country name	X.121 address
ADMD name	ADMD name	ADMD name	Teletex Terminal identifier
[PRMD]	UA unique identifier	X.121 address	
[Organization name]			
[Organization unit]			
[Personal name]	[Domain-defined]	[Domain-defined]	
[Domain-defined]			

---

**Refinements to  
the MHS—X.400  
(1988)**

between O/R names and O/R “addresses” used for routing would be acquired from a directory service. Fortunately, UAs of mail-processing systems typically offer users a means of creating personal lists of aliases or abbreviated names, so rather than having to remember and type “C = US,” “ADMD = ATTMAIL,” “PRMD = DNA6L,” “ORG = UNISYS,” “PN = JudyGertz,” it is quite possible that you’ll be able to type or select “Judy Gertz.”

The 1984 version of the MHS predated the completion of the OSI upper- and application-layer structures described in Chapters 6 and 10, respectively. Some of the facilities identified in entirely separate application service elements—notably, remote operations, reliable transfer, and association control—were embedded in the message handling service, along with the specification of presentation transfer syntax (X.409-1984 is actually a precursor to ASN.1/BER). The X.400-1984 MHS model also suggested an altogether unnecessary layering relationship between the UA and MTA service elements.

Deployment of X.400-1984 revealed some serious limitations. It did not readily accommodate mailing lists, and no security features had been defined. The naming structure was poorly defined, as was the “store” component of store and forward. None of these limitations seemed insurmountable, and certainly, remedies could be found in four years. Remarkable what field experience reveals . . .

The 1988 version of the MHS describes a restructured MHS model, more closely in line with the OSI upper- and application-layer structures. Specifically, open systems behaving as user agents, message store, message transfer agents, and access units—the *functional objects* in the Message Handling System—are described as application processes (APs). Central to each AP is the application entity, which consists of a set of MHS-specific application service elements that perform message administration (the MASE), submission (the MSSE), delivery (the MDSE), retrieval (the MRSE), and transfer services (the MTSE). These ASEs use the supporting “core” ASEs described in Chapter 10—remote operations, reliable transfer, and association control. Conversion to/from abstract/transfer syntax has migrated to the presentation layer.

The user agent and message transfer agent are part of a set of what are called *consumers* and *suppliers* of message handling services. Consider Figure 8.9. A user agent application entity that consumes services that perform message submission, delivery, retrieval, and administration services consists of a user element (UE) plus the four ASEs that provide these services. These services are supplied by the MTA along with an

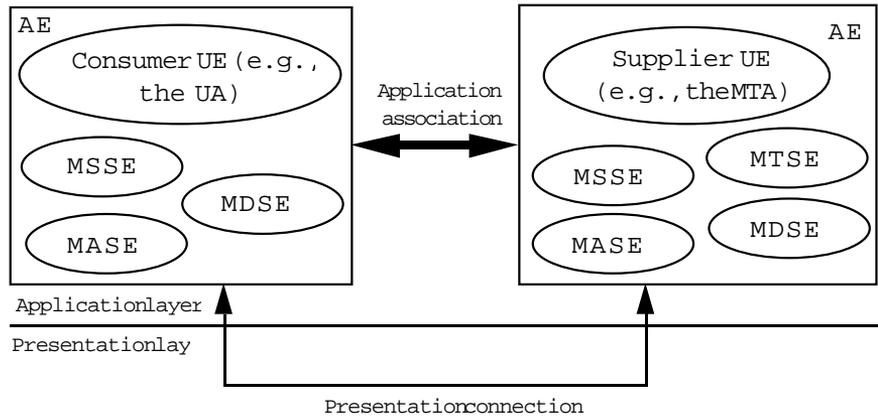


FIGURE 8.9 MHS Application Entity Structure (1988)

explicit message archive called the *Message Store*.

The relationship between the functional objects and the consumer/supplier ASEs is illustrated in Table 8.2.

Some of these changes are reflected in the MHS protocols (the revised protocol architecture is depicted in Figure 8.10). A message store access protocol (P7) has been introduced to enable the UA to contact the message store directly. The submission and delivery (P3) protocol, used by both the message store and user agents to access the message transfer service, now has *extension fields* to accommodate the identification of mail users through directory names, the use of object identifiers to identify content type, and the ability to specify external encoded information types. The structure of the IPM remains the same, and the P2 message content remains OCTETSTRING, but some of the body parts identified in 1984 are eliminated (telex and simple formatable document, found to be

TABLE 8.2 Relationship between the MHS Functional Objects and the Consumer/Supplier ASEs

ASE	Functional Objects of the MHS			
	UA	MS	MTA	AU
MTSE	—	—	Consumer/supplier	—
MSSE	Consumer	Consumer/supplier	Supplier	—
MSDE	Consumer	Consumer	Supplier	—
MRSE	Consumer	Supplier	—	—
MASE	Consumer	Consumer/supplier	Supplier	—

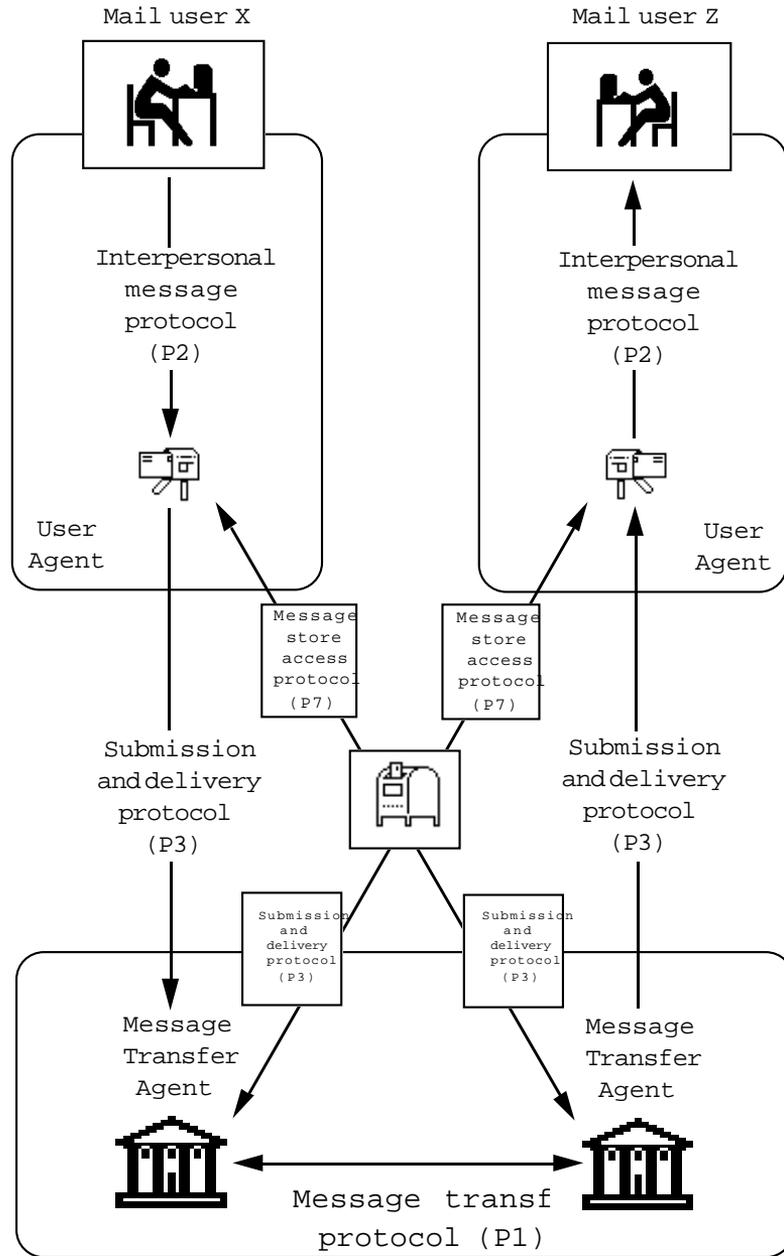


FIGURE 8.10 MHS Protocol Architecture (1988)

redundant). A new value for the content type (22) is defined to distinguish X.420-1988–encoded contents from ones that are entirely consistent with content types that might have been generated using X.420-1984. (In such cases, the value of this data type remains 2.)

X.400-1988 introduces new functionality as well. It is now possible to create distribution lists and, within this context, closed user groups. Distribution lists provide mail users with the ability to send messages to a group of mail users using a single O/R name (which is the name of the list). A mail user who is authorized to send mail to a distribution list sends a single message to the distribution-list O/R name; the message transfer system sees that a copy of this message is forwarded to all members of the list (this process is called *expansion*). Typically, a single user manages the distribution list and is responsible for adding members to and removing members from the list. The distribution list identifies the recipients of messages but imposes no restrictions on who may post messages to the distribution-list O/R name, once that name is discovered by a nonmember. An administrator of a distribution list may also limit who is allowed to send messages to that list (the constituency of who may post messages to a distribution list is called a *closed user group*).

X.400-1988 provides a means of introducing privacy to electronic mail. Using the security mechanisms developed for the X.500 Directory, a public-key encryption system can be used to generate an electronic signature. The means of decoding the signature are understood only by the communicating parties and the public-key administrator. With the signature, a recipient can, for example, authenticate the origin of a message, verify the integrity of the message content, and authenticate the message partner (peer). Using mechanisms also recommended in X.509, a message can be encrypted and thus kept confidential. (A thorough discussion of the security aspects of X.400-1988 is provided in Plattner et al. 1991.)

---

## MHS and the Directory

X.400-1988 introduces the use of X.500 directory names as a complementary way of identifying mail users (O/R names as composed in X.400-1984 can still be used). Directory distinguished names, which can be entirely free of “addressing” attributes, can be used by the UA to access the MS or MTA, and between the MS and an MTA as well; for routing between MTAs, however, a directory name-to-UA address mapping must be performed. Specifically, either the user agent of the originator of a message or the first message transfer agent on the path from the originator to the recipient must perform the directory name to O/R address mapping by querying the OSI Directory whenever the originator uses a directory name rather than an O/R address. Such mappings can be regis-

tered in the OSI Directory by the administrator of a management domain; thereafter, MTAs may search the OSI Directory for the name-address mapping. The name-to-address mappings may also be modified by an administrator if, for example, a mail user moves from one management domain to another. If directory naming is done independently of UA addressing, the migration of the mail user from one management domain to another is entirely transparent to all other mail users.

Message Handling System use of the OSI Directory is not limited to name-address resolution. MHS elements may use the OSI Directory to expand distribution lists (i.e., to obtain the UA addresses of all the UAs that provide service to members of the list) and to learn what services and functions MHS components support.

---

**MHS Use of Remote Operations and Reliable Transfer Facilities**

In configurations in which an MTA is remote from a UA, the submission and delivery protocol (P3) provides the means by which an MTA and the user agent accomplish what would otherwise be signaled across a local interface, perhaps in something as simple as a procedure call; i.e., the UA invokes operations at the MTA (literally “store and forward the message”), and similarly, the MTA delivers messages to the UA (deliver it to the user). In OSI, procedure calls that are performed across a distributed interface (i.e., across an OSI connection between two applications) are called *remote operations*.

Interpersonal messages containing graphic, digitized voice, or facsimile body parts can be quite large. Generally speaking, if communication between UAs and MTAs is disrupted during the transfer of a large message, restarting from the beginning of the message is time-consuming and, in some cases, expensive. For such messages, the ability to provide checkpoints during message transfer so that UAs and MTAs can recover from (temporary) communications failures, resynchronize to a common point to restart, and continue from that point with a minimum amount of retransmission is an important aspect of message forwarding and delivery. In OSI this is called *reliable transfer*. OSI provides both reliable transfer and remote operations capabilities as part of the communications toolbox described in Chapter 10, ‘Core’ Application Service Elements. In the 1984 version of the MHS, remote operations and reliable transfer were embedded in X.400 (1988), the MHS application service elements make use of the remote operations and reliable transfer services provided by the core ASEs to operate the MHS protocols.

---

**Interworking between X.400 (1984) and X.400 (1988)**

By the time the 1988 version of MHS was available, a modest but grow-

---

4. The mechanisms to assist in negotiating the correct OSI upper layer protocol environment are described in Chapter 10.

ing installed base of 1984 implementations existed. Eventually, these will go away. However, with the extensions to the MHS and protocols, with the introduction of new naming conventions, and in particular, with the considerable modifications to the OSI upper layers over which MHS would operate, a fair number of incompatibilities exist.<sup>4</sup>

From the perspective of the 1988 MHS user, interconnecting 1988 and 1984 MHS implementations is a process, however temporary, of lowering one's expectations and is called *downgrading*. The extensions introduced in the 1988 version were done so that the MHS would run without them (they are encoded as additional elements of protocol). X.419-1988 Appendix B notes that unless the extension is marked as critical for transfer or critical for delivery, it can simply be deleted; otherwise, "downgrading" cannot be performed, and message forwarding/delivery will fail. (A particularly disappointing aspect of downgrading is that it prevents the use of the security features.) Directory names cannot be used, and X.419-1988 suggests that downgrading be accomplished by deleting the directory name and the O/R address. The Internet community has spent considerable time and effort piloting the use of X.400 and has some practical solutions to dealing with "downgrading." As a general approach, RFC 1328 suggests the use of a domain-defined attribute, always a standard O/R name as defined in RFC 1327.

RFC 1328 also suggests several alternatives that may be applied when dealing with the issue of downgrading interpersonal messages. Depending on gateways and their configuration, it will in some cases be necessary to downgrade from an X.400-1988 content type to one conforming to X.400-1984. In such circumstances, only protocol control information and addressing that can be parsed by a 1984 MHS implementation can remain in the IPM that is to be forwarded. Body-part conversion is another story. Five scenarios exist:

- Some of the information encoded in the 1988 body part is lost.
- The 1988 body part is converted without loss (not always in the "a miracle occurs" category, but close).
- Conversion is simply not possible, and the message must be discarded.
- The body part can be discarded and replaced with a (typically IA5 text) message.
- The body part can be encapsulated as an external 1984 body part.

Although these scenarios paint a somewhat bleak picture when downgrading, the net effect is more positive. Mail users on 1984 MHS may still exchange primarily textual messages with 1988 MHS mail users. The bottom line on downgrading: avoid it if possible.

## Internet Mail

Although there are many mail systems in operation across the TCP/IP Internet, the most popular is based on the Simple Mail Transfer Protocol (SMTP, RFC 821), a protocol used to reliably transfer mail, and the standard for Internet text messages (RFC 822), which specifies a syntax for text messages. SMTP uses an interprocess communication paradigm for mail submission, relay, and delivery; since mail transfer is expected to be reliable, SMTP operates over a TCP connection (see Chapter 12) between hosts. Hosts in the mail system provide mailboxes for mail users, behaving in this role like MHS user agents.<sup>5</sup> Hosts attached to more than one network relay mail messages between hosts that are not connected to the same network; their behavior in this role can be compared to MHS mes-

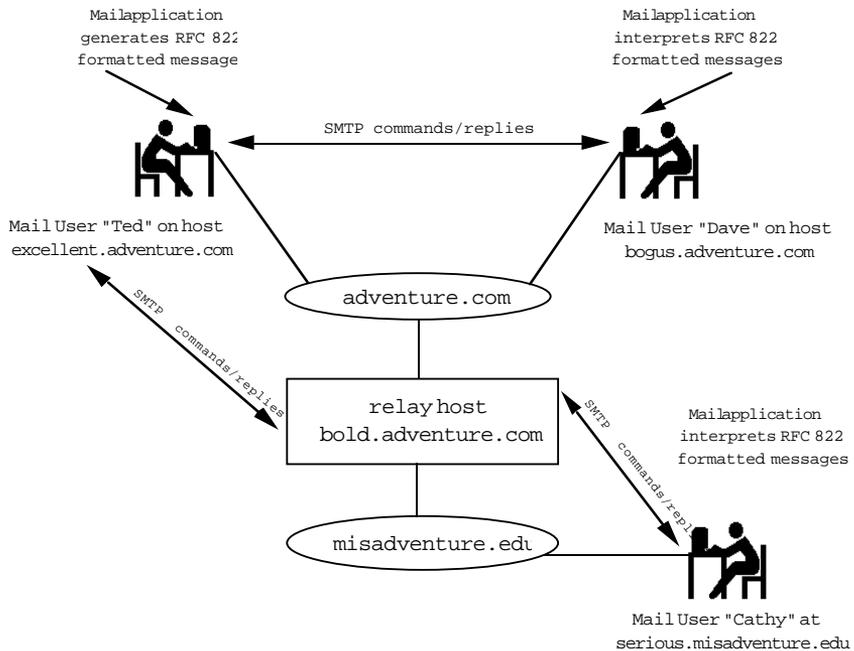


FIGURE 8.11 SMTP Model for Mail Delivery

5. The analogy is not intended to suggest that Internet based its architecture on MHS; it is merely an artifact of having elected to describe MHS first. It is probably safer to assume that while developing the X.400 MHS recommendations, CCITT and ISO experts may have happened across the Internet mail standards among the many they researched, and some are likely to have used Internet mail during the MHS development period.

sage transfer agents. The collection of hosts involved in these activities forms an SMTP-based mail-delivery system, as shown in Figure 8.11.

RFC 822 defines the syntax and composition of mail messages. As with MHS, mail consists of an envelope containing information necessary to forward and deliver mail, and a message content. Mail consists of a header and a single body part consisting of lines of text; standard “822” mail doesn’t address how one might encode graphics, facsimile, voice, etc. Through years of experience, folks in the Internet have applied “ad hack” means of transferring documents more complex than mere 7-bit ASCII text; common ways to transfer binary information include programs that expand each digit of a binary octet into two 7-bit ASCII bytes or encode 3 octets as 4 plus control information (e.g., UNIX *uuencode*) RFC 1113. More recent extensions to SMTP provide a wider range of possibilities and are discussed later in the chapter.

The RFC 822 mail header consists of single lines of ASCII text called header fields.<sup>6</sup> Each field contains a field name and a field body. A field name is separated from the field body by a colon (“:”) and may not contain any SPACE or control (CTL) characters including the colon. The field body is terminated by a carriage return character followed by a line feed character (CRLF). RFC 822 does not require that the fields in the mail header appear in a particular order, except that all header fields must precede the body of the mail message.<sup>7</sup> The general form of mail in the metasyntax used by RFC 822 is shown in Figure 8.12. Figure 8.13 provides an “annotated” example:

```

field      = field-name ":" [ field-body ] CRLF
field-name = 1*<any CHAR, excluding CTLs, SPACE, and ":">
field-body = field-body-contents
            [CRLF LWSP-char field-body]

field-body-contents =
<the ASCII characters making up the field-body, as
defined in the following sections, and consisting
of combinations of atom, quoted-string, and
```

6. The notion of a “single” line is slightly deceiving, since it is more accurately interpreted as a set of ASCII characters terminated by a carriage return/line feed (CRLF) combination.

7. RFC 822 does recommend that, if present, header fields should be sent in the following order: “Return-Path,” “Received,” “Date,” “From,” “Subject,” “Sender,” “To,” “cc.”

special tokens, or else consisting of text>

(Source: RFC 822 (1982), *Standard for the Format of ARPA Internet Text Messages*)

FIGURE 8.12 Metasyntax of “822” Mail

```

Replied: Wed, 20 Jan 93 16:16:04 -0500           % date this mail reply was sent
Replied: "Christine Hemrick <hemrick@cisco.com>" % identifies who replied to mail
  Return-Path: hemrick@cisco.com                % information about the address and
                                                % route back to mail originator,
                                                % provided by final transport system
Received: by mail.bellcore.com;id 9301202023.AA05391 % a copy of this field is added by
each
                                                % transport service that relays the
                                                % message—used for trace
Received: from ash.cisco.com by breeze.bellcore.com (5.61/1.34)
  id AA06912; Wed, 20 Jan 93 15:20:49 -0500
Message-Id: <9201202020.AA06912@breeze.bellcore.com>. % unique message identifier
Received: by ash.cisco.com; Wed, 20 Jan 93 12:20:44 -0800
From: Christine Hemrick <hemrick@cisco.com>      % sender of message
Subject: RE: meeting time and venue              % subject
To: dave@bellcore.com (Dave Piscitello)          % intended recipient
Date: Wed, 20 Jan 93 12:20:43 PST                % time message was received by
                                                % transport system serving recipient
In-Reply-To: <9201201625.AA17281@sword.bellcore.com>; % identification of message
to which
  from "Dave Piscitello" at Jan 20, 93 11:25 am   % this message replies
X-Mailer: ELM [version 2.2 PL16 mips 1]          % user-defined field name

```

FIGURE 8.13 Sample “822” Mail Header

Mail is composed according to this syntax and submitted by an end-user application to a mail facility—e.g., UNIX *sendmail*—for forwarding; in this example, the *sendmail* facility uses “ARPANET” mail format and SMTP commands.

SMTP commands and replies are also encoded as 7-bit ASCII characters. The core aspects of sending and receiving mail are straightforward. Suppose that mail user Ted at host “*excellent.adventure.com*” wishes to send mail to user Dave at host “*bogus.adventure.com*” and that Ted’s computer is able to establish a TCP connection to Dave’s computer. Ted uses a mail application to create and send an 822 mail message. The mail application at *excellent.adventure.com* initiates a mail transaction with host *bogus.adventure.com* by invoking a local sender SMTP process, which establishes a TCP connection to a receiver SMTP at host *bogus.adventure.com* at service port 25. To indicate that the TCP connection was successful, the receiver SMTP at *bogus*.

---

8. The three-digit reply code is used by SMTP; any text that follows is meant to assist postmasters . . . or humor them.

adventure.com returns the SMTP reply "220 bogus.adventure.com Service Ready." The sender SMTP process at excellent.adventure.com next sends the command "HELO excellent.adventure.com"; the receiver SMTP accepts the mail connection by returning the reply "250 bogus.adventure.com, Hello excellent pleased to meet you" to the sender SMTP.<sup>8</sup> At this point, the sender and receiver SMTP processes have completed greetings, and mail transactions may proceed. The sender SMTP at excellent.adventure.com issues the command "MAIL FROM: <Ted@excellent.adventure.com>." The receiver SMTP at bogus.adventure.com acknowledges the identification of the sender by replying "250 <Ted@excellent.adventure.com>...Sender OK." The sender SMTP then submits the recipient information for the mail in the command "RCPT TO: <Dave@bogus.adventure.com>." The receiver SMTP at bogus.adventure.com indicates that it knows about the mailbox "Dave," so it replies "250 <Dave@bogus.adventure.com>... Recipient OK." The sender SMTP now sends the mail message "DATA," indicating that it wishes to forward a mail message. The receiver SMTP replies "354 Start mail input; end with <CRLF>. <CRLF>" and will treat the text lines that are transferred as mail data until it receives a mail data termination sequence; i.e., a CRLF, followed by an ASCII period character ("."), followed by a CRLF. If the message transfer is successful, the receiver SMTP at bogus.adventure.com replies "250 OK" and attempts to notify Dave that mail has arrived. (If this were the only message that the mail system at excellent.adventure.com had to send to bogus.adventure.com, it would then issue a "QUIT" command, and the receiver SMTP would close the mail service connection [reply code 221]; otherwise, the mail sequence is repeated.)

Note that if there were multiple recipients for this mail, the sender SMTP would issue one "RCPT TO: <forward-path>" command for each recipient, where the <forward-path> argument indicates a single mail recipient's address (a source route to a mail recipient may accompany the address as part of the argument).

This example illustrates only the scenario in which the host systems involved are able to establish direct connectivity—e.g., within a single domain (adventure.com). In configurations in which mail-delivery hosts cannot directly connect using TCP (e.g., for policy/administrative reasons, an enterprise network may not allow all mail systems within its domain to exchange mail directly), mail must be forwarded through multiple mail-delivery systems. (This is also true, and even more complicated, when mail application "gateways" are used to send and deliver

mail between mail users operating over different mail systems—e.g., MHS and Internet mail, discussed later in this chapter.) Suppose, for example, that Ted tries to send mail to Cathy’s mailbox at `serious.misadventure.edu`, and `excellent.adventure.com` must relay mail through `bogus.adventure.com` to do so. The sequence of mail commands and replies might be as shown in Figure 8.14.

```
SMTP Process      Command/Reply
{excellent.adventure.com is the source host, and bold.adventure.com the relay host, see
Figure 8.11}
sender            (opens TCP connection to host bold.adventure.com)
receiver         220 bold.adventure.com Service Ready
sender           HELO excellent.adventure.com
receiver        250 bogus.adventure.com, Hello excellent pleased to meet you
sender           MAIL FROM: <Ted@excellent.adventure.com>
receiver        250 <Ted@excellent.adventure.com>...Sender OK
sender           RCPT TO: <@bold.adventure.com: Cathy@serious.misadventure.edu>
receiver        250 <@bold.adventure.com:
Cathy@serious.misadventure.edu... Recipient OK
sender           DATA
receiver        354 Start mail input; end with <CRLF>.<CRLF>
sender           Date: 14 MAY 1993 10:10:11
sender           From: Ted@excellent.adventure.com
sender           Subject: reschedule conference time & venue
sender           To: Cathy@serious.misadventure.edu
sender           I have a conflict; can we reschedule to Tuesday at 9 am?
sender           .
receiver        250 OK
sender           QUIT
receiver        221 excellent.adventure.com Service closing transmission channel
{relay host bold.adventure.com now becomes the sender, and destination host serious.misad-
venture.
  edu, the receiver}

sender           (opens TCP connection to host serious.misadventure.edu)
receiver        220 serious.misadventure.edu Service Ready
sender           HELO bold.adventure.com
receiver        250 serious.misadventure.edu
sender           MAIL FROM: <@bold.adventure.com: Ted@excellent.adventure.com>
receiver        250 OK
sender           RCPT TO: <Cathy@serious.misadventure.edu>
receiver        250 OK
sender           DATA
receiver        354 Start mail input; end with <CRLF>.<CRLF>
sender           Received: from excellent.adventure.com by
bold@adventure.com; 14 MAY 1993 10:11:45
sender           Date: 14 MAY 1993 10:10:11
sender           From: Ted@excellent.adventure.com
sender           Subject: reschedule conference time & venue
sender           To: Cathy@serious.misadventure.edu
sender           I have a conflict; can we reschedule to Tuesday at 9 am?
sender
```

```

sender      .
receiver    250 OK
sender      QUIT
receiver    221 serious.misadventure.edu Service closing transmission channel

```

FIGURE 8.14 Relayed Mail Scenario

It is important to note that, while sending, mail-delivery systems keep a copy of the mail they forward until they have successfully transferred it to the destination system (the host of the recipient); in the case of relaying, a sender only keeps a copy until it has successfully transferred the mail to the relay (and the process iterates as each relay acts as a sender). The positive aspects of using a reliable transport service like TCP to transfer mail in these circumstances is sometimes negated: mail can be misrouted or lost by mail systems, and it is often difficult if not impossible to provide a failure notification to the originator.

**Mail Addresses** In its simplest form, a mail address—or mailbox—is by convention of the form {"local part," "@," "domain"}. The local part may be as simple as the name of a user; e.g., Ted, Dave, or Cathy in the earlier examples. The local part may be decidedly complex, especially if used at gateways to convey mail addresses of mail systems other than SMTP/822 systems (e.g., UUCP or proprietary mail systems). "Domain" always names a host in the mail system (see Chapter 7); it is typically constructed as a sequence of {"element<sub>1</sub>," "." "element<sub>2</sub>," "." . . . "element<sub>n</sub>"} . The *n*th elements (e.g., "com" and "edu") are "top-level" name domains that share a common root—the Internet naming domain—and elements *n* – 1 to 1 are children of the root; for example, "serious" is a host in the "misadventure" name domain, which itself is in "edu."

Most mail applications allow users to create nicknames or aliases that are easier to remember—i.e., "user-friendly." The UNIX *mail* processing system permits users to create single-name aliases and personal distribution lists using an alias command line; e.g., the entry "alias Ted Ted@excellent.adventure.com" will allow a user to type "Ted" rather than the full mail address.

**Distribution Lists** SMTP provides distribution-list capabilities by means of an EXPAND (EXPN) command, which has a single argument <string> that identifies a mailing list. A sender SMTP process that must forward mail containing an unrecognized "To:" argument—one that is neither a legitimate mailbox nor a locally maintained alias—opens a TCP connection to a host that knows the mailing list and, following the exchange of greetings, sends an "EXPN <string>" command; if the host

indeed knows how to expand the string, it returns a succession of positive-completion “250” replies, each conveying one mail address. The sender SMTP process concludes this mail session and begins another, sending a succession of “RCPT TO <forward-path>” commands to identify each of the members of the mailing list who should receive copies of the mail. This example, of course, presupposes that the sender SMTP actually *knew* that the argument of the “To:” was in fact a mailing list.

**MIME—Multipurpose Internet Mail Extensions** RFC 1341 defines mechanisms for generalizing the message content of 822 mail to include multiple body parts, which may be both textual and nontextual; i.e., like OSI MHS, the mail contents can be combinations of voice, graphics, and text, and the text can be multifont and multicharacter set. The extensions include:

- A MIME-version header field (like the P2 content-type field in X.400 MHS), to distinguish MIME message contents from 822 message contents.
- A content-type header field, to specify the type and native representation of data in the body of a message.
- A content-transfer-encoding header field, to specify an auxiliary encoding applied to the data to allow them to pass through mail-transport mechanisms incapable of transferring the data in their native representation.
- Content-ID and content-description header fields, two optional header fields to further describe the data in a message body.

MIME will support the following content-types:

- *Text*: textual information in many character sets, and possibly formatted.
- *Multipart*: several body parts, possibly of differing types of data, combined in a single message.
- *Application*: application-specific or binary data.
- *Message*: an encapsulated mail message.
- *Image*: still images, “pictures”.
- *Audio*: audio or voice.
- *Video*: video, composite audio/video, or moving image data.

MIME specifies two encodings for the extended content types. Where data largely consist of octets that correspond to printable ASCII characters, MIME recommends a *quoted-printable* encoding; mail systems will process such encodings without modification, leaving the encoded version in a mostly human-readable form. Where data consist of arbi-

trary octet strings, MIME recommends *Base64*, a variant of the encoding scheme from RFC 1113; briefly, and proceeding from left to right, 24 bits are grouped together, represented as output strings of four encoded 6-bit characters, and then translated into a single alphabetic character from the base-64 ASCII set.

MIME also describes extensions that permit the use of character sets other than ASCII in text tokens of 822 header fields such as “Subject” or “Comments,” within a comment delimited by “(” and “),” and in a word or phrase in a “From,” “To,” or “cc” header field. Both the header and body contents extensions are intended to be compatible with existing mail implementations. MIME uses the mail header fields defined in RFC 822, leaving the field names intact and in ASCII, but extends the encoding of the field body by introducing the notion of an *encoded word*, which, although transparent to mail systems that do not implement MIME, conveys semantics in addition to the header field—e.g., a character set and encoding. Specifically, an encoded word begins and ends with an ASCII “=,” and three arguments—character set, encoding, and encoded text—are bounded by an ASCII “?” (there are thus always four “?”s) and terminated by a SPACE or new line. The *character sets* include US-ASCII and the ISO/IEC 8859 (1987) ISO 8859 family of character sets. Like the MIME message body parts, the *encoding* is either an ASCII “B,” for Base64, or “Q,” for “quoted-printable.” The *encoded text* is any printable ASCII character string. (Using “Q,” however, there are some constraints; i.e., you cannot embed a “?” or SPACE in the string, and other characters—“/,” “\,” “<,” “>,” and “@”—are illegal in header fields, where they are significant.) Thus, using encoded text in the field body of the “To:” header field in the following fashion

```
To: =?ISO8859-1?Q?Andr=E9_?= Pirard <PIRARD@vm1.ulg.ac.be>
```

---

## Security

allows one to encode the name “André Pirard” without sacrificing the accent mark over the *e*. The loss of transparency in field bodies is small in comparison to the gain for mail users who benefit from the extensions.

MIME is a promising and valuable method of enhancing the interworking between Internet mail and the OSI Message Handling System: with it, the multimedia aspects of MHS can be extended to environments where Internet mail is preferred.

RFC 1113, *Privacy Enhancement for Internet Electronic Mail: Part I—Message Encipherment and Authentication Procedures*, describes how to provide confidentiality, authentication, and message integrity assurance using cryptographic techniques on messages exchanged between originator and

recipient user agent processes in environments where RFC 822 mail messages are used. Like many extensions to TCP/IP applications having a large embedded base, privacy-enhanced mail (PEM) is designed so that the mail-transfer agents—SMTP processes or any message-transfer system that supports RFC 822 mail message formats—aren't affected by the deployment of the extensions. PEM provides data confidentiality (protection against unauthorized disclosure of a message and certain mail header fields); sender authentication (corroboration that the originator of a mail message is indeed who he or she claims to be); message integrity (proof that the message has not been tampered with); and if asymmetric key management is used, nonrepudiation of message origin (proof of the integrity and origin of the message). These privacy facilities are provided by encoding a set of header fields to carry the cryptographic control information and an encrypted message and conveying these as the text portion of an RFC 822-formatted mail message (the encrypted message is encoded in printable form). (See also RFC 1422, *Privacy Enhancement for Internet Electronic Mail: Part II—Certificate-based Key Management*, RFC-1423, *Privacy Enhancement for Internet Electronic Mail: Part III—Algorithms, Modes, and Identifiers*, and RFC 1424, *Privacy Enhancement for Internet Electronic Mail: Part IV—Key Certification and Related Services*, which supercede RFCs 1113, 1114, and 1115.)

---

## Interworking between MHS and Internet Mail

In general, interworking between mail applications involves any or all of the following:

- Mail address translation—e.g., from OSI MHS O/R names to Internet mailbox addresses.
- Protocol mapping—e.g., from RFC 822 Internet text message format to an X.400 interpersonal message (P2), or SMTP to X.400 message transfer access protocol (P1).
- Message content handling—the preservation of content type of all message content parts supported in one message transfer system to another—e.g., the preservation of the text content of an RFC 822 message across an X.400 MHS as content type 2.

Much of the useful work in interworking between OSI MHS and Internet mail systems has been performed in the IETF message-handling working groups and documented by Steve Hardcastle-Kille (RFC 987, RFC 1026, RFC 1137, RFC 1327), who has also done extensive work in the

development of mail interworking for the U.K. academic community, which uses the Joint Network Team or Grey Book mail system (Kille 1984a, 1984b). The most recent proposed standard, RFC 1327, defines interworking or mapping between X.400-1988 and RFC 822, with backward compatibility with earlier mappings to X.400-1984-based mail systems.

A comparison between Figures 8.4 and 8.13 suggests that the “header information” used when composing electronic mail in these two systems is quite similar. The mappings between the RFC 822 message header and the interpersonal message system protocol (P2) described in X.420-1988 when an 822 message system is the point of mail origin are accomplished by mapping the RFC 822 header into an extension field in the IPM. When an X.400 MHS is the point of mail origin, mappings are accomplished by (1) mapping existing RFC 822 header fields onto corresponding IPMS protocol information and (2) introducing extension header fields where required. Currently, multipart bodies are supported, but with some loss of information; with MIME extensions, however, it is anticipated that further investigation into interworking between MIME and the OSI MHS will yield mappings that will preserve multipart messages, as well as messages containing multimedia body parts.

Mapping of an RFC 822 mail address is onto an X.400 O/R address. The simplest incarnation of this mapping assumes that the country, ADMD, PRMD, organization, and organizational unit attributes in an O/R address are present; these are mapped to elements in the domain part of an 822 address. The personal name is mapped to the local or user part. For example, it isn’t that difficult to see how the following e-mail addresses can be converted into equivalent O/R addresses:

<b>S.Kille@cs.ucl.ac.uk</b>	<b>mdavies@nri.reston.va.us</b>
C = “GB”	C = “US”
ADMD = “GOLD 400”	ADMD = “ATTCOM”
PRMD = “ac”	PRMD = “va”
O = “UCL”	O = “reston”
OU = “cs”	OU = “nri”
PN = “S.Kille”	S = “mdavies”

Of course, when attributes corresponding to local and domain parts are absent or ambiguous—e.g., a mail address of the form “dave@mail.bellcore.com” or “tredysvr!dvnspl!dvncnms!lap@gvlv2.gvl.unisys.com”—things get stickier. RFC 1327 devotes considerable attention to the details of providing gateway mappings

between complex RFC 822 addresses and O/R addresses and also describes methods of mapping between directory names and RFC 822 addresses.

RFC 1327 is unlikely to be the last of the 822-to-MHS interworking documents; the current incarnation of interworking does not yet address security extensions or dealing with different and multiple message content types. These are likely to be more important as field experience with X.400-1988, MIME, and privacy-enhanced mail increases. In any event, don't be surprised if you begin to receive via your Internet mailer mail headers such as those shown in Figure 8.15.

```

From Alf.Hansen@delab.sintef.no Tues May 18 07:27:58 1993
Return-Path: <Alf.Hansen@delab.sintef.no@sabre.bellcore.com>
X400-Received: by mta mhs-relay.cs.wisc.edu in /PRMD=XNREN/ADMD= /C=US/;
    Relayed; Tues, 18 May 1993 06:21:04 +0000
X400-Received: by /PRMD=uninett/ADMD= /C=no/; Relayed;
    Tues, 18 May 1993 06:18:10 +0000
X400-Received: by /PRMD=uninett/ADMD= /C=no/; Relayed;
    Tues, 18 May 1993 06:18:06 +0000
Date: Tues, 18 May 1993 06:18:06 +0000
X400-Originator: Alf.Hansen@delab.sintef.no
X400-Recipients: non-disclosure:;
X400-Mts-Identifier: [/PRMD=uninett/ADMD= /C=no/;930518131806]
X400-Content-Type: P2-1984 (2)
Content-Identifier: 2483
Conversion: Prohibited
From: Alf Hansen <Alf.Hansen@delab.sintef.no>
To: Erik Huizer <Erik.Huizer@surfnet.nl> (IPM Return Requested)
Cc: dave <dave@sabre.bellcore.com> (IPM Return Requested),
    skh <skh@merit.edu> (IPM Return Requested),
    "Kevin.E.Jordan" <Kevin.E.Jordan@mercury.oss.arh.cpg.cdc.com> (IPM Return Requested),
    "S.Kille" <S.Kille@cs.ucl.ac.uk> (IPM Return Requested),
    "Harald.T.Alvestrand" <Harald.T.Alvestrand@delab.sintef.no> (IPM Return Requested),
    sjt <sjt@gateway.ssw.com> (IPM Return Requested),
    Megan Davies <mdavies@nri.reston.va.us> (IPM Return Requested)
In-Reply-To: <9305151310.AA20876@survival.surfnet.n>
Subject: Re: Boston IETF scheduling
Status: RO

Erik,
. . .

```

FIGURE 8.15 822 Headers from an 822-MHS Relay

---

## Conclusion

This chapter has examined the electronic mail and message handling services of OSI and the Internet. The evolution of the OSI MHS functional model and protocol architecture was traced from the 1984 version to that

---

described in X.400-1988. (Readers should note that the inability on the part of standards makers to arrive at a consensus on the OSI application-layer structure in 1984 introduced perturbations not only in the MHS architecture but in the entire OSI upper-layer architecture; these are discussed in Chapters 10 and 11.) The 1988 version of the OSI MHS is a substantial improvement over its 1984 ancestor, providing a useful, scalable, and secure framework for multimedia messaging.

SMTP/822, or "Internet mail," has evolved from a humble text-only platform to a message handling service that is seemingly feature for feature the equal of OSI MHS by the introduction of privacy enhanced mail and MIME. When PEM was introduced, and MIME shortly thereafter, hard-line Internetters were quick to proclaim X.400 dead, while equally hard-line OSI types dismissed these SMTP extensions as yet another hack. There is, in fact, more posturing than truth in such statements. Readers are asked to consider whether it is really important that there be a winner or whether the true benefit of having two equally powerful message-handling services is that interworking between them will result in a considerably more robust and globally interconnected electronic mail system.