El correo electrónico o e-mail

Introducción

El correo electrónico o e-mail, como se conoce en ingles, es una de las aplicaciones de uso más común desde el origen de las redes informáticas. Contribuye a la comunicación entre las personas. Se distingue de todas las demás aplicaciones porque los usuarios trabajan sin conexión, lo que significa que las personas que se comunican no tienen por qué estar simultáneamente conectadas.

El correo electrónico se asemeja notablemente al correo postal tradicional. Cuando queremos mandar una carta escrita en papel a alguien, la redactamos tranquilamente en casa, la metemos en un sobre timbrado, escribimos cuidadosamente la dirección del destinatario y la llevamos a una oficina postal. A partir de ahí, los servicios de correo se encargan de que la carta llegue a la oficina postal mas cercana a la dirección del destinatario. Por ultimo, un cartero intenta hacer llegar la carta a la dirección del destinatario. Si no la encuentra, la carta volverá al remitente, y si la encuentra, la carta se pondrá en su buzón. El destinatario la leerá cuando quiera.

Servicios que brinda:

Funcionalidad típica:

- Composición de mensajes (creación, respuesta, ...)
- Transferencia (estableciendo conexiones con la máquina final o con intermedias.)
- Informes (confirmación de no enviado, de recibo, ...)
- Visualización: distintos formatos (texto, imagen, vídeo, ps)
- Acciones programadas (filtrado, reenvío, mensaje de vacaciones...)

Otras funcionalidades habituales:

- buzones de correo
- listas de correo
- prioridad
- cifrado

Direcciones de correo electrónico

Para enviar una carta necesitamos especificar de alguna forma quien es el destinatario de nuestros envíos. Una dirección de correo electrónico esta formada al menos por el nombre de la maquina que maneja el correo del destinatario, y un identificador de usuario reconocido por esa maquina. Por tanto, las direcciones de correo tienen el siguiente formato:

nombre@maquina.dominio

El signo arroba denomina a la preposición inglesa "at", que significa "en". Por "*nombre*" entendemos normalmente el de una persona, o algún identificador de esa persona, conocido por la maquina perteneciente al dominio indicado.

Las direcciones pueden expresarse, para mayor claridad, en la forma:

descripcion1<nombre@maquina.dominio>descripcion2

Las descripciones son ignoradas por los servicios postales, pero podría dárseles alguna utilidad, aunque no se recomienda. Dado que no se indica la ruta hasta la maquina de destino, sino que tan solo se indica el nombre (único) de dicha maquina, a esta notación se la suele llamar **dirección absoluta**.

Existe una notación especial para especificar la ruta de relays que obligatoriamente debe seguir el correo electrónico hasta llegar a la dirección del destinatario. Esta notación es la siguiente:

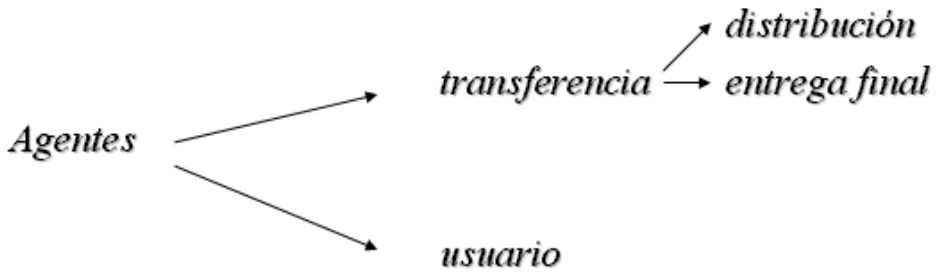
<@maquinaA.dominioA,@maquinaB.dominioB:nombre@dominioC.maquinaC>

La dirección anterior denota como destinatario al identificador de usuario nombre en la maquina C, indicando que se debe llegar a la maquina C a través de maquinaA y maquinaB, en ese orden. Este tipo de dirección se suele llamar una **dirección Route-addr**, de ruta (Route) y dirección (Address).

ARQUITECTURA

Agentes de correo

Normalmente, los sistemas de correo electrónico están organizados en dos subsistemas: los **agentes de usuario de correo** (AU) (En ingles: mail User Agent, MUA) y los **agentes de transporte de correo** (mail Transport Agent, MTA).



Estos agentes se clasifican en:

- De **distribución**: utilizando para ello el protocolo SMTP (Simple Mail Transfer Protocol) RFC 821 o SMTP extendido (ESMTP) RFC 1425
- De **entrega final**: que permita al usuario gestionar su correo a través de una máquina remota, utilizando los protocolos POP3 (Post Office Protocol) RFC 1225 e IMAP (Interactive Mail Access Protocol) RFC 1064

EL AGENTE DE USUARIO (AU ó MUA)

Los MUA o agentes de usuario de correo son programas locales (a veces llamados lectores de correo) que aceptan una variedad de comandos para componer, recibir, y contestar mensajes. Existen incontables lectores de correo electrónico. Algunos funcionan a base de comandos de un carácter desde el teclado (mail, elm, pine, ...). Otros, diseñados para principiantes, asocian a cada uno de estos comandos un icono en la pantalla (Eudora, Microsoft Outlook, Netscape Communicator, ...). Funcionalmente, son iguales.

- Es un programa ("*lector de correo*") más o menos amigable.
- Comandos de una línea
- Interfaz gráfico con iconos

Siempre la misma funcionalidad:

- Enviar mensajes
- Componer mensaje
- Responder a mensajes
- Leer mensajes
- Manipular buzones

Envío de correo electrónico:

- Dirección destino
- Internet: xx@it.uc3m.es
- Lista (local, remota)

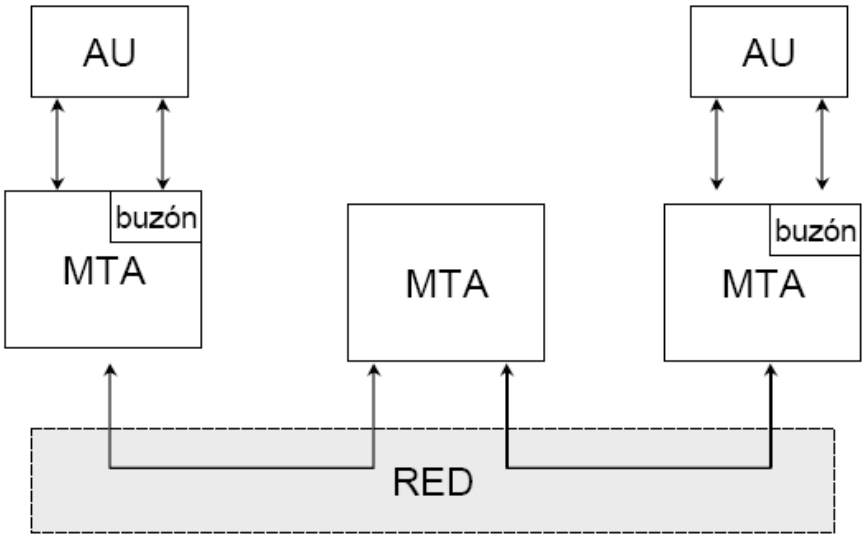
- Mensaje:
- Fichero(s) a transmitir, o entrar en editor.
- Otros: prioridad, seguridad, …

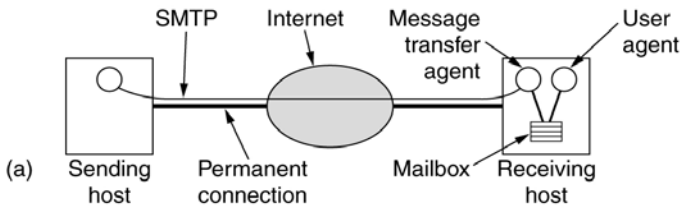
Lectura del correo electrónico:

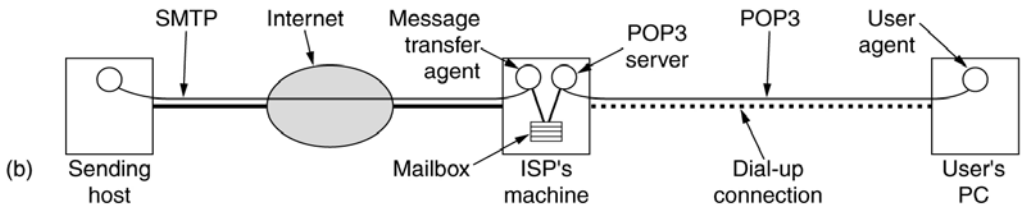
- Busca en el buzón del usuario
- Anuncia número de mensajes
- Breve descripción de cada uno
- Admite comandos

Agentes de transporte de correo (MTA)

Los MTA o agentes de transporte de correo son por lo común daemons del sistema que operan en segundo plano y que mueven el correo electrónico del origen al destino. Hay una gran cantidad de agentes de transporte para sistemas Linux. Uno de los más conocidos es Sendmail, de la Universidad de Berkeley. Sendmail tiene capacidad para actuar también como un agente de usuario.







(a) Enviando y leyendo un mail cuando el receptor tienen una conexión permanente a Internet y el agente de usuario corre en la misma máquina que el agente MTA.

(b) Leyendo un e-mail cuando el receptor tiene una conexión dial-up a un ISP.

Buzones de correo electrónico

Un **buzón de correo** (también conocido como **mail box**) es simplemente un lugar donde se van depositando el e-mail que recibe una persona, y se almacena habitualmente en un fichero del ordenador donde el agente de transporte ha determinado que debe depositar los mensajes para el destinatario de los mismos.

El ordenador en el que se colocan los buzones de correos debe ser un ordenador permanentemente conectado a Internet (al menos, es lo habitual), por lo que rara vez será el ordenador personal del destinatario.

Acceso a buzones de correo electrónico

El destinatario desea leer los mensajes que hay almacenados en el buzón. Si el buzón esta en una maquina Linux, y el destinatario tiene acceso directo a esa maquina (a través de una cuenta de usuario) puede acceder directamente a su buzón vía Linux. Así lo hacen agentes de usuario clásicos de Linux como el mail, elm o pine.

Lo habitual es que el destinatario tenga un acceso restringido al ordenador donde esta depositado su buzón. Entonces puede recurrir a dos protocolos que permiten acceder a los buzones, POP3 e IMAP. Ambos protocolos requieren la presentación de un nombre de usuario y una contraseña, que protegen el buzón de miradas indiscretas. Una vez se ha conseguido el acceso, el comportamiento del POP3 y el IMAP es diferente.

Protocolos de entrega final de usuario

El protocolo POP3

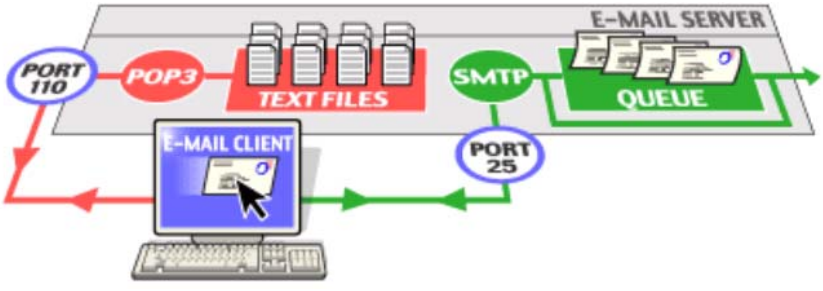
POP3 (Post Office Protocol) RFC 1225 tiene comandos para que un usuario establezca una sesión (USER y PASS), la termine (QUIT), obtenga mensajes (RETR) y los borre (DELE). El protocolo mismo consiste en texto ASCII y se asemeja a SMTP. El objetivo del POP3 es obtener correo electrónico del **buzón remoto y almacenarlo en la máquina local del usuario** para su lectura posterior. *Existen versiones actualmente, que ya permiten no descargar el correo del buzón como IMAP.*

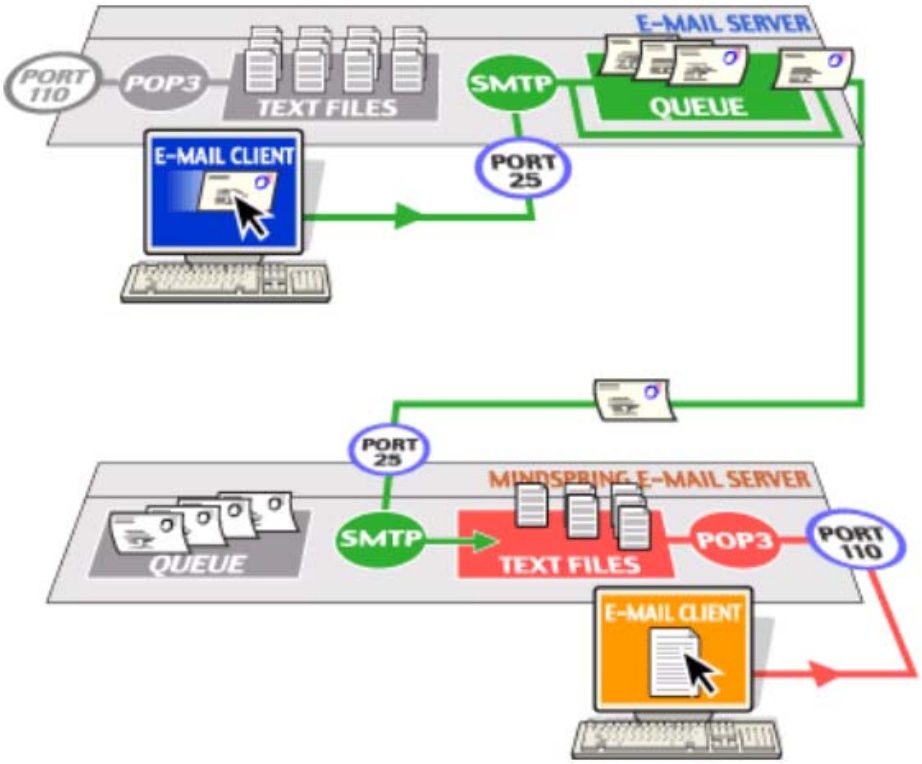
Se emplea principalmente en conexiones con modem en las que el tiempo de conexión cuesta dinero.

De las posibilidades que el POP3 permite, los agentes suelen ofrecer:

- Bajar todos los mensajes y borrarlos del servidor
- Bajar solo los mensajes no leídos y no borrar ninguno del servidor

El inconveniente principal es que, si se ha recibido un mensaje largo, habrá que esperar a que el agente lo traiga desde el buzón para poder ver siquiera de que trata. Este es un problema de como los agentes usan el POP, no del protocolo en si mismo.





El protocolo IMAP

IMAP (Interactive Mail Access Protocol) RFC 1064. La idea en que se basa IMAP es que el servidor de correo electrónico mantenga un depósito central al que puede accederse desde cualquier máquina. Por tanto, a diferencia del POP3, **no copia el correo electrónico en la máquina personal del usuario dado que el usuario puede tener varias computadoras para consultar el correo**, y observa si sus correos han sido leídos con anterioridad.

El IMAP es más potente y eficaz, pero su uso es solo conveniente cuando el costo de la conexión no va en función del tiempo y podemos estar leyendo los mensajes mientras estamos conectados. La gran ventaja del IMAP es que los mensajes están siempre en el servidor.

Cuando el agente se conecta, obtiene la lista de las cabeceras de los mensajes y las muestra al usuario. Este puede entonces ver asunto y remitente y, ello le permitirá leer selectivamente los mensajes y no tener que esperar a traer mensajes largos.

El IMAP tiene otras muchas características. Puede mostrar los mensajes de llegada en función de cualquier atributo. Por ejemplo, “dame el primer mensaje de María”. En este enfoque, un buzón se parece mas a un sistema de base de datos relacional que a una secuencia línea de mensajes.

Una herramienta especialmente valiosa para muchos usuarios de correo electrónico es la capacidad de establecer filtros. Los **filtros** son reglas que se consultan cuando llega el correo electrónico. Cada regla especifica una condición y una acción. Por ejemplo, una regla podría decir que cualquier mensaje que llegue de tu profesor debe presentarse en rojo negrita parpadeante (o alternativamente descartarlo automáticamente sin comentarios).

Otra facilidad importante de IMAP es que permite manejar múltiples buzones auxiliares en el mismo servidor. Esto es útil para el usuario que quiere clasificar los mensajes que quiere guardar. De hecho, los agentes potentes permiten clasificar automáticamente los mensajes recibidos haciendo uso de los filtros.

Otra característica común es la capacidad de instalar un daemon de vacaciones. Este es un programa que examina cada mensaje de entrada y envía al transmisor una respuesta insípida que dice algo así como:

“Hola, estoy de vacaciones. Regresare el 7 de Enero…”

La mayoría de los daemons de vacaciones mantienen un registro de a quienes se enviaron respuestas prefabricadas y se abstienen de enviar a la misma persona una segunda respuesta.

Envío de correo electrónico

Para enviar un mensaje de correo electrónico, el usuario debe proporcionar el mensaje, la dirección de correo electrónico del destinatario y opcionalmente algunos otros parámetros como la prioridad o el nivel de seguridad. El cuerpo de mensaje puede redactarse con un editor de textos independiente o con un editor de textos incorporado en el agente de usuario. La dirección del destinatario debe tener el formato expuesto antes.

Una vez hecho esto, le pedimos al agente de usuario que envié el mensaje. Entonces, el agente de usuario añade una cabecera y contacta con un agente de transporte para que lo transmita. El agente de transporte, que puede estar en otro ordenador distinto, usa algunos campos de la cabecera del mensaje para construir el sobre, y seguidamente intenta transmitir el mensaje al agente de transporte donde se encuentra el buzón del destinatario.

Si lo consigue, perfecto. El agente de reparto eliminara el sobre del mensaje y lo depositara en el buzón del usuario destinatario. Si no lo consigue, no pasa nada. El agente de transporte mantiene una cola de mensajes salientes donde se encuentran los mensajes que no se han podido enviar. Periódicamente procesa la cola e intenta enviar los mensajes. Si pasan varias horas sin conseguir enviar un mensaje, se manda un aviso al remitente avisando de tal situación. El mensaje se intenta enviar durante varios días. Si no se consigue, envía al remitente un error y deja de intentarlo. Lo habitual en la práctica es que mensajes pequeños lleguen en pocos minutos.

ESTRUCTURA DE UN MENSAJE

Estructura básica del correo electrónico

En total analogía con el correo tradicional, los mensajes de correo electrónico se componen de dos partes fundamentales: un **sobre** y un **mensaje**. El sobre encapsula el mensaje. Contiene toda la información necesaria para transportar el mensaje: dirección de destino, prioridad y nivel de seguridad. Los agentes de transporte usan el sobre para enrutar el mensaje (igual que las oficinas postales).

Contenido:

- Cabecera: información de control para AU
- Cuerpo del mensaje: para los humanos

FORMATO DE MENSAJES RFC 822

El RFC 822 define: “Formato estándar de mensajes de texto en Internet”, y data de 1982. Básicamente es un formato de intercambio (puede almacenarse en un formato distinto)

Estructura:

Cabeceras.

- De transferencia: To, Cc, Bcc, From, Sender, Received, Return-Path
- De AU: Date, Reply-To, Message-Id, In-Reply-To, References, Keywords, Subject
- Cada una es una línea de texto ASCII.
- No tienen que ir en ningún orden especial (sí tienen que preceder todas al mensaje)
- Formato Identif: valor (– El valor puede sustituirse un blanco por CRLF + al menos un blanco)
- El usuario puede inventarse sus propias cabeceras, con identificador `X-...`
- Comentarios entre paréntesis (se ignoran).

La cabecera consiste en varias Líneas separadas por caracteres de retorno de carro. Cada línea consiste en un nombre de campo, que comienza en la columna uno, seguido de ":" dos puntos y a continuación un valor que acompaña a la mayoría de los campos. Nótese que a la derecha de los ":" dos puntos hay un espacio en blanco. Los campos pueden aparecer en cualquier orden. Algunos valores son opcionales. Una cabecera típica seria:

- From al006741@llevant.uji.es
- To: al004140@llevant.uji.es (Castell Rovira Ángel Iván)
- Subject:

- Date: Mon, 9 Dec 1996 14:10:08 +0100 (MET)
- Reply-To:
- Message-ID: AA261457010
- Received: by llevant.uji.es (1.37.109.18/16.2)
- From: Foix Prats M. Inmaculada
- X-Mailer: ELM [version 2.4 PL24]

Nótese que la única excepción a la estructura de "campo: valor" indicada antes es la primera línea, que comienza con el campo **From** seguido de un espacio en blanco, en vez de dos puntos. El campo From continua existiendo por compatibilidad con los agentes de usuario antiguos, que se basan en él para marcar el comienzo de un mensaje en el buzón de correo del usuario. Para evitar problemas potenciales con Líneas del cuerpo del mensaje que comiencen también por From se ha convenido en distinguir este último caso precediéndolo de un ">**.**

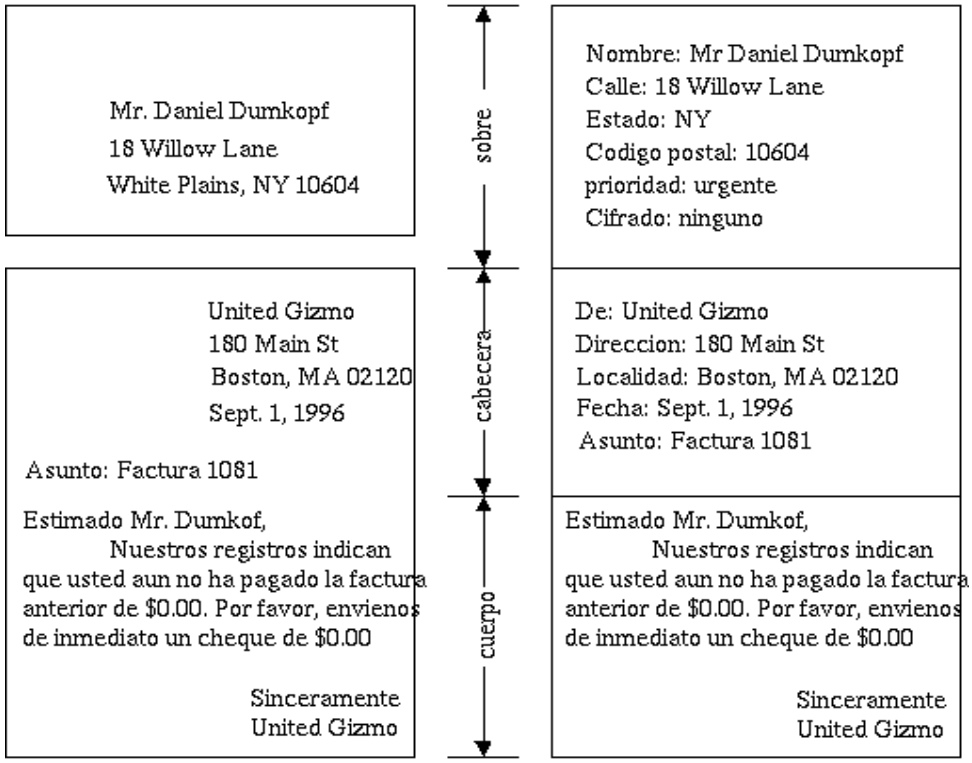
A continuación se indican los principales campos de cabecera y una explicación de cada uno:

- From:
    - Especifica la dirección de correo electrónico del remitente del mensaje.
- To:
    - Especifica la dirección de correo electrónico del destinatario del mensaje. Puede no aparecer cuando son muchos los destinatarios.
- Subject:
    - Especifica el contenido del mensaje en pocas palabras. Es interesante que aparezca, aunque no es obligatorio.
- Date:
    - Especifica la fecha y hora en la que el mensaje fue enviado.
- Reply-To:
    - Especifica la dirección a la que el remitente desea que el destinatario le conteste. Esto puede ser útil si el remitente tienen varias direcciones de correo electrónico, pero desea recibir la mayor parte del correo solo en aquella que se usa más a menudo.
- Message-ID:
    - Identificador unívoco del mensaje en el sistema remitente.
- Received:
    - Cada agente de transporte que procesa el correo electrónico (incluyendo los de las maquinas remitente y destinataria) añaden a este campo su nombre de nodo, de forma que el destinatario pueda conocer la ruta que ha seguido el mensaje.
- X-cualquiercosa:
    - El usuario define un nuevo campo llamado cualquiercosa para uso privado.

Cuerpo del mensaje.

- Sólo líneas de texto (ASCII)
- No considera otros formatos. (Posteriormente extensión MIME.)
- No considera compresión.

La cabecera se separa del cuerpo por una línea en blanco. Tras esa línea en blanco viene el cuerpo del mensaje. Los usuarios pueden poner aquí lo que les venga en gana. Se ha convertido en habitual que mucha gente añada una **firma** (signature) al final del cuerpo del mensaje, usualmente conteniendo información sobre el autor junto con alguna carátula sencilla ASCII, algún comentario y alguna cita celebre.



La figura anterior muestra un ejemplo en el que se compara una carta de correo postal tradicional (a la izquierda) con un mensaje de correo electrónico (a la derecha).

Principales Request For Comments sobre correo electrónico

- **821** Simple Mail Transfer Protocol
- **822** Standard for the format of Arpa Internet text message's
- **1334** Implications of MIME for Internet Mail Gateways
- **1421** Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures
- **1422** Privacy Enhancement for Internet Electronic Mail: Part II: Certificate Based Key Management
- **1423** Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers
- **1424** Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services
- **1425** SMTP service extensions
- **1426** SMTP service extension for 8bit-MIME Transport
- **1427** SMTP service extension for message's Size declaration
- **1428** Transition of Internet Mail from JustSend8 to 8bitSMTP/MIME
- **1502** X.400 use of extended character sets
- **1521** MIME Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies
- **1522** MIME Part Two: Message Header Extensions for Non ASCII Text
- **1556** Handling of Bi-directional Texts in MIME
- **1563** The text/enriched MIME Content type
- **1641** Using Unicode with MIME
- **1642** A Mail Safe Transformation Format of Unicode
- **1651** SMTP Service Extensions
- **1652** SMTP Service Extension for 8 bit MIME Transport

Uuencode

Hasta hace unos años los usuarios que querían transferir por medio de correo electrónico ficheros que contuvieran algo más que caracteres ASCII de 7 bits podían sólo recurrir a codificar -ellos personalmente- en 7 bits sus ficheros.

Muchas personas siguen haciendo eso hoy utilizando los programas *uuencode* y *uudecode*, presentes normalmente en los sistemas Unix, pero de los que también existen versiones para ordenadores con MS-DOS y otros sistemas operativos.

Con estos programas el remitente del mensaje se encarga de codificar el fichero convirtiéndolo en una cadena de caracteres de 7 bits. El fichero resultante es incluido dentro de un mensaje *SMTP* tradicional. Por su parte, el usuario receptor aplica al mensaje recibido un programa decodificador, que sirve para recuperar el fichero original.

Los attachments

Frente a este sistema de codificación "no transparente", existen hoy formas más evolucionadas y cómodas para el usuario. Son, en definitiva, sistemas más "transparentes": requieren menores conocimientos y menos intervención por parte de quien usa las aplicaciones de correo electrónico.

El método más usado es el del *attachment* (añadido). Se trata de lo siguiente: El remitente del correo indica que quiere enviar, junto con su mensaje de texto, un fichero de cualquier tipo (un programa, un gráfico, etc.). El sistema de correo se encarga de codificarlo y transmitirlo, dividiéndolo, si es necesario, en varias partes.

Como es natural, es necesario que el programa de correo electrónico del receptor sea capaz de "entender" la codificación utilizada.

Los sistemas de codificación más extendidos son *BinHex* y *MIME*. *BinHex* se utiliza principalmente entre ordenadores *Macintosh*, aunque también se ha generalizado su uso en otras plataformas informáticas. La aceptación y uso de *MIME* ha sido mucho mayor. Por eso nos detendremos especialmente exponiendo sus características.

MIME – (Multipurpose Internet Mail Extensions)

En los primeros días de ARPANET, el cuerpo del correo electrónico consistía exclusivamente en texto escrito en ASCII de 7 bits (letras, números y signos de puntuación). En este entorno, el formato de cabecera definido anteriormente hacia todo el trabajo.

Con 7 bits (128 posibilidades distintas) es posible representar todos los caracteres necesarios para hacerlo.

Así *SMTP (Simple Mail Transfer Protocol)*, el sistema de correo usado en Internet, regulado por los documentos *RFC-821* y *RFC-822* [los documentos *RFC (request for Comments)* son los que se utilizan para reglamentar las funciones de Internet, con un proceso mucho más ágil y sencillo que el propio de los documentos *ISO (International Standards Organization)*], es un protocolo de 7 bits.

No se previó inicialmente mayor capacidad, pues para la transferencia de ficheros no textuales se contaba con el protocolo *ftp (file Transfer Protocol)*.

Sin embargo, en la red mundial Internet hoy en día este formato ya no es adecuado. Los problemas incluyen envío y recepción de:

- Mensajes en idiomas con acentos (por ejemplo español, francés y alemán).
- Mensajes en idiomas sin alfabetos (por ejemplo, chino y japonés).
- Mensajes que no contienen texto (por ejemplo, imágenes, audio y video).

- Mensajes combinados de los anteriores (por ejemplo texto, una foto y una canción).

MIME

Para facilitar, a través de correo electrónico, el intercambio de ficheros más complejos que los que sólo contenían caracteres de 7 bits, un grupo de trabajo de la *IETF (Internet Engineering Task Force)* comenzó a trabajar en *MIME*.

La idea básica de MIME es continuar usando el formato primitivo de las cabeceras, pero agregando una estructura al cuerpo del mensaje para que contenga diversas partes y definiendo reglas para codificar con ASCII de 7 bits los mensajes que no tengan este formato. Al no cambiar las cabeceras, todos los mensajes MIME pueden enviarse usando los agentes de transporte y protocolos de correo electrónico existentes. Todo lo que hay que cambiar son los agentes de usuario para que interpreten correctamente las cabeceras, algo que pueden hacer los usuarios mismos.

Solución: RFC 1341, Actualizada por RFC 1521
El nuevo estándar ha quedado recogido en dos documentos, aprobados por la *IETF* en 1993: *RFC-1521* y *RFC-1522*. Estos escritos, que no son más que la actualización de las *RFC-1341* y *RFC-1342*, fueron puestos al día, a su vez, en marzo de 1994 por la *RFC-1590*.

MIME no establece un nuevo protocolo. Tan sólo constituye una forma normalizada de intercambio de mensajes electrónicos multimedia. Este sistema es compatible con los programas de correo electrónico que surgieron a partir de la *RFC-821*.

Una característica a destacar es que se trata de un sistema multiplataforma, ya que protege el formato binario del fichero, evitando así el tener que convertirlo a ASCII antes de leerlo.

Con respecto a los primeros sistemas, *MIME* dio un paso adelante en "transparencia", ya que las aplicaciones cliente pueden saber automáticamente de qué tipo de formato de fichero se trata, y, por tanto, la codificación y decodificación se realiza de forma transparente para el usuario.

Por otra parte, con *SMTP* sólo se garantizaba la integridad de mensajes cuya longitud de líneas no excediera los 1000 caracteres. *MIME*, por contra, divide el contenido del mensaje en múltiples partes que se recomponen, también de forma transparente, cuando el mensaje llega al receptor.

Caracteres no ASCII en cabeceras

MIME permite enviar caracteres no ASCII en el cuerpo del mail, no en cabeceras

Tipos de codificación

MIME permite que sea la aplicación del usuario la que elija el tipo de codificación que se utilizará para el contenido del mensaje.

En la cabecera del mensaje existe un campo donde se representa el tipo de codificación elegida. Si, por ejemplo, el campo indicaría lo siguiente:

Content-Transfer-Encoding: quoted-printable

El MIME define 5 nuevos campos en la cabecera del mensaje (para UA), que se muestran a continuación junto con la explicación de su valor.

- MIME-Version:
    - Indica al agente de usuario receptor del mensaje que esta tratando con un mensaje MIME. Valor identifica la version del MIME (actualmente la 1.0). Se considera que cualquier mensaje que no contenga una cabecera MIME-Version: es un mensaje de texto que únicamente contiene ASCII de 7 bits y se procesa como tal.
- Content-Description:
    - Valor contiene una cadena ASCII que especifica el contenido del mensaje MIME en pocas palabras. Esta cadena es útil para que el destinatario sepa si vale la pena decodificar y leer el mensaje o no.

- Content-Id:
    - Valor contiene un identificador unívoco del mensaje. Usa el mismo formato que la cabecera estándar Message-Id.
- Content-Transfer-Encoding:
    - Valor indica al agente de usuario receptor la manera en que está codificado el cuerpo del mensaje para su correcta interpretación. Se proporcionan cuatro esquemas estándar, aunque es posible especificar una codificación definida por el usuario.
    - 7bit
        - Es la codificación más sencilla. Los caracteres ASCII usan 7 bits y pueden transportarse directamente mediante el protocolo de correo electrónico, siempre y cuando ninguna línea exceda 1000 caracteres.
    - 8bit
        - Igual que la codificación anterior pero los caracteres ASCII usan 8 bits. Esta codificación viola el protocolo original del correo electrónico de Internet, pero es usado por algunas partes de Internet que implementan ciertas extensiones al protocolo original. Los mensajes que usan esta codificación aún deben adherirse a la longitud máxima de línea estándar.
    - base64
        - Con este método se aplica una codificación de 8 a 7 bits, de tal forma que de cada tres bytes de entrada se generan cuatro de salida.
        - Los archivos binarios arbitrarios usan 8 bits, pero no respetan el límite de 1000 caracteres por línea. No se da ninguna garantía de que los mensajes en binario llegaran correctamente, pero mucha gente los envía de todos modos. La manera correcta de codificar mensajes binarios es usar **codificación base64**. En este esquema, se dividen grupos de 24 bits en unidades de 6 bits, enviándose cada unidad como carácter ASCII de 7 bits.

Ejemplo**: Content-Transfer-Encoding **base64

Supongamos que deseamos enviar el siguiente fragmento de código binario de un programa

Código binario (en bytes): 214, 04, 23, 97, 08, 200

	214	04	23	97	08	200
Binario	11010110	00000100	00010111	01100001	00001000	11001000

Código Base 64: Dividimos cada grupo de 24 bits en grupos de 6 bits, con lo que resultan lo grupos de seis bits siguientes

110101	100000	010000	010111	011000	010000	100011	001000

Calculando ahora su valor decimal y teniendo en cuenta la codificación en caracteres descrita con anterioridad: 0fPWXPiH

53	32	16	23	24	16	35	8
0	f	P	W	X	P	i	H

    - quoted-printable
        - En el caso de mensajes que son casi completamente ASCII de 7 bits, pero con algunos caracteres del ASCII extendido, la codificación base 64 es ineficiente. En cambio, se usa una codificación conocida como **codificación entrecomillada imprimible**. Simplemente

es ASCII de 7 bits, con todos los caracteres por encima del 127 codificados como un signo "=" igual seguido del valor del carácter en dos dígitos hexadecimales.
        - Se conservan los caracteres de 7 bits. Además se forman las letras especiales de cada idioma con un código de escape y una letra del conjunto primario. Ésta es la opción elegida en idiomas que emplean el alfabeto latino
- Content-Type:
    - Valor que especifica la naturaleza del cuerpo del mensaje. Hay muchísimos valores distintos, pero todos siguen el formato tipo/subtipo. Observe que el tipo y el subtipo se separan mediante una diagonal. Ambos, tanto el tipo como el subtipo, deben indicarse explícitamente, pues no se proporcionan valores predeterminados.

Tipos de formato

En el mundo multimedia existe gran cantidad de formatos, tanto para imágenes como para audio. Éste fue el motivo de que *MIME* se decantara por agrupar estos formatos según el contenido, y, dentro de cada contenido, se escogieron dos o tres subtipos iniciales.

Con el fin de evitar conflictos, las futuras implementaciones de la norma deben registrar nuevos subtipos ante el *Iana (Internet assigned Numbers Authority)*. No obstante, se pueden especificar subtipos particulares utilizando el prefijo *X-*, sin necesidad de que sean aprobados y registrados. Por ejemplo,

- video/x-msvideo
- se utiliza para especificar ficheros de *video for Windows*, el estándar de *Microsoft*, habitualmente reconocible por la extensión *.avi*.

Los siete tipos de contenido definidos originariamente para la norma *MIME* fueron:

Texto

*D*entro de esta modalidad se eligieron los subtipos *plain* (texto ASCII sin formatear) y *richtext*. Se contempló, además, la posibilidad de adjuntar texto realizado mediante procesador. Se admitió la norma *ISO 8859 [1-9]* e *ISO 2022* (para texto Kanji, el alfabeto japonés).

- text/plain, text/html

El tipo texto es para texto normal. La combinación text/plain es para mensajes ordinarios que pueden visualizarse como se reciben, sin codificación ni ningún procesamiento posterior. Esta opción permite el transporte de mensajes ordinarios en MIME con sólo unas pocas cabeceras extra. La combinación text/html es para mostrar paginas Web escritas en html.

Imagen

Los dos formatos que se admitieron en un principio fueron *GIF (Graphics Interchange Format)* y *JPEG (Joint Photographic Experts Group)*. Se eligieron estas dos extensiones gráficas por ser las más extendidas y porque para ellos existe mayor cantidad de software de dominio público utilizable en la mayoría de las plataformas informáticas.

- image/gif, image/jpeg

El tipo image se usa para transmitir imágenes fijas. Hoy día se usan muchos formatos para almacenar y transmitir imágenes, tanto con compresión como sin ella. Dos de los subtipos oficiales son GIF y JPEG, pero sin duda hay muchos más.

Vídeo

El subtipo inicial, al igual que ocurrió con el tipo *imagen*, correspondió a un formato en concreto, el *Mpeg*.

- video/mpeg, audio/basic

El tipo video es para imágenes en movimiento. El único formato de video definido hasta ahora es el diseñado por el grupo de expertos de imágenes en movimiento (moving picture experts Group, MPEG). Note que el tipo video

solo incluye la información visual, pero no la pista de sonido. Si debe transmitir una película con sonido, tal vez sea necesario transmitir las partes de video y de audio por separado. El tipo audio es para sonido.

Audio

El subtipo definido originariamente fue *Basic*, sonido de calidad de telefonía básica, con un único canal de 8000 Hz.

Mensaje

Este tipo se usa para encapsular un mensaje de correo. Contiene tres subtipos; *RFC-822*, *Partial* (para fragmentar un mensaje en varias partes) y *External Body* (textos creados por una fuente ajena al programa de correo).

El tipo **message** permite que un mensaje esté encapsulado por completo dentro de otro. Este esquema es útil para reenviar, correo electrónico.

El subtipo **rfc822** se utiliza cuando se encapsula un mensaje RFC 822 completo en un mensaje exterior.

El subtipo **partial** hace posible dividir un mensaje encapsulado en pedazos y enviarlos por separado. **Los parámetros hacen posible ensamblar correctamente todas las partes en el destino**. Ej: 1/3, 2/3, 3/3.

El subtipo **external-body** puede usarse para mensajes muy grandes, por ejemplo películas de vídeo. En lugar de incluir el archivo mpeg en el mensaje, se da una dirección de FTP y el agente de usuario del receptor puede obtenerlo a través de la red cuando se requiera.

Aplicación

Para cualquier otro tipo de información que no puede ser interpretada como dato binario y necesita ser procesada por una aplicación. Los subtipos originariamente definidos fueron *Octet-Stream*, *Postscript* y *ODA (Open Document Architecture)*.

Multiparte

Para mensajes formados por diferentes tipos de formatos entrelazados entre sí y visualizados mezclados, de forma secuencial o en paralelo (una imagen y un sonido que haga referencia a la acción desarrollada en la imagen).

- multipart/alternative, multipart/mixed, multipart/related

El tipo ***multipart***, permite que un mensaje contenga más de una parte, con el comienzo y el fin de cada parte claramente delimitados.

El subtipo ***mixed*** permite que cada parte sea diferente.

El subtipo ***alternative*** indica que cada parte contiene el mismo mensaje, pero expresado en un medio o codificación diferente.

El subtipo ***parallel*** se usa cuando todas las partes deben “verse” simultáneamente, por ejemplo, en los canales de audio y vídeo de las películas

El subtipo ***digest*** se usa cuando se juntan muchos mensajes en un mensaje compuesto.

La combinación multipart/alternative indica que cada parte contiene el mismo mensaje pero codificado en diferentes formatos. El agente de usuario elegirá la que prefiera. La combinación multipart/mixed indica que las diferentes partes son diferentes, sin ninguna estructura adicional impuesta. La combinación multipart/related indica que las partes son diferentes pero relacionadas, por lo que hay que hacer una interpretación conjunta de las mismas.

En MIME aparece la idea de mandar un mensaje compuesto por múltiples partes, y cada una de ellas a su vez de múltiples partes. Aparece por tanto una estructura recursiva que permite la generación de combinaciones complejas de varias partes. Veamos el siguiente ejemplo:

- From: elinmor@abc.com
- To: carolyn@xyz.com
- Subject: Quería decirte ...
- MIME-Version: 1.0

- Message-Id: <0704760941.AA00747@abc.com>
- Content-Type: multipart/alternative;
- boundary=qwertyuiopasdfghjklzxcvbnm
-
- --qwertyuiopasdfghjklzxcvbnm
- Content-Type: text/plain
-
- Feliz cumpleaños para ti,
- Feliz cumpleaños para ti, querida Carolina
- Feliz cumpleaños para ti.
-
- --qwertyuiopasdfghjklzxcvbnm
- Content-Type: audio/basic;
- directory="tmp";
- name="birthday.snd"
- Content-Transfer-Encoding: base64
-
- MIIC1DCCAn6gAwIBAgIBADANBgkqhkiG9w0BAQQFADCBgDELMAkGA1UEBhMCRVMx
- DzANBgNVBAgUBkVzcGHxYTESMBAGA1UEBxMJQ2FzdGVsbG9uMQ0wCwYDVQQKEwMX
- aXN1MR0wGwYDVQQDExRNYW5vbG8gTW9sbGFyIEdhcmNpYTEeMBwGCSqGSIb3DQEJ
- ARYPbW9sbGFyQG5pc3Uub3JnMB4XDTAwMTEyMzA4MTIxOVoXDTAwMTIyMzA4MTIx
- OVowgYAxCzAJBgNVBAYTAkVTM
- --qwertyuiopasdfghjklzxcvbnm--

En el mensaje anterior se transmite una felicitación de cumpleaños como texto y como canción. Observe como la cabecera Content-Type aparece en tres lugares distintos. En el nivel superior, esta cabecera indica que el mensaje tiene varias partes, y que cada una contiene el mismo mensaje codificado en diferentes formatos. Las partes están delimitadas por dos guiones seguidos de una cadena, definida por el usuario, especificada en el parámetro boundary. Los estándares recomiendan que esta cadena sea única en la historia y en el mundo. El final de las distintas partes se marca con dos guiones seguidos de la cadena boundary seguida de otros dos guiones.

En las dos partes la cabecera Content-Type indica el tipo y el subtipo de la parte. En la primera parte indica que sigue un texto sin formato en ASCII de 7 bits. En la segunda parte indica que sigue un fichero de audio. Observe como en esta parte también se requiere una cabecera Content-Transfer-Encoding para indicar la codificación adecuada para el sonido. Si el receptor tiene capacidad de audio, el agente de usuario decodificara y ejecutara el archivo de sonido birthday.snd.

Subtipos de formatos *MIME* registrados por *Iana*

- ***Text:*** plain, richtext, enriched, tabseparatedvalues, sgml
- ***Multipart:*** mixed, alternative, digest, parallel, appledouble, headerset, formdata, related, report, voice-message
- ***Message:*** partial, externalbody, news
- ***Application:*** octetstream, postscript, oda, atomicmail, andrewinset, slate, wita, decdx, dcarft, activemessage, rtf, applefile, macbinhex40, newsmessageid, newstransmission, wordperfect5.1, pdf, zip, macwriteii, msword, remoteprinting, mathematica, cybercash, commonground, iges, riscos, eshop, x400-bp, sgml, cals1840
- ***Image:*** jpeg, gif, ief, g3fax, tiff
- ***Audio:*** basic, 32kadpcm
- ***video:*** mpeg, quicktime

El grupo de trabajo contempló la posibilidad de incluir dentro de este esquema inicial los ficheros codificados según el método *uuencode*. Se rechazó esta posibilidad por la carencia de una especificación única del método que fuera compatible con todos los programas que había en el mercado.

Se decidió que los ficheros *uuencode* podrían ser incluidos bajo la etiqueta *base64*.

MIME y Web

La clasificación de formatos de ficheros *MIME* se ha utilizado como base para el funcionamiento de los browsers Web. Éstos reconocen los subtipos *MIME* especificados (y otros identificados con prefijos *x-*) y analizan la codificación normalizada por *IANA* para "decidir" qué es lo que se debe hacer con los ficheros que se reciben a través de la Red.

Dentro de la configuración del browser, el usuario puede definir qué aplicaciones serán las encargadas de procesar los diversos ficheros. Así, un fichero de vídeo *quicktime* será visualizado con una aplicación específica que sea capaz de gestionar ese tipo de formato.

El universo X.400

El sistema *X.400*, integrado dentro del esquema *OSI (Open Systems Interconnection)*, se desarrolló, desde sus comienzos en 1984, pensando en el correo multimedia, al recoger la idea de "tipos de contenido" y "partes del cuerpo de un mensaje".

Los tipos de contenido que admite pueden ser voz, *g3fx*, mensaje encapsulado, texto encriptado, *T.61* (colección de caracteres usados en el télex) e *IA5* (conjunto universal de caracteres, que usa 8 bits). A pesar de permitir el uso de voz e imagen, no se especifica el método de codificación de voz, así como tampoco se decanta por ningún formato gráfico en concreto.

El hecho de tener en cuenta las partes del cuerpo de un mensaje supone que el protocolo transfiere las partes de un mensaje como unidades de datos distintas, añadiendo información de control sobre los diferentes tipos de contenido que forman el mensaje.

En contraposición con *MIME*, ***X.400*** es un sistema más robusto y que cuenta con protocolos más complejos.

En el *X.400* de 1988 se definen, de forma específica, varios tipos de contenido básicos, añadiendo un nuevo tipo de cuerpo denominado *EBP3 (Extended Body Part)*, que soporta varios juegos de caracteres (entre ellos *US-ASCII* e *ISO 8859 [1-9]*) y que facilita la posibilidad de definir nuevos formatos fuera de la norma.

A pesar de que en la actualización de 1988 continúe sin estar especificado el formato de imagen y la codificación de voz, *EBP3* permite integrar formatos idénticos a los definidos por *MIME*.

La interoperatividad entre *X.400* y *MIME* pasa siempre por *X.400*. No obstante, en la *RFC-1496* se define un nuevo estándar, llamado *Harpoon*, con el que se pretende conseguir la interoperatividad entre ambos sistemas.

MAPI

La propuesta de Microsoft se denomina *MAPI (Messaging Application Programming Interface)*.

Como su propio nombre indica, se trata de una *API*, una "interfaz para el desarrollo de aplicaciones". Esto es, una serie de funciones ofrecidas por *Microsoft* que pueden ser utilizadas por las distintas empresas que desarrollan aplicaciones de correo existentes para el mercado.

Usando *MAPI*, los diversos sistemas de correo que funcionen bajo *Windows* pueden integrar "objetos" *OLE (Object Linking and Embedding)* dentro del entorno de trabajo diseñado por *Microsoft*.

OLE es un sistema "orientado a objetos" en el que el "documento" es el centro, y todo gira en torno a él. Éste puede integrar en su seno una serie de elementos -textos, gráficos, tablas, imágenes, sonido...- creados por diferentes aplicaciones y que permanecen ligados a ellas.

Cada parte del documento puede más adelante ser modificada utilizando una aplicación distinta, y conservando siempre su relación con el resto de las partes.

Desde el punto de vista orgánico, la información de los documentos se almacena en diferentes ficheros, lo que permite una mayor flexibilidad al sistema: cada aplicación puede operar sobre los correspondientes ficheros.

Una herramienta útil para trabajar con MIME

Linux proporciona la herramienta **mimencode**, muy útil para trabajar con MIME. Se aconseja al lector interesado que vea las páginas man de la aplicación. En cualquier caso, se muestran a continuación algunos ejemplos de como usar esta aplicación:

- mimencode < texto
- mimencode -q < texto
- mimencode -u < coded

El primer ejemplo codifica el fichero “texto” en formato base64. El segundo ejemplo codifica el fichero “texto” en formato quoted-printable. El último ejemplo decodifica el fichero “coded” codificado en formato base64.

Algunos ejemplos de MIME

Ejemplo 1:

- From: none@none.at.all
- Subject: Pruebas-(S)MIME
- To: someone@pretty.world
- MIME-Version: 1.0
- Content-Type: text/plain
- Content-Transfer-Encoding: quoted-printable
- Este es un mail MIME simple

Ejemplo 2:

- From: none@none.at.all
- Subject: Pruebas-(S)MIME
- To: someone@pretty.world
- MIME-Version: 1.0
- Content-Type: multipart/alternative;
- boundary=linux.00.sáb.dic..9.13:11:01.CET.2000
-
- --linux.00.sáb.dic..9.13:11:01.CET.2000
- Content-Type: text/plain
- Content-Transfer-Encoding: quoted-printable
- Alternativa texto
- --linux.00.sáb.dic..9.13:11:01.CET.2000
- Content-Type: text/html
- Content-Transfer-Encoding: quoted-printable
- Content-Disposition: inline
- <html>Alternativa HTML
- --linux.00.sáb.dic..9.13:11:01.CET.2000-

Ejemplo 3:

- From: none@none.at.all
- Subject: Pruebas-(S)MIME
- To: someone@pretty.world
- MIME-Version: 1.0
- Content-Type: multipart/mixed;
- boundary=linux.00.sáb.dic..9.13:25:31.CET.2000
-
- --linux.00.sáb.dic..9.13:25:31.CET.2000
- Content-Type: text/plain
- Content-Transfer-Encoding: quoted-printable
-
- Este mail lleva una pequeña imagen
- --linux.00.sáb.dic..9.13:25:31.CET.2000
- Content-Type: image/gif;
- name="sound2.gif"
- Content-Transfer-Encoding: base64
- Content-Disposition: inline
-
- R0lGODlhKwAmAPf/AP//////zP//mf//Zv//M///AP/M///MzP/Mmf/MZv/MM//MAP+Z//+ZYgoK0
- sWrab651h9lK8jghUd+stdKID/Jlxgi+NlX32ugRrnCFSjxUd8gO1GaUEPzWWFTgl54MQg/VEm
- 2qkpPIdACFN7tkeqKMSVmBbAJEnZaRiOKF8CEJc4AhY4rtijth0kBhtppguDTAlQTSgVFhs+O
- Ga0VgXwjGCILZ8TPFSwsV+JTUVY6EkkH54PILFuaNp0+UUTklAyv0fsWxl6BlU8X+wTyZF
- dj5YgkSSXdY7NYY5FF8JFtJQmUWrXxLPRBAQEAOw==
- --linux.00.sáb.dic..9.13:25:31.CET.2000-

Ejemplo 4:

- From: none@none.at.all
- Subject: Pruebas-(S)MIME
- To: someone@pretty.world
- MIME-Version: 1.0
- Content-Type: multipart/mixed;
- boundary=linux.00.sáb.dic..9.13:25:33.CET.2000
-
- --linux.00.sáb.dic..9.13:25:33.CET.2000
- Content-Type: multipart/alternative;
- boundary=linux.11.sáb.dic..9.13:25:33.CET.2000
-
- --linux.11.sáb.dic..9.13:25:33.CET.2000
- Content-Type: text/plain
- Content-Transfer-Encoding: quoted-printable
-
- Alternativa texto
- --linux.11.sáb.dic..9.13:25:33.CET.2000
- Content-Type: text/html
- Content-Transfer-Encoding: quoted-printable
- Content-Disposition: inline
-
- <html>Alternativa HTML
- --linux.11.sáb.dic..9.13:25:33.CET.2000-

- --linux.00.sáb.dic..9.13:25:33.CET.2000
- Content-Type: image/gif;
- 	name="sound2.gif"
- Content-Transfer-Encoding: base64
- Content-Disposition: inline
-
- R0lGODlhKwAmAPf/AP//////zP//mf//Zv//M///AP/M///MzP/Mmf/MZv/MM//MAP+Z//+ZYgoK0
- sWrab651h9lK8jghUd+stdKID/Jlxgi+NlX32ugRrnCFSjxUd8gO1GaUEPzWWFTgl54MQg/VEm
- 2qkpPIdACFN7tkeqKMSVmBbAJEnZaRiOKF8CEJc4AhY4rtijth0kBhtppguDTAlQTSgVFhs+O
- Ga0VgXwjGCILZ8TPFSwsV+JTUVY6EkkH54PILFuaNp0+UUTklAyv0fsWxl6BlU8X+wTyZF
- dj5YgkSSXdY7NYY5FF8JFtJQmUWrXxLPRBAQEAOw==
- --linux.00.sáb.dic..9.13:25:33.CET.2000-

Ejemplo 5:

- From: none@none.at.all
- Subject: Pruebas-(S)MIME
- To: someone@pretty.world
- MIME-Version: 1.0
- Content-Type: multipart/mixed;
- 	boundary=linux.00.sáb.dic..9.13:25:39.CET.2000
-
- --linux.00.sáb.dic..9.13:25:39.CET.2000
- Content-Type: text/plain
- Content-Transfer-Encoding: quoted-printable
-
- Con 'vcard'
- --linux.00.sáb.dic..9.13:25:39.CET.2000
- Content-Type: image/gif;
- 	name="world2.gif"
- Content-Transfer-Encoding: base64
- Content-Disposition: inline
-
- R0lGODlhKwAnAPf/AP//////zP//mf//Zv//M///AP/M///MzP/Mmf/MZv/MM//MAP+Z//+ZzJkAmZ
- kAZpkAM5kAAGb//2b/zGb/mWb/Zmb/M2b/AGbM/2bMzGbMmWbMZmbMM2bMAGaZ/2aZ
- zAAAmQAAZgAAM+4AAN0AALsAAKoAAIgAAHcAAFUAAEQAACIAABEAAADuAADd
- AAC7AACqAACIf/RAiw4d2uQ/VvlWdFR7uW5HsmlZy55N2269qjwRBcK3gi5tgn4FC5SrD1w
- /uB1XELW8Fvc9mA9/UHXmZ2nqHOgPK4OqhuxHt+GmFYWuJtaYfvh4KJZdIYK05Vf0ghUW
- --linux.00.sáb.dic..9.13:25:39.CET.2000
- Content-Type: text/x-vcard;
- 	name="mm.nisu.vcard"
- Content-Transfer-Encoding: quoted-printable
- Content-Disposition: attachment;
- 	filename="mm.nisu.vcard"
-
- este es un =
- attachment
- --linux.00.sáb.dic..9.13:25:39.CET.2000-

Ejemplo 6 (anidamiento simple):

- From: none@none.at.all

- Subject: Pruebas-(S)MIME
- To: someone@pretty.world
- MIME-Version: 1.0
- Content-Type: multipart/mixed;
- 	boundary=linux.00.sáb.dic..9.16:21:34.CET.2000
-
- --linux.00.sáb.dic..9.16:21:34.CET.2000
- Content-Type: multipart/mixed;
- 	boundary=linux.11.sáb.dic..9.16:21:34.CET.2000
-
- --linux.11.sáb.dic..9.16:21:34.CET.2000
- Content-Type: text/plain
- Content-Transfer-Encoding: quoted-printable
-
- Primer bloque con imagen
- --linux.11.sáb.dic..9.16:21:34.CET.2000
- Content-Type: image/gif;
- 	name="sound2.gif"
- Content-Transfer-Encoding: base64
- Content-Disposition: inline
-
- R0lGODlhKwAmAPf/AP//////zP//mf//Zv//M///AP/M///MzP/Mmf/MZv/MM//MAP+Z//+ZYgoK0
- sWrab651h9lK8jghUd+stdKID/Jlxgi+NlX32ugRrnCFSjxUd8gO1GaUEPzWWFTgl54MQg/VEm
- 2qkpPIdACFN7tkeqKMSVmBbAJEnZaRiOKF8CEJc4AhY4rtijth0kBhtppguDTAlQTSgVFhs+O
- Ga0VgXwjGCILZ8TPFSwsV+JTUVY6EkkH54PILFuaNp0+UUTklAyv0fsWxl6BlU8X+wTyZF
- dj5YgkSSXdY7NYY5FF8JFtJQmUWrXxLPRBAQEAOw==
- --linux.11.sáb.dic..9.16:21:34.CET.2000-
- --linux.00.sáb.dic..9.16:21:34.CET.2000
- Content-Type: multipart/mixed;
- 	boundary=linux.11295.sáb.dic..9.16:21:37.CET.2000
-
- --linux.22.sáb.dic..9.16:21:37.CET.2000
- Content-Type: text/plain
- Content-Transfer-Encoding: quoted-printable
-
- Segundo bloque con imagen
- --linux.22.sáb.dic..9.16:21:37.CET.2000
- Content-Type: image/gif;
- 	name="world2.gif"
- Content-Transfer-Encoding: base64
- Content-Disposition: inline
-
- R0lGODlhKwAnAPf/AP//////zP//mf//Zv//M///AP/M///MzP/Mmf/MZv/MM//MAP+Z//+ZzJkAmZ
- kAZpkAM5kAAGb//2b/zGb/mWb/Zmb/M2b/AGbM/2bMzGbMmWbMZmbMM2bMAGaZ/2aZ
- zAAAmQAAZgAAM+4AAN0AALsAAKoAAIgAAHcAAFUAAEQAACIAABEAAADuAADd
- AAC7AACqAACIf/RAiw4d2uQ/VvlWdFR7uW5HsmlZy55N2269qjwRBcK3gi5tgn4FC5SrD1w
- /uB1XELW8Fvc9mA9/UHXmZ2nqHOgPK4OqhuxHt+GmFYWuJtaYfvh4KJZdIYK05Vf0ghUW
- --linux.22.sáb.dic..9.16:21:37.CET.2000-
- --linux.00.sáb.dic..9.16:21:34.CET.2000-

MIME Seguro (Secure MIME, SMIME)

Cuando un mensaje de correo electrónico se envía entre dos sitios distantes, generalmente transitara por docenas de maquinas en el camino. Cualquiera de estas puede leer y registrar el mensaje para un uso posterior. La confidencialidad es inexistente, a pesar de lo que piensa mucha gente. Igualmente alguien podría modificar información, o hacerse pasar por otra persona, con lo que tampoco podemos estar seguros de la autenticidad de la información.

La única forma de resolver ambos problemas es mediante el uso de la criptografía. Los dos protocolos criptográficos fundamentales empleados en la seguridad de Internet son SMIME para el correo electrónico y el SSL para la Web. Nos centraremos aquí en el SMIME.

SMIME no son más que unos determinados tipos MIME que añaden al MIME la capacidad de firmar y cifrar mensajes de correo electrónico, manteniendo por tanto toda la funcionalidad del MIME en nuestros mensajes seguros. El SMIME permite enviar mensajes firmados, con lo que el receptor estará seguro de que el mensaje proviene de nosotros (autenticidad) y también permite enviar mensajes cifrados, de modo que solo el receptor puede leerlos (confidencialidad). La combinación de los dos tipos anteriores se puede conseguir empleando el anidamiento de tipos MIME, consiguiendo por tantos mails firmados y a la vez cifrados. Muchos browsers tienen hoy día capacidad SMIME, entre ellos Netscape e Internet Explorer.

Content-Type:

- multipart/signed
- El tipo multipart permite que el mensaje tenga varias partes independientes. La combinación multipart/signed indica que es un bloque compuesto de dos partes, un mensaje (que puede tener cualquier estructura MIME o SMIME) y la firma del mensaje. Es tarea del agente de usuario el comprobar que la firma es buena.
- Application/x-pkcs7-mime, Application/x-pkcs7-signature
- El tipo Application es un tipo general para los formatos que requieren un procesamiento externo no cubierto por ninguno de los otros tipos. La combinación Application/x-pkcs7-mime indica que lo que viene esta cifrado. El algoritmo empleado para cifrar el bloque, el algoritmo empleado para cifrar la llave de sesión, así como algunos parámetros adicionales, van incluidos en unas cabeceras que se transmiten en claro junto con el bloque cifrado. Por supuesto estas cabeceras también viajaran codificadas con la codificación MIME que se haya indicado en el campo Content-Transfer-Encoding. La combinación Application/x-pkcs7-signature indica que lo que viene es una firma digital. En este caso las cabeceras que se transmiten en claro son para indicar el algoritmo empleado para realizar el resumen, el algoritmo empleado para firmar dicho resumen, el nombre de la autoridad certificadora, su certificado y otros parámetros adicionales. Es tarea del agente de usuario el comprobar que la firma es buena.

Algunos ejemplos de SMIME

Ejemplo 1 (un mail SMIME firmado):

- From: none@none.at.all
- Subject: Pruebas-(S)MIME
- To: someone@pretty.world
- MIME-Version: 1.0
- Content-Type: multipart/signed;
- protocol="application/x-pkcs7-signature";
- micalg=sha1;
- boundary="linux.00.sáb.dic..9.18:16:59.CET.2000"
-
- --linux.00.sáb.dic..9.18:16:59.CET.2000
- Content-Type: multipart/mixed;

- boundary=linux.11.sáb.dic..9.18:16:59.CET.2000
-
- --linux.11.sáb.dic..9.18:16:59.CET.2000
- Content-Type: text/plain
- Content-Transfer-Encoding: quoted-printable
-
- Este mail firmado lleva una pequeña imagen
- --linux.11.sáb.dic..9.18:16:59.CET.2000
- Content-Type: image/gif;
- name="sound2.gif"
- Content-Transfer-Encoding: base64
- Content-Disposition: inline
-
- R0lGODlhKwAmAPf/AP//////zP//mf//Zv//M///AP/M///MzP/Mmf/MZv/MM//MAP+Z//+ZYgoK0
- sWrab651h9lK8jghUd+stdKID/Jlxgi+NlX32ugRrnCFSjxUd8gO1GaUEPzWWFTgl54MQg/VEm
- 2qkpPIdACFN7tkeqKMSVmBbAJEnZaRiOKF8CEJc4AhY4rtijth0kBhtppguDTAlQTSgVFhs+O
- Ga0VgXwjGCILZ8TPFSwsV+JTUVY6EkkH54PILFuaNp0+UUTklAyv0fsWxl6BlU8X+wTyZF
- dj5YgkSSXdY7NYY5FF8JFtJQmUWrXxLPRBAQEAOw==
- --linux.11.sáb.dic..9.18:16:59.CET.2000-
- --linux.00.sáb.dic..9.18:16:59.CET.2000
- Content-Type: Application/x-pkcs7-signature;
- name="smime.p7s"
- Content-Transfer-Encoding: base64
- Content-Disposition: attachment;
- filename="smime.p7s"
- Content-Description: S/MIME Cryptographic Signature
-
- MIAGCSqGSIb3DQEHAqCAMIIEVgIBATELMAkGBSsOAwIaBQAwgAYJKoZIhvcNAQcBA
- ACgggLhMIIC3TCCAoegAwIBAgIBADANBgkqhkiG9w0BAQQFADCBgzELMAkGA1UEBh
- MCZXMxDzANBgNVBAgUBmVzcGHxYTESMBAGA1UEBxMJY2FzdGVsbG9uMQ8wDQYD
- VQQKEwZtaWNhU0ExDjAMBgNVBAsTBWhhYjAxMQ0wCwYDVQQDEwRub25lMR8wHQ
- YJKoZIhvcNAQkBFhBub25lQG5vbmUuYXQuYWxsMB4XDTAwMTIwOTE3MTQyMloXDTA
- --linux.00.sáb.dic..9.18:16:59.CET.2000-

Ejemplo 2 (un mail SMIME cifrado):

- From: none@none.at.all
- Subject: Pruebas-(S)MIME
- To: someone@pretty.world
- MIME-Version: 1.0
- Content-Type: Application/x-pkcs7-mime;
- name="smime.p7m"
- Content-Transfer-Encoding: base64
- Content-Disposition: attachment;
- filename="smime.p7m"
- Content-Description: S/MIME Encrypted Message
-
- MIAGCSqGSIb3DQEHA6CAMIIBbwIBADGB4zCB4AIBADCBiTCBgzELMAkGA1UEBhMC
- RVMxDzANBgNVBAgUBmVzcGHxYTESMBAGA1UEBxMJY2FzdGVsbG9uMQ8wDQYDV
- QQKEwZtaWNhU0ExDjAMBgNVBAsTBWhhYjAyMQ0wCwYDVQQDEwRub25lMR8wHQY
- JKoZIhvcNAQkBFhBub25lQG5vbmUuYXQuYWxsAgEAMA0GCSqGSIb3DQEBAQUABEBY
- SQjpM+71rcCm3ncESaQ/0CCX1K0bnC/UdxSJvRd9douEjBr1A8DZTCQo2LD0CkCWI4OFqH

- 26Ya0gzsG/ljIXMIGDBgkqhkiG9w0BBwEwFAYIKoZIhvcNAwcECHn5ZIy3ftzBgGB7eKBBw
- 6pDo5EbWeT6CHkAdNv5LsxdyGxDCFAicU6+J4FqxjB+vfwOHHci214XGE9d4elvjdVKB0x7i
- sBnKaEi8Q+wOzM2/Ipjt+QXS2Vdpx72WoqNiR0jIaMl3G

Su estructura es text/plain, pero el cifrado oculta toda la información

Ejemplo 3 (un mail SMIME firmado y cifrado):

- From: none@none.at.all
- Subject: Pruebas-(S)MIME
- To: someone@pretty.world
- MIME-Version: 1.0
- Content-Type: Application/x-pkcs7-mime;
- name="smime.p7m"
- Content-Transfer-Encoding: base64
- Content-Disposition: attachment;
- filename="smime.p7m"
- Content-Description: S/MIME Encrypted Message
-
- MIAGCSqGSIb3DQEHA6CAMIIK9gIBADGCAcYwgeACAQAwgYkwgYMxCzAJBgNVBAYTAmV
- zMQ8wDQYDVQQIFAZlc3Bh8WExEjAQBgNVBAcTCWNhc3RlbGxvbjEPMA0GA1UEChMGbWlj
- YVNBMQ4wDAYDVQQLEwVoYWIwMTENMAsGA1UEAxMEbm9uZTEfMB0GCSqGSIb3DQEJA
- RYQbm9uZUBub25lLmF0LmFsbAIBADANBgkqhkiG9w0BAQEFAARASeD3XBU3TNa3s1EaOrfum
- XIZAg5v6GAf0HTM/wJ2Dgn8uR8omZpSaMPCsWPgcPNpWy2rC86aK3gK+Kuf20Z2ZjCB4AIBAD
- CBiTCBgzELMAkGA1UEBhMCZXMxDzANBgNVBAgUBmVzcGHxYTESMBAGA1UEBxMJY2Fz
- dGVsbG9uMQ8wDQYDVQQKEwZtaWNhU0ExDjAM

Un mail firmado y cifrado, primero se firma y luego el resultado se cifra, con lo que externamente tiene la apariencia de un mail cifrado.

Transferencia de correo electrónico

El sistema de transferencia de mensajes se ocupa de transmitir los mensajes del remitente al destinatario. La manera mas sencilla de hacer esto es establecer una conexión de transporte de la maquina de origen a la de destino y sencillamente transferir el mensaje. Tras examinar la manera en que se hace normalmente esto, estudiaremos algunas situaciones en las que no funciona esto, y lo que puede hacerse al respecto.

SMTP (Simple Mail Transfer Protocol)

El SMTP es un sencillo protocolo **cliente/servidor** en formato **ASCII**. Establecida una comunicación **TCP** entre la computadora transmisora del correo, que opera como cliente, y el **puerto 25** de la computadora receptora del correo, que opera como servidor, el cliente permanece a la espera de recibir un mensaje del servidor.

SMTP está basado en *la entrega punto-a-punto*; un cliente SMTP contactará con el servidor SMTP del host de destino directamente para entregar el correo. Guardará el correo hasta que se haya copiado con éxito en el receptor. Esto difiere del principio de retransmisión común a muchos sistemas de correo en las que el correo atraviesa un número de host intermedios de la misma red y donde una transmisión con éxito implica sólo que el correo ha alcanzado el host correspondiente al siguiente salto.

En varias implementaciones, existe la posibilidad de intercambiar correo entre los sistemas de correo locales y SMTP. Estas aplicaciones se denominan *pasarelas o puentes de correo*. Enviar correo a través de una pasarela puede alterar la entrega punto-a-punto, ya que SMTP sólo garantiza la entrega fiable a la pasarela, no al host de

destino, más allá de la red local. La transmisión punto SMTP en estos casos es host-pasarela, pasarela-host o pasarela-pasarela; SMTP no define lo que ocurre más allá de la pasarela. CSNET proporciona un interesante ejemplo de servicio de pasarela de correo. Diseñada en principio como un servicio barato para interconectar centros científicos y de investigación, CSNET opera una pasarela que permite a sus suscriptores enviar y recibir correo en Internet con sólo un Modem con dial. La pasarela sondea a los suscriptores a intervalos regulares, les entrega su correo y recoge el correo de salida. A pesar de no ser una entrega punto-a-punto, ha demostrado ser un sistema muy útil.

El protocolo SMTP

En Internet, el correo electrónico se entrega al hacer que la maquina de origen establezca una conexión TCP con el puerto 25 de la maquina de destino. Escuchando en este puerto está un daemon de correo electrónico que habla con el SMTP (simple mail transfer Protocol, protocolo sencillo de transferencia de correo). Este daemon acepta conexiones de entrada y copia mensajes de ellas a los buzones adecuados. Si no puede entregarse un mensaje, se devuelve al transmisor un informe de error que contiene la primera parte del mensaje que no pudo entregarse.

El SMTP es un protocolo ASCII sencillo. Tras establecer la conexión TCP con el puerto 25, la maquina emisora, operando como cliente, espera que la maquina receptora, operando como servidor, hable primero. El servidor comienza por enviar una línea de texto que proporciona su identidad e indica si está preparado o no para recibir correo. Si no lo está, el cliente libera la conexión y lo intenta después.

Si el servidor esta dispuesto a aceptar correo electrónico, el cliente anuncia de quien viene el mensaje y para quien esta dirigido. Si existe tal destinatario en el destino, el servidor da al cliente permiso para enviar el mensaje. Entonces el cliente envía el mensaje y el servidor acusa su recibo. Por lo general no se requieren sumas de comprobación porque el TCP proporciona una corriente de bits confiable. Si hay mas correo electrónico, se envía ahora. Una vez que todo el correo electrónico ha sido intercambiado en ambas direcciones, se libera la conexión. Un ejemplo de dialogo para el envío del mensaje del "Feliz cumpleaños", incluidos los códigos numéricos usados por el SMTP, se muestra a continuación. Las Líneas enviadas por el cliente se marcan como C: y aquellas enviadas por el servidor se marcan con S:

• Se abre conexión TCP al puerto 25.

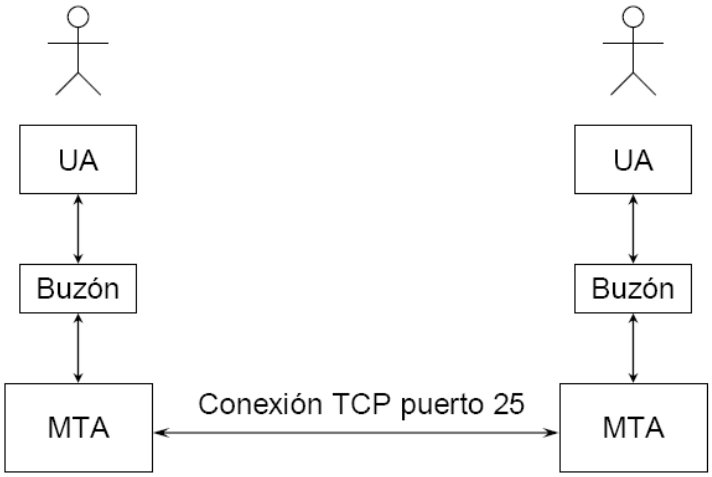
```
        S: 220 servicio SMTP xyz.com listo
C: HELO abc.com
        S: 250 xyz.com dice hola a abc.com
C: MAIL FROM: <elinmor@abc.com>
        S: 250 transmisor ok
C: RCPT TO: <carolyn@xyz.com>
        S: 250 receptor ok
C: DATA
        S: 354 envía correo; termina con una línea únicamente con "."
C: From: elinmor@abc.com
C: To: carolyn@xyz.com
C: MIME-Version:1.0
C: Message-Id: <0704760941.AA00747@abc.com>
C: Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
C: Subject: La Tierra orbita al Sol un numero entero de veces
C:
C: Éste es el preámbulo. El agente de usuario lo ignora. Tenga un buen día.
C:
C: --qwertyuiopasdfghjklzxcvbnm
C: Content-Type: text/richtext
C:
C: Feliz cumpleaños para ti
C: Feliz cumpleaños para ti, querida <bold> Carolina </bold>
C: Feliz cumpleaños para ti
C:
```

C: --qwertyuiopasdfghjklzxcvbnm
C: Content-Type: message/external-body;
C: access-type="anon-ftp",
C: site="bicicle.acb.com";
C: directory="pub";
C: name="birthday.snd"
C:
C: content-type: audio/basic
C: content-transfer-encoding: base64
C: --qwertyuiopasdfghjklzxcvbnm
C:.
S: 250 mensaje aceptado
C: QUIT
S: 221 xyz.com cerrando conexión

El mensaje se envía a un solo receptor, por lo que se usa un solo comando RCPT. Se permiten muchos de tales comandos para enviar un solo mensaje a receptores múltiples; cada uno se acusa o rechaza individualmente. Incluso si se rechazan algunos destinatarios (porque no existen en el destino), el mensaje puede enviarse a los demás.

Por ultimo, aunque la sintaxis de los comandos de cuatro caracteres del cliente se especifica con rigidez, la sintaxis de las respuestas es menos rígida. Solo cuenta el código numérico. Cada Implementación puede poner la cadena que desee después del código.

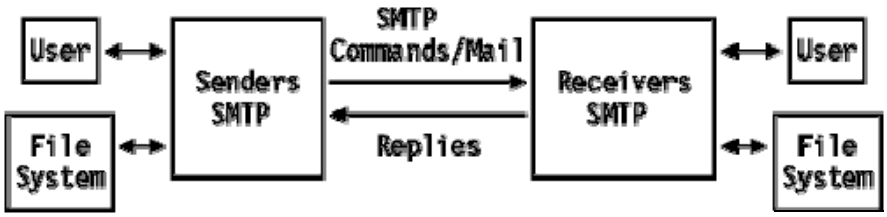
- El UA pasa el mensaje al MTA
- En Internet:
  - MTA típico en UNIX: Sendmail
  - MTA siempre a la escucha en puerto 25
  - Para enviar, MTA local (cliente) establece conexión TCP con destino (servidor)
  - Una vez establecida la conexión se comunican mediante el protocolo SMTP (RFC 821)



Protocolo

El diseño de SMTP se basa en el modelo de comunicación mostrado en la figura. Como resultado de la solicitud de correo de un usuario, el emisor SMTP establece una conexión en los dos sentidos con el receptor SMTP. El

receptor puede ser el destinatario final o un intermediario (pasarela de correo). El emisor generará comandos a los que replicará el receptor.



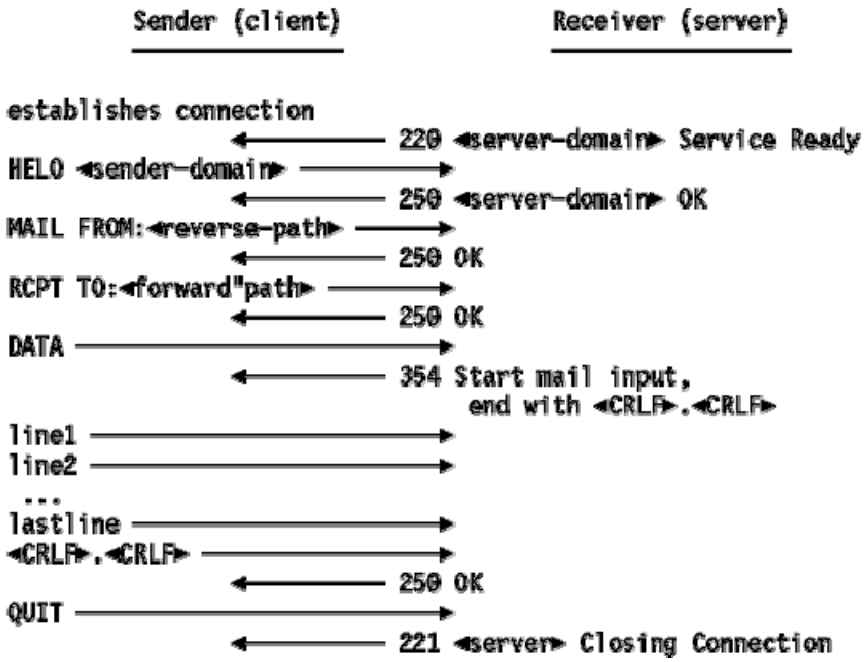
El servidor **comienza por enviar una línea de texto** que proporciona su identidad **e indica si está preparado o no** para recibir correo:

a.-Si no lo está, el cliente libera la conexión y lo intenta después. Por defecto en Sendmail, cada 15 minutos durante 4 días.

b.-Si está dispuesto a aceptar correo electrónico, el cliente anuncia de quién viene el mensaje, y a quién está dirigido. Si existe tal destinatario en el destino, el servidor da al cliente permiso para enviar el mensaje. Entonces el cliente envía el mensaje y el servidor acusa su recibo. Si existe más correo electrónico también se envía ahora. Una vez que todo el correo ha sido intercambiado **en ambas direcciones**, se libera la conexión.

Pasos:

Aunque los comandos y réplicas de correo están definidas rígidamente, el intercambio se puede seguir en la siguiente figura.



Todos los comandos, réplicas o datos intercambiados son líneas de texto, delimitadas por un <CRLF>. Todas las réplicas tienen un código numérico el comienzo de la línea.

1. El emisor SMTP establece una conexión TCP con el SMTP de destino y espera a que el servidor envíe un mensaje *"220 Service ready"* o *"421 Service not available"* cuando el destinatario es temporalmente incapaz de responder.
2. Se envía un HELO (abreviatura de "Hello"), con el que el receptor se identificará devolviendo su nombre de dominio. El SMTP emisor puede usarlo para verificar si contactó con el SMTP de destino correcto.

Si el emisor SMTP soporta las extensiones de SMTP definidas en el RFC 1651, puede sustituir el comando HELO por EHLO. Un receptor SMTP que no soporte las extensiones responderá con un mensaje *"500 Syntax*

error, command unrecognized". El emisor SMTP debería intentarlo de nuevo con HELO, o si no puede retransmitir el mensaje sin extensiones, enviar un mensaje QUIT.

Si un receptor SMTP soporta las extensiones de servicio, responde con un mensaje multi-línea *250 OK* que incluye una lista de las extensiones de servicio que soporta.

3. El emisor inicia ahora una transacción enviando el comando MAIL al servidor. Este comando contiene la ruta de vuelta al emisor que se puede emplear para informar de errores. Nótese que una ruta puede ser más que el par *buzób@nombre de dominio del host*. Además, puede contener una lista de los hosts de encaminamiento. Si se acepta, el receptor replica con un *"250 OK"*.
4. El segundo paso del intercambio real de correo consiste en darle al servidor SMTP el destino del mensaje(puede haber más de un receptor). Esto se hace enviando uno o más comandos RCPT TO:<*forward-path*>. Cada uno de ellos recibirá una respuesta *"250 OK"* si el servidor conoce el destino, o un *"550 No such user here"* si no.
5. Cuando se envían todos los comandos rcpt, el emisor envía un comando *DATA* para notificar al receptor que a continuación se envían los contenidos del mensaje. El servidor replica con *"354 Start mail input, end with <CRLF>.<CRLF>"*. Nótese que se trata de la secuencia de terminación que el emisor debería usar para terminar los datos del mensaje.
6. El cliente envía los datos línea a línea, acabando con la línea <CRLF>. <CRLF> que el servidor reconoce con *"250 OK"* o el mensaje de error apropiado si cualquier cosa fue mal.
7. Ahora hay varias acciones posibles:

- El emisor no tiene más mensajes que enviar; cerrará la conexión con un comando QUIT, que será respondido con "221 Service closing transmission channel".
- El emisor no tiene más mensajes que enviar, pero está preparado para recibir mensajes(si los hay) del otro extremo. Mandará el comando TURN. Los dos SMTPs intercambian sus papelees y el emisor que era antes receptor puede enviar ahora mensajes empezando por el paso 3 de arriba.
- El emisor tiene otro mensaje que enviar, y simplemente vuelve al paso 3 para enviar un nuevo MAIL.

Comandos SMTP: cliente

Comando	**Descripción**
HELO	• Para identificarse el origen al destino. • HELO <domain> <CRLF> Identifica el remitente al destinatario.
MAIL FROM	• Para comenzar la transacción de un mail. • MAIL FROM:<reverse-path> <CRLF> Identifica una transacción de correo e identifica al emisor.
RCPT TO	Se utiliza para **identificar un destinatario individual**. Si se necesita identificar múltiples destinatarios es necesario repetir el comando.
DATA	Permite enviar una serie de líneas de texto. El tamaño máximo de una línea es de 1.000 caracteres. Cada línea va seguida de un retorno de carro y avance de línea <CR><LF>. **La última línea debe llevar únicamente el carácter punto "."** seguido de <CR><LF>.
RSET	Aborta la transacción de correo actual.
NOOP	No operación. **Indica al extremo que envíe una respuesta positiva**. **Keepalives.** El destino responde

	"200: OK".
QUIT	Pide al otro extremo que envíe una respuesta positiva y cierre la conexión.
VRFY	• Pregunta al destino si existe usuario <string> • VRFY <string> <CRLF> Pide al receptor que confirme que un nombre identifica a un destinatario valido.
EXPN	Pide al receptor la **confirmación de una lista de correo** y que devuelva los nombres de los usuarios de dicha lista. • EXPN <string> <CRLF>
HELP	Pide al otro extremo información sobre los comandos disponibles. • HELP [<string>] <CRLF>
TURN	El emisor pide que se **inviertan los papeles**, para poder actuar como receptor. El receptor puede negarse a dicha petición.
SOML	Si el destinatario está conectado, entrega el mensaje directamente al terminal, en caso contrario lo entrega como correo convencional.
SAML	Entrega del mensaje en el buzón del destinatario. En caso de estar conectado también lo hace al terminal.
SEND	Si el destinatario está conectado, entrega el mensaje directamente al terminal.

Los 7 primeros (HELO a QUIT) son obligatorios y son aceptados por todos los agentes SMTP. Este protocolo se puede llevar a cabo utilizando una conexión telnet post.uv.es 25, y tecleando los comandos según el protocolo tal como se especifica a continuación.

SMTP: Comandos opcionales

Raramente implementados (se pueden usar en lugar de MAIL):

Send: Envía el mensaje al terminal

- SEND FROM:<reverse-path> <CRLF>

Send or mail: Si está conectado, al terminal; si no por mail

  - SOML FROM:<reverse-path> <CRLF>

Send and mail: Al terminal y por mail

  - SAML FROM:<reverse-path> <CRLF>

Códigos de respuesta SMTP: servidor

Código	**Descripción**
211	Estado del sistema.
214	Mensaje de ayuda.

220	**Servicio preparado.**
221	**Servicio cerrando el canal de transmisión.**
250	**Solicitud completada con éxito.**
251	Usuario no local, se enviará a <dirección de reenvío>
354	**Introduzca el texto, finalice con <CR><LF>.<CR><LF>.**
421	Servicio no disponible.
450	Solicitud de correo no ejecutada, servicio no disponible (buzón ocupado).
451	Acción no ejecutada, error local de procesamiento.
452	Acción no ejecutada, insuficiente espacio de almacenamiento en el sistema.
500	Error de sintaxis, comando no reconocido.
501	Error de sintaxis. P.ej **contestación de SMTP a ESMTP**
502	Comando no implementado.
503	Secuencia de comandos errónea.
504	Parámetro no implementado.
550	Solicitud no ejecutada, buzón no disponible.
551	Usuario no local, pruebe <dirección de reenvío>. **Si no se tiene cuenta**
552	Acción de correo solicitada abortada.
553	Solicitud no realizada (error de sintaxis).
554	Fallo en la transacción.

- La sintaxis de los comandos del **cliente** se especifica con **rigidez**.
- La sintaxis de las respuestas del **servidor es menos rígida**, sólo cuenta el código numérico, pudiendo cada implementación del protocolo SMTP poner la cadena de texto que desee después del código numérico

Inconvenientes

- Algunas implementaciones más viejas de SMTP no pueden manejar mensajes *mayores de 64 Kbytes*.
- Si el cliente y el servidor tienen *temporizaciones distintas*, uno de ellos puede terminar mientras que el otro continúa trabajando, terminando inesperadamente la conexión.
- En ocasiones pueden dispararse *tormentas de correo infinitas* cuando ambos servidores mutuamente tienen una lista que incluye a la otra lista del otro servidor.

- **Solución**, un nuevo protocolo extendido: **SMTP extendido (ESMTP)** en el RFC 1425. Los clientes que deseen usarlo deben enviar un mensaje **EHLO**, en lugar de HELO. Si el saludo se rechaza, **código 500**,

esto indica que el servidor es un servidor SMTP normal (basado en el RFC 821) y el cliente debe proceder de la manera normal.

SMTP: Ejemplo Transacción normal

- R: 220 BBN-UNIX.ARPA Simple Mail Transfer Service Ready
- S: HELO USC-ISIF.ARPA
- R: 250 BBN-UNIX.ARPA
- S: MAIL FROM:<Smith@USC-ISIF.ARPA>
- R: 250 OK
- S: RCPT TO:<Jones@BBN-UNIX.ARPA>
- R: 250 OK
- S: RCPT TO:<Green@BBN-UNIX.ARPA>
- R: 550 No such user here
- S: RCPT TO:<Brown@BBN-UNIX.ARPA>
- R: 250 OK
- S: DATA
- R: 354 Start mail input; end with <CRLF>.<CRLF>
- S: Blah blah blah...
- S: ...etc. etc. etc.
- S: .
- R: 250 OK
- S: QUIT
- R: 221 BBN-UNIX.ARPA Service closing transmission channel

SMTP y el DNS

Si la red usa el concepto de dominio, un SMTP no puede entregar simplemente correo a TEST.IBM.comando abriendo una conexión TCP con TEST.IBM.comando. Primero debe consultar al servidor de nombres para hallar a que host(en un nombre de dominio) debería entregar el mensaje.

Para la entrega de mensajes, el servidor de nombres almacena los RRs("Resource records") denominados MX RRs. Mapean un nombre de dominio a dos valores:

- Un valor de preferencia. Como pueden existir múltiples RRs MX para el mismo nombre de dominio, se les asigna una prioridad. El valor de prioridad más bajo corresponde al registro de mayor preferencia. Esto es útil siempre que el host de mayor preferencia sea inalcanzable; el emisor SMTP intenta conectar con el siguiente host en orden de prioridad.
- Un nombre de host.

También es posible que el servidor de nombres responda con una lista vacía de RRs MX. Esto significa que el nombre de dominio se halla bajo la autoridad del servidor, pero no tiene ningún MX asignado. En este caso, el emisor SMTP puede intentar establecer la conexión con el mismo nombre del host.

El RFC 974 da una recomendación importante. Recomienda que tras obtener los registros MX, el emisor SMTP debería consultar los registros WKS(*Well-Known Services*) del host, y chequear que el host referenciado tiene como entrada WKS a SMTP.

ESMTP

Aunque el protocolo SMTP esta bien definido, (por el RFC 821), pueden surgir algunos problemas. Uno se relaciona con la longitud del mensaje. Algunas implementaciones más viejas no pueden manejar mensajes mayores de 64KB. Otro problema se relaciona con las terminaciones de temporización. Hay más.

• SMTP RFC 821 es sencillo y está bien definido, pero tiene problemas:

- – Longitud de los mensajes

• Algunas implementaciones no soportan mensajes de más de 64 KB

- – Cliente y servidor pueden tener diferentes temporizadores (*timeout*)
- – Pueden originarse *mailstorms*

• Sobre todo con listas de correo

Para superar el problema se ha definido el SMTP extendido (ESMTP) en el RFC 1425

- – RFC 1425
- – Compatible hacia atrás
- – El cliente que desee utilizar las nuevas características comienza con un EHLO
- – servidor responde con los comandos ESMTP disponibles

MTAs DE REENVÍO

Conceptos previos:

- **DNS y registros MX**: son intercambiadores de correo, que reciben correo en nombre de otro servidor cuando el principal está fuera de servicio
- **Relay o reenvío:** indica si un servidor de correo, de transferencia de distribución, acepta correo de otro servidor para reenviar. *Ejemplo*: post.uv.es cuando manda un correo a un servidor de EEUU no lo hace directamente, si no lo va reenviando a través de servidores.
- **SPAM:** envío masivo a un conjunto de direcciones gestionadas por un servidor. El servidor puede configurarse para marcarlas como SPAM. La obtención de usuarios de un servidor se puede realizar utilizando los comandos SMTP “VRFY” y “EXPN”.

• Normalmente la conexión TCP no es directa entre MTA origen y destino, sino que se pasa por MTA intermedias.

• Dos tipos de MTA:

- – MTA local: configuradas para enviar todo mail no local a MTA relay
- – MTA relay: maneja todo el correo de entrada y salida de la organización

• Dos razones para esta división:

- – Simplifica la configuración de las MTAs
- – El MTA relay da una visión unitaria de la organización

• Típicamente entre dos UA se pasa por cuatro MTA

ENTREGA FINAL

• Puede haber ordenadores con UA pero sin MTA (típicamente: PCs)

• POP3 (Post Office Protocol)

- – RFC 1225
- – Protocolo ASCII (al estilo SMTP) que permite
  - • Ver qué mensajes hay en el buzón
  - • Traer mensajes del MTA al UA, para leerlos posteriormente.
  - • Borrar mensajes

• IMAP (Interactive Mail Access Protocol)

- – RFC 1064
- – Más sofisticado
- – deja el correo en el servidor
- – mejor cuando se accede desde varios ordenadores

• DSMP (Distributed Mail System Protocol)

- – RFC 1506
- – Puede traer el correo de varios buzones en varios MTAs

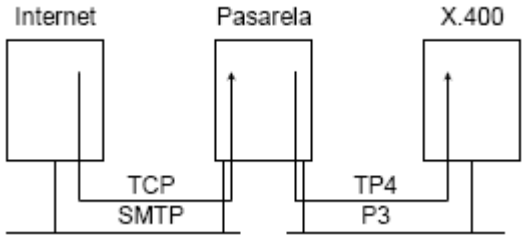
PASARELAS DE CORREO

• Hay otros sistemas de correo que no son el de Internet

- – X.400
- – cc:Mail
- – BBSs
- – …

• Las pasarelas de correo permiten intercambiar mensajes entre los diferentes sistemas.

• Tienen muchas limitaciones



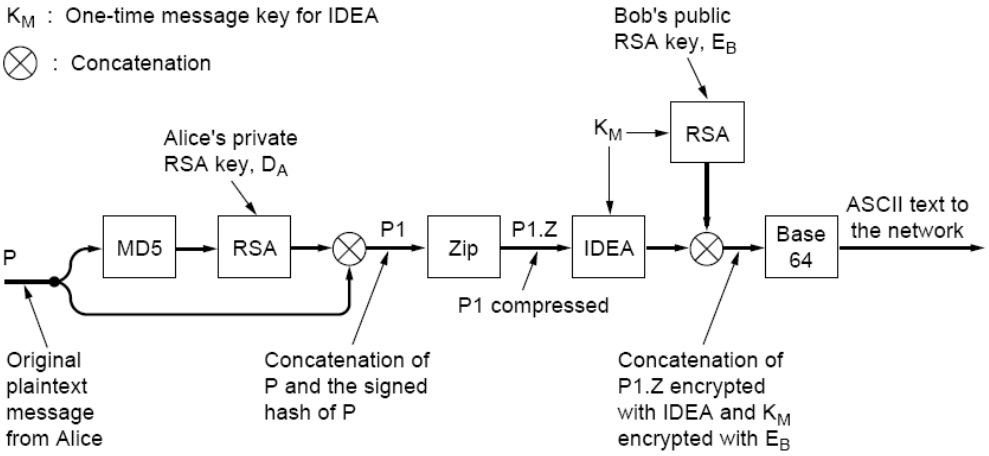
INTIMIDAD

El SMTP envía los mensajes sin cifrar por lo que cualquier sistema intermedio puede leerlos o copiarlos. Si se quiere que sólo el remitente y el destinatario puedan entenderlo hay que recurrir a técnicas de cifrado:

- PGP (Pretty Good Privacy)

CIFRADO CON PGP

- PGP (Pretty Good Privacy)
- – Obra de un hombre: Phil Zimmermann
- – gratuito, múltiples plataformas, muy extendido
- – Usa RSA, IDEA y MD5
- – Secreto y autentificación de mail RFC822



- – Clave RSA de 384, 512 ó 1024 bits
- – Mensaje PGP: