# MIME: A Portable and Robust Multimedia Format for Internet Mail

Nathaniel S. Borenstein

<nsb@bellcore.com>

Bellcore Room MRE 2D-296

445 South St.

Morristown, New Jersey 07960

for *Multimedia Systems* journal

## Abstract

Although multimedia systems are rapidly becoming more widely available and affordable, most present-day multimedia applications are not designed with interoperability in mind. Multimedia data from one application can be used in another only in relatively specialized circumstances or with special translator software, effort, and expense. While this situation is undesirable, it is not a critical impediment to the use of multimedia in many application domains. In others, however, the lack of interoperability is crucial, and nowhere is this more the case than for multimedia electronic mail.

MIME (Multipurpose Internet Mail Extensions), the new standard format defined by an Internet Engineering Task Force Working Group, offers a simple standardized way to represent and encode a wide variety of media types, including images, audio, video, and

non-ASCII textual data, for transmission via Internet mail. MIME extends Internet mail in a manner that is simple, completely backward-compatible, yet flexible and open to extension. In addition to enhanced functionality for Internet mail, the new mechanism offers the promise of interconnecting X.400 "islands" without the loss of functionality currently found in X.400-to-Internet gateways. This paper describes the general approach and rationale of the new mechanisms for Internet multimedia mail.

## Motivation: Interoperable Multimedia Mail

Electronic mail is one of the most widely-used services on almost every computer network, including the Internet. The Internet standard for message formats, RFC-822 [4], is used widely beyond the boundaries of the Internet itself. However, the vast majority of electronic mail traffic is limited to US-ASCII text only. Missing, for most users, is the ability to send pictures, audio, or even text in most non-English languages.

This limitation is not necessary. Even a relatively unsophisticated computer user has little trouble understanding the concept and utility of enclosing pictures in a mail message, for example. This concept is easily extended to include virtually any other type of digital data. Research systems, including Andrew [2], Diamond/Slate [5], and many others, have demonstrated the feasability of much richer mail on top of RFC 822. However, unlike other multimedia applications, electronic mail is critically dependent on standardization, as the utililty of a "standalone" mail system would be very low. What has been prominently missing, however, is format standards that would allow such systems to interoperate. For example, each of the research prototype multimedia mail systems allowed its users to send and receive images, but not in a format that would be recognized by any of the other systems. The recognition of this problem led gradually to a widespread consensus that a standardized format for multimedia mail exchange was necessary. The format devised by the Internet community for this purpose became known as MIME, the Multipurpose Internet Mail Extensions, and is the subject of this paper.

Of course, electronic mail is not the only application that could benefit enormously from the widespread adoption of a standard format for multimedia data exchange. However, it is a logical application to pioneer the development of such a format. To begin with, standards are more critical to multimedia mail than to other applications. Further, electronic mail systems operate under a huge and diverse set of constraints. In particular, email needs to work interoperably on virtually every kind of computer ever built. This adds enormous complexity when one considers machines with unusual byte or word size, the need for ASCII-EBCDIC interoperation, the frequent limitation of 7-bit paths for data exchange, and so on. Beyond the technical problems posed by this kind of extreme heterogeneity, there are political problems. If one image format is commonly used on UNIX, for example, while another is commonly used on DOS, then the choice of a standardized image format can become a political issue that might be seen as favoring one platform over another. Because of all these constraints, it is far more likely that a mail-oriented multimedia interchange format will work for non-mail applications than vice versa.

In fact, within a few months after the publication of the MIME standard, MIME was already in use as the basic multimedia format for a half dozen non-mail applications, and its use is being proposed in a wide range of other areas. Thus while the demands of electronic mail were critical to the definition of MIME -- and will play a prominent role, in this paper, in explaining the design of MIME -- the applicability of MIME has a far broader potential range.

## Extending Mail Standards to Support Multimedia

Increasingly, users are aware of the possibility of multimedia mail. In the presence of FAX and voice mail services, it is easy for anyone to imagine a more integrated multimedia mail facility on their computer. More and more, users are starting to request or even demand multimedia mail, and they expect such mail to interoperate. If multimedia mail is to work on the Internet, some kind of extensions to RFC 822 are necessary.

The main alternative to Internet mail is X.400 [9], the international standard for mail transport. Some have suggested that, since X.400 was designed with multimedia in mind, the demand for multimedia should simply be used to force the transition to X.400. This is an oversimplified view, however. To begin with, X.400 has been extremely slow to win acceptance and deployment in the technical community. The established base of Internet mail users and the perceived complexity of X.400 create substantial resistance to such a transition, particularly in North America. Moreover, X.400 systems currently exist mainly as islands in a sea of Internet mail. In order to interoperate, X.400 mail must often be gatewayed into and then out of the Internet. Such gatewaying, in the absence of Internet multimedia mail standards, loses information, because Internet mail has no standard representation for image, audio, or even non-ASCII textual data. Finally, X.400's support for multimedia data has proved to be far les complete than originally advertised. Overall, standards for multimedia Internet mail will benefit X.400 users as well as X.400 opponents, and indeed both groups have cooperated in the creation of the new mechanisms.

In order to appreciate the design of the Internet multimedia mail facilities, one must first recognize some of the constraints. First, there is a strong need for compatibility with existing practice in the Internet mail world. That practice, as defined by RFC 822 (message format) and RFC 821 [7] (SMTP message transport), imposes several important limitations. Both limit messages to 7 bit US-ASCII characters. RFC 822 defines a message as a structured header followed by a single, monolithic text body, which creates problems for multipart mixed-media mail. SMTP imposes further limits on the length of lines within message headers and bodies.

The 7 bit limitation is particularly critical. There are two obvious approaches to alleviating this limitation: one is to encode all data in 7 bit form, using a standard mechanism. The other is to extend the standards to permit 8 bit data. The work described here pursued the former path, on the grounds that backward-compatibility with 7-bit mail will be more easily and widely deployable, with less impact on existing implementations. However, another group, with significantly overlapping membership, is pursuing the idea of 8-bit SMTP, and the groups have taken care to make sure that their work is compatible.

As stated before, there have been several successful but non-standard systems that overcame these limitations and provided multimedia mail. The current work builds on those experiences, generalizing and standardizing their solutions. In particular, it borrows from the work that resulted in RFC 1049 [10], which defined a mechanism for single part non-text mail, and from RFC 1154 [8], which provided a mechanism for multipart mail that, though problematic, demonstrated its feasability and desirability.

## Technical Overview

RFC 822 defines an Internet message as consisting of two parts: a header and a body. The header consists of a series of field names and field bodies, after which a blank line marks the end of the header and the beginning of the body, which (according to RFC 822) consists of only US ASCII text. The following message, for example, has a "From" header field with the author’s name as the field body, a "Subject" header field with a descriptive field body, and a one-line message body:

```
From: Nathaniel Borenstein
Subject: Hello there

Actually, "hello" is all I wanted to say.
```

A major constraint of the working group charged with extending RFC 822 was the imperative that this basic model not be changed. In particular, it was strongly and widely felt that *nothing* in the new document should cause existing mail systems to break. Not only was the header/body model left unchanged, but so too were the syntax and semantics of all of the standard header fields defined by RFC 822.

Given these constraints, there were two basic models for extension. One was to add a single header field that described the structure and type of the body as a whole, no matter how many sub-parts it might consist of. This was the approach of RFC 1154, but it had problems with describing the boundaries between parts and did not appear likely to scale up well to messages with very many sub-parts. Moreover, it did not permit the description of nested parts, a functionality that has proven very useful in systems such as Andrew.

The other approach, chosen by the working group, was that introduced by RFC 1049. That document defined a header field, "Content-type", which marked the entire message body as being a certain type of data. In the absence of a Content-type field, the body was assumed to be US ASCII text, as before.

Although RFC 1049 had been used by several implementations, it was not without problems. The most severe problem was its total lack of support for multi-part mail. RFC 1049 allowed a message body to be specified as containing something other than text, but only one such thing.

MIME generalizes and extends RFC 1049 in several ways. Most important, it defines a new Content-type, "multipart", which can be used to encapsulate several body parts within a single RFC 822 message body. It also goes far beyond RFC 1049 in explicitly describing the set of allowable content-types, which are relatively few, by defining a subtype mechanism for Content-types, by providing for standardized encoding of non-ASCII data, and by explicitly addressing the issue of non-ASCII character sets.

This paper presents only a high-level overview and summary of the new Internet mechanisms. Those interested in the technical details, and particularly implementors, should consult the official MIME document, RFC 1341 [1].

In the following section, the MIME standard content-types will each be discussed in turn. Before going into such details, however, it is worth noting that the very philosophy embodied in MIME's  content-type/subtype mechanism is a substantial departure from earlier attempts to standardize interchange formats.  For example, one such attempt was ODA, the Office Document Architecture [6].  ODA is a single, integrated format for representing enriched text, images, and a few other types of objects.  The standard is large, specifies every detail of  how data may be represented, is not easily extended, and does not offer the ability to include data in other formats that have been found useful in certain communities or on certain platforms.  In contrast, MIME has been designed as a mechanism for encapsulating a wide variety of data types that have been found useful in different environments, and was explicitly designed for easy extension to include new formats that have been found practical in other contexts.  Instead of a monolithic approach, MIME represents an attempt to design a universal format from the bottom up, by incorporating those formats which have already won acceptance for specific media types.

## The Seven Mail Content-Types

MIME defines precisely seven valid Content-types, and requires that any additions to this set be specified in a new, similarly formal standards document.  This restrictiveness is a major change from RFC 1049, which allowed for much freer definition of new content-types. Instead, the new mechanism for extension is to define new subtypes  of established content-types.  In general, implementors are required to register new subtypes with the Internet Assigned Numbers Authority (IANA) to avoid name conflicts.  (However, "private" subtypes, beginning with the letters "X-", may be used freely and without registration.) The advantage of this scheme is that even if the subtype is unrecognized, a mail reader is more likely to be able to do something reasonable if it knows something about the basic type of data involved.

A separate part of the Content-type header field may be used to convey supplemental information that may be either optional or required, depending on the content-type. Such "parameters" are given in keyword=value notation, and are used, for example, to convey information about character sets for text objects. Thus the default message type for Internet mail may be given a MIME content-type of:

```
Content-type: text/plain; charset=us-ascii
```

The seven defined content-type values are:

1. text. This is the default content-type. The default subtype is plain text, with subtypes associated with particular rich text formats. Thus a vendor might use "content-type: text/product-name" for "rich" textual mail, with the understanding that recipients using other mail software might read the raw rich text representation. Importantly, MIME defines a subtype of text, "richtext", that provides a very simple lingua franca for those who wish to experiment in multifont formatted email. A critical parameter for all text subtypes is "charset", which specifies the character set in which the text is written.

2. image. This content-type is for still images. Subtypes are image format names, two of which, "image/gif" and "image/jpeg", are defined by the core MIME standard Mail readers that do not recognize an image format will at least know that it is an image, and that showing the raw data to the recipient is not useful. GIF and JPEG were selected as

the two primary image formats for very practical reasons:  they are currently the most widely-used image formats, with public domain software readily available for most major platforms.

3.  audio.  This content-type is for audio information.  Subtypes are audio format names, one of which is defined by the new document.  This subtype, "audio/basic", denotes single channel 8000 HZ u-law audio data, an intended lingua franca for telephone-quality email audio.  Again, this format was chosen for its ubiquity; it is easily interpreted on nearly any computer with audio facilities.  It is expected, however, that one or more additional subtypes will eventually be defined for higher-quality compressed audio.

4.  video.  Similarly, subtypes correspond to video format names, such as the ones defined by MIME, "video/mpeg" and "video/h261".  MPEG and H.261 are currently the two most popular digital video formats, although it must be admitted that experimentation with digital video, and particularly video mail, is still in its infancy.

5.  message.  This content-type is to be used to encapsulate an entire RFC 822 format message.  For example, it can be used in forwarding or rejecting mail.  The standard defines two subtypes of message:  "message/partial" can be used to break a large message into several pieces for transport, so that they may be put back together automatically on the other end, and "message/external-body" can be used to pass a very large message body by reference, rather than including its entire contents.  (The contents may be referred to using one of a number of standard mechanisms, such as wide area file systems or file transfer protocols.)  It should be noted that a message with "Content-type: message" may contain a message that has its own, different Content-type field -- that is,

the message structure may be recursive. The use of the "message" type for encapsulation of partial or external data, therefore, provides an easy recursive mechanism for specifying the type of data in the partial or external entity. The various subtypes of "message" have much of their critical data specified as parameters on the content-type header field.

6. multipart. This content-type is used to pack several parts, of possibly differing types and subtypes, into a single RFC 822 message body. The Content-type field specifying type multipart also includes a "boundary" parameter, which is used to separate each consecutive body part. Each body part is itself structured more or less as an RFC 822 message in miniature -- in particular, possibly containing its own Content-type field to describe the type of the part. Subtypes of multipart are specifically required to have the same syntax as the basic multipart type, thus guaranteeing that all implementations can successfully break a multipart message into its component parts. An expected use of subtypes of multipart is to add further structure to the parts, to permit a more integrated structure of multipart messages among cooperating user agents.

7. application. This content-type is to be used for most other kinds of data that do not fit into any of the above categories, such as list servers, mail-based information servers, and mail-based application languages such as Bellcore’s ATOMICMAIL language[3].

## The Content-Transfer-Encoding Field: Binary Data and Seven-Bit Transport

If Internet mail transport (SMTP, as described by RFC 821) is ever upgraded to permit arbitrary binary data of unlimited line length in message bodies, the issue of encoding a message for transport will go away. However, even those who are working towards such changes to SMTP generally recognize that they will be slow in coming. In the meantime, there is wide perception of the need for a standardized mechanism for encoding arbitrary binary data for mail transport.

MIME defines a new header field for this purpose, Content-Transfer-Encoding, which can be used to specify the encoding technique that has been used to render binary data in short lines of seven bit data. After much debate, the working group settled on two encodings, either of which may be used interchangably. One of them, the "base64" encoding, encodes each three bytes of binary data as four bytes of 7-bit data, using a base 64 alphabet selected for maximum portability across SMTP implementations, including ASCII to EBCDIC gateways. The other, the "quoted-printable" encoding, is a less efficient representation that preserves nearly all 7-bit ASCII characters as themselves. It is expected that base64 will be preferred for genuine binary data, while quoted-printable will be preferred for data that is largely US-ASCII, but has scattered non-ASCII characters within it. In particular, this may be the preferred encoding for textual email in the national-use variants of ASCII, ISO 8859-X.

If the Content-Transfer-Encoding field appears in the RFC 822 message header, it refers to the body of the message. If it appears in the "header" area of one part of a multipart message, it refers to the body area of that part only. The Content-Transfer-Encoding field is prohibited when the Content-type field has a value of "multipart" or "message". This is necessary in order to prevent nested encodings, as described later in this paper.

## An Example of a Multipart Message

The following example shows the format of a multipart message. This message has three top-level parts, to be displayed serially: an introductory plain text part, an embedded multipart segment, and a closing text in a non-US-ASCII character set. The embedded multipart piece itself has two parts to be displayed in parallel (if possible), a picture and an audio fragment.

```
From: Nathaniel Borenstein <nsb@bellcore.com>
Subject: A MIME message
MIME-Version: 1.0
Content-type: multipart/mixed; boundary=tweedledum

This text is in the multipart "prefix" area and might
be invisible to many users.
--tweedledum

This is a multipart message.  This is US-ASCII text
because it is not marked otherwise.
--tweedledum
Content-type: multipart/parallel; boundary=tweedledee

This text is in the multipart "prefix" area and might
be invisible to many users.
--tweedledee
Content-type: audio/basic
```

```
Content-Transfer-Encoding: base64
Content-Description:  An Audio Message

... base64-encoded audio data goes here...
--tweedledee
Content-type: image/gif
Content-Transfer-Encoding: base64

...base64 encoded gif image data goes here...
--tweedledee--
--tweedledum
Content-type: text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: quoted-printable

It’s nice to be able to actually use a name like
"Lagerstr=F6m="  in the mail!
--tweedledum--
```

Figure 1 shows how an integrated MIME reader, the Andrew "messages" program displays this example message. Note that the audio fragment is displayed as a button the user clicks on to hear the audio. Figure 2 shows how a non-integrated MIME reader, the metamail program, views the same message. Note that the user was offered a chance to listen to the audio but declined, knowing that it would in any event not show up very well in Figure 2.

## The MIME Extension Mechanisms

The authors of MIME recognized early on that they were unable to fully specify all the different types of data that would ultimately be needed and desired for use in electronic mail. Instead of attempting to do so -- which was the basic approach taken in the definition of X.400, ODA, and most other relevant approaches to standardization -- they focused instead on ensuring that MIME would be easily extended to accomodate additional formats (typically as new subtypes) as they became relevant. Thus, for example, the lack of a high-quality audio subtype in MIME was not considered critical by the authors, as long as it is easy for the audio experts to define such a subtype and have it added to MIME in the future.

MIME extension works by a simple process. The interested parties work together to define the format details of a new MIME subtype, an publish those details as an Internet RFC, or request for comments. Among the details they must specify are the type/subtype name by which the new format will be known. This type/subtype pair -- along with any required or optional parameters for the content-type header field -- are registered with the Internet Assigned Names Authority (IANA) to ensure that the type/subtype names are unique identifiers for a given format. Such formats are considered "experimental" unless and until the RFCs that define them have passed through the Internet standards process and been declared as official Internet standards.

In general, it is expected that most of what will be registered with IANA will be subtypes of the seven existing MIME types, but it is possible that the development of new media types (such as odor or virtual reality) might necessitate the addition of a few new top-level types eventually. IANA will also register new values for certain critical MIME parameters such as character sets.

## Controversies and Problems

Not surprisingly, in an effort to devise a standard approach to a shared multipart multimedia mail facility, there have been a number of controversies and problems along the way. In order to understand the design of this approach, it is helpful to understand some of the discussions that led to its current structure.

One thing worth noting initially, however, is that the MIME effort was largely able to avoid "holy wars" between advocates of two different formats. Not a lot of time was wasted, for example, on the question of the relative merits of GIF and JPEG as image formats. Instead, the MIME mechanisms allow the sender to use either format (or, using multipart/alternative, to use both formats), and expect that mail readers will evolve to support all the formats that come into widespread use. Because MIME is an open-ended evolutionary framework for encapsulating data of multiple types, holy wars were largely avoided.

Several other issues were less easily skirted, however:

**Nested encoding:** In an early draft of MIME, encodings were permitted on an entire message or any of its sub-parts. Ultimately, however, encodings were forbidden when the content-type was "multipart" or "message". If messages or parts of those types were encoded, this would mean that the overall structure of a message -- its breakdown into its smallest constituent parts -- might not be visible without a decoding operation, a situation which many people found unacceptable. Moreover, this could lead to nested encodings, where the same data was passed multiple times through an encoding algorithm and had to be decoded several times as well. The inefficiency of this possibility was distressing to many. One consequence of the ban on nested encoding may be somewhat increased complexity for gateways. A gateway between a hypothetical future 8-bit SMTP mail world and a 7-bit world would presumably have to encode 8-bit messages; in the case of multipart messages, it will now have to parse the body and encode only the lowest-level parts, rather than simply passing the whole message through an encoding. However, this was felt to be less troublesome than the consequences of permitting nested encodings, possibly because the discussants were generally more concerned with user agents than with hypothetical future gateways from a transport system that does not yet exist.

**Compression:** It was widely felt that, in addition to quoted-printable and base64, there should be a compressed encoding mechanism, possibly based on the UNIX compress facility. No such mechanism is currently specified, both because of a lack of compression expertise among the authors and because of the uncertain legal/patent situation regarding the commonly-used compression algorithms.

**Uuencode:** Initially, many people questioned the use of the new base64 scheme instead of the widely-available "uuencode" mechanism. Uuencode was considered and rejected for several reasons. It is nowhere well-specified, and in fact there are several uuencode implementations that do not interoperate. The output of uuencode is not robust across many mail gateways, owing to problems of character set and the cavalier way many relays treat "white space" in message bodies. It is not uncommon for uuencoded files to arrive at a remote site in a form from which they simply cannot be decoded. The base64 format was designed specifically to avoid the problems associated with uuencode.

**Number of Content-types:** There was a significant initial controversy about the overall number of content-types to be permitted. Some favored a "let a thousand flowers bloom" philosophy, while others wanted assurances that mail readers would have a good prospect of recognizing the content-type. The subtype mechanism was a very successful compromise that met the needs of both camps.

**Relation to X.400:** As stated previously, the new Internet mechanisms are the result of a remarkable degree of cooperation between X.400 advocates, die-hard X.400 opponents, and everyone in between. Not surprisingly, many issues of X.400 gatewaying have arisen, but most of these have been deferred to a follow-up document that will specify how X.400 gateways should use and interpret the new Internet mechanisms. However, such issues were discussed long enough to determine that the new mechanisms would not be too problematic for such gateways. Indeed, several aspects of the new mechanisms are expressly designed to facilitate the implementation of such gateways.

**Parts are not messages:** The alert reader may have wondered why the "message" type is necessary, in the presence of "multipart". The answer is that the parts of multipart mechanisms are explicitly specified as not being actual RFC 822 messages, but rather a new type of object (parts) that have very similar syntax. This distinction, it turns out, is important for X.400 gateways. In the absence of this distinction, it is impossible to tell the difference between a multipart message containing an audio part and a multipart message containing an encapsulated message, the body of which is of content-type audio. The part/message distinction allows a more precise semantic mapping between the

Internet and X.400 models.

**Multipart boundaries:** The boundary delimiters that separate the parts of a multipart message have themselves been the subject of a remarkable amount of controversy. After much debate, the current document specifies that the area before the first delimiter and the area after the last delimiter is to be ignored, and that gateways -- particularly gateways to X.400, which has no concept of such a "prefix" or "postscript" -- are free to throw them away. It also specifies that a delimiter string appears as part of the "Content-type: multipart" header field, and that the inter-part delimiter will consist of two hyphens ("--") followed by that string, except that the last delimiter will end with an additional pair of hyphens. Moreover, no such delimiter line is permitted to appear in any of the parts, so that a composing agent must choose its delimiter with care. Although these requirements may seem somewhat baroque, they are not without their reasons. Given the possible presence of a non-meaningful suffix area at the end of the message, the distinguished closing delimiter is particularly important.

**Implications of 8-bit or binary transport:** Those who worked on the extensions to RFC 822 were sharply divided in their opinions regarding the desirability and feasability of 8-bit or binary transport, i.e. extended SMTP. The group’s progress was made, in large part, by agreeing to disagree on this issue. The result is that the current mechanisms will work with 7-bit transport, but will move gracefully into an 8-bit world should 8-bit transport become commonplace in accordance with the mechanisms currently being drafted by the SMTP extensions working group. However, both groups are essentially united in rejecting an alternative approach that has been advocated in a recent Internet Draft [11] that essentially declares SMTP to permit 8-bit data, with no provision for negotiation among SMTP servers to ensure that 8-bit data is not sent willy-nilly to 7-bit implementations.

**"Preferred" encodings:** There have been proposals for a statement that certain encoding types are "preferred" for certain content-types. These have not yet been adopted, largely because it seems likely that common sense will suffice to encourage people to use, for example, base64 rather than quoted-printable when transmitting audio data. There have also been proposals for a mechanism by which mail sent in the future 8-bit world could include a specification of a "preferred" encoding should the mail ever need to be passed off to a 7-bit mailer, but this too can probably be done without.

**Non-ASCII header data:** Many, particularly those from non-English speaking countries, feel strongly that they should be able to use their own character sets in the RFC 822 header area, particularly in human names and in the Subject specification. However, doing so opens up signicant problems in terms of interoperability and compatibility with older systems. A mechanism was finally agreed upon, and was published as a separate document [12] that is generally used as a companion document with MIME.

**Character Set Specification:** Character sets are a perennial source of controversy, and the mail extensions discussion was no different. The working group settled on a relatively small set of "legal" character sets, and hopes to avoid the proliferation of an unnecessarily wide variety of character sets in international electronic mail. It is expected, however, that several more character sets will inevitably be added to the base set defined in MIME, as that set is fundamentally incomplete. (In particular, most of the

MIME designers sincerely hope that ISO 10646 will succeed in becoming the world's universal character set, and will be happy to see the future specification of the use of ISO 10646 in MIME, but were unwilling to pre-specify the details of that usage before ISO 10646 had itself been completely standardized.) There was also extended debate on the proper way to specify the character set syntactically, and whether or not it should be possible to specify, probably nonsensically, that an audio message, for example, uses a specific character set.

## Implementations and Interoperation

Various parties in the Internet mail community have been moving quickly to implement the new standards. A publicly-available implementation by the author adds full MIME support to over a dozen of the most common UNIX and DOS mail readers, including Berkeley Mail, MH, XMH, Elm, Emacs rmail, and Andrew. This implementation supports both encoding mechanisms, the multipart, image, message, audio, text, and text/richtext content-types, and is easily configurable to handle more. At least three other "freeware" versions are already available, and the author is aware of approximately three dozen other implementations, most of them commercial, that were announced or under development as of the writing of this paper. A large number of prominent vendors of email software, in particular, have announced their intention to support MIME.

## Summary and Future Prospects

The Internet RFC 822 extensions are the combined effort of a great many people who share the goal of making multipart, multi-character set, multimedia email widely available on the Internet. They specify mechanisms for including and encoding a wide variety of types and formats of information in mail, but remain flexible and open to future extensions. In particular, they are inherently far more open, flexible, and extensible than solutions based on a single message format such as ODA [6]. Implementation experience so far suggests that the extensions are technically feasible and relatively easy to implement.

Although MIME was designed for the severe constraints imposed by making multimedia data interoperate in the bizarre world of electronic mail, the result is a general-purpose format for representing almost any kind of data in a very robust manner. The growing acceptance of MIME increases the likelihood that software to interpret MIME-format data will be available on a wide variety of platforms, thus increasing the appeal of the format for non-mail applications. There is no reason why MIME could not be used for such non-mail applications as electronic publishing, multimedia database servers, and so on. Indeed, interest in using MIME for such purposes has been wide and growing since the publication of RFC 1341, the official MIME definition.

It remains to be seen, of course, whether MIME will ultimately succeed to make multimedia mail ubiquitous. But if multimedia mail between software and hardware of multiple vendors begins to interoperate regularly and reliably for a significant user community, it is hard to see what benefit any mail software vendors or service providers will find in bucking the trend.

## Acknowledgements

The Internet mail extensions are the product of an Internet Extensions Task Force Working Group. Far too many people participated in that effort to acknowledge all by

name. I particularly want to thank Ned Freed and Greg Vaudreuil for their tireless efforts to bring this work to fruition, and Bob Kraut, Al Buzzard, and Stu Personick, my managers at Bellcore, for allowing me to devote an unexpectedly large amount of time to this effort.

Parts of this paper are derived from a shorter paper published at the ULPAA ’92 conference in Vancouver in May of 1992.

## References

Note that Internet RFC documents referenced below may be retrieved on the Internet via anonymous ftp from any of the following machines:

East Coast (US): nnsc.nsf.net (128.89.1.178)

West Coast (US): ftp.nisc.sri.com (192.33.33.22)

Pacific Rim: munnari.oz.au (128.250.1.21)

Europe: nic.nordu.net (192.36.148.17)

Login with the username "anonymous" and password "guest". After logging in, type "cd rfc". Then type, for example, "get rfc1341.txt" to get a plain text copy of RFC 1341 Internet draft documents are retrieved similarly, except the directory name is "internet-drafts" instead of "rfc".

[1] Borenstein, N.; Freed, N. MIME (Multipuprpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies, RFC 1341.

[2] Borenstein, N; Thyberg, C. "Power, Ease of Use, and Cooperative Work in a Practical Multimedia Message System", International Journal of Man-Machine Studies, April, 1991.

[3] Borenstein, N. Computational Mail as Network Infrastructure for Computer-Supported Cooperative Work, CSCW ’92 proceedings, Toronto.

[4] Crocker, D. Standard for the format of ARPA Internet text messages. August, 1982, Network Information Center, RFC 822.

[5] Forsdick, H.C., Thomas, R.H., Robertson, G. G., and Travers, V. M., "Initial Experience with Multimedia Documents in Diamond", Computer Message Service, Proceedings IFIP 6.5 Working Conference, IFIP, 1984.

[6] ISO, Information processing -- Text and office systems -- Office Document Architecture (ODA), ISO IS 8613, Parts 1-8, March, 1988.

[7] Postel, J. B., Simple Mail Transfer Protocol. August, 1982, Network Information Center, RFC 821.

[8] Robinson, D.; Ullmann, R. Encoding header field for Internet messages. March, 1988, Network Information Center, RFC 1154.

[9] Schicker, P., Message Handling Systems, X.400, Message Handling Systems and Distributed Applications, E. Stefferud, O-j. Jacobsen, and P. Schicker, eds., North-Holland, 1989, pp. 3-41.

[10] Sirbu, M. A., Content-type header field for Internet messages. March, 1988, Network Information Center, RFC 1049.

[11] Ullmann, R. International character support in SMTP. Internet Draft prime-ullmann-smpto-00.txt, March, 1991.

[12] Moore, Keith, Representation of Non-Ascii Text in Internet Message Headers. June, 1992, University of Tennessee, RFC 1342.

Figure 1: How the multipart MIME example looks, using the Andrew "messages" program, an integrated MIME reader.

Figure 2: How the multipart MIME example looks, using the metamail program, a non-integrated MIME reader.

From: Nathaniel Borenstein, Bellcore

To:  P. Venkat Rangan, Managing Editor, Multimedia Systems

Subject:  New version of my paper

Dear Dr. Rangan,

In response to the reviews of my paper, I enclose a new version in which I have made the following changes:

1.  I've changed  the paper's title.

2.  I have removed the keywords entirely.  The reviewer was right that they were incomplete, but I don't actually know what keyword system is being used for this journal anyway. I need further guidance from you on this score, I'm afraid.

3.  I've added text explaining how to get RFC documents.

4.  I've added two illustrations.

I believe that no other changes were suggested.  Please let me know if I can be of any further assistance.

Sincerely,

Nathaniel Borenstein