

Redes y Comunicaciones de Datos II

Cristian Escudero

Resumen 2

June 17, 2018

1 La Capa de Red

1.1 NAT y PAT

1.1.1 NAT - Traducción de Dirección de Red (*Network Address Translation*)

Las direcciones IP son escasas. Si tiene más clientes que IPs, tiene un problema. Para clientes propios con las conexiones de línea conmutada, una manera de resolver el problema es asignar dinámicamente una dirección IP a una computadora cuando ésta llama e inicia la sesión, y tomar de vuelta la dirección IP cuando se termina la sesión y reasignarla a otra visita.

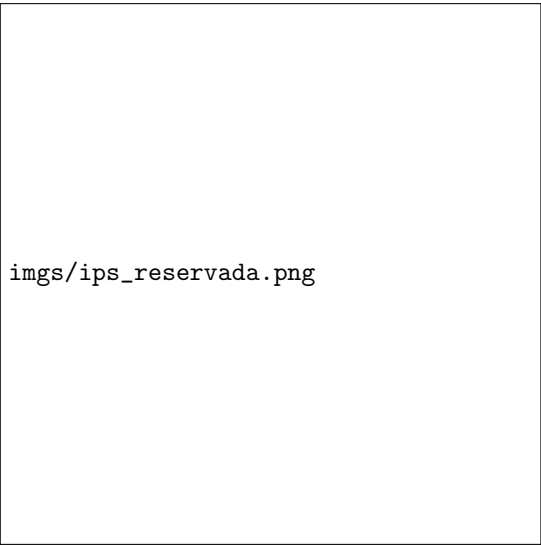
Esta estrategia falla para ISPs que sirven a una gran cantidad de usuarios, y más aún, si estos están conectados permanentemente a la red. La solución temporal que se designó hasta la transición a IPv6 fue la **Traducción de Dirección de Red (NAT)**.

La idea básica de NAT es asignar un rango de direcciones IP públicas pequeño a cada compañía para el tráfico de Internet. **Dentro** de la compañía, cada computadora tiene una dirección IP única que se usa para enrutar el tráfico interno. Sin embargo, cuando un paquete sale de la compañía y va al ISP, se presenta una **traducción de dirección**. Para hacer posible este esquema existen tres rangos de direcciones IP que se han declarado como privados, que las compañías pueden usar internamente cuando lo deseen. La única regla es que ningún paquete que contiene estas direcciones puede aparecer en la propia Internet. De esta manera simultáneamente sólo pueden salir a Internet con una dirección IP pública tantos equipos como direcciones públicas se hayan contratado.

Dentro de las instalaciones de la compañía, cada máquina tiene una dirección única privada. Sin embargo, cuando un paquete sale de las instalaciones de la compañía, pasa a través de una caja NAT que convierte la dirección interna de origen de IP, a una dirección IP pública de la compañía. A menudo, la caja NAT se combina en un solo dispositivo con un **firewall** (servidor de seguridad) que proporciona seguridad controlando cuidadosamente lo que entra y sale de la compañía.

En NAT, el enrutador sigue la pista de los datos básicos de cada conexión activa. Cuando una respuesta llega al enrutador, utiliza los datos de seguimiento de la conexión almacenados en la fase de salida para determinar la dirección privada de la red interna a la que remitir la respuesta.

¿Qué se hace cuándo se posee un rango muy pequeño o único de IPs públicas? El problema realmente surge cuando dos o más *hosts*



imgs/ips_reservada.png

Figure 1: Los tres rangos de IPs reservadas.

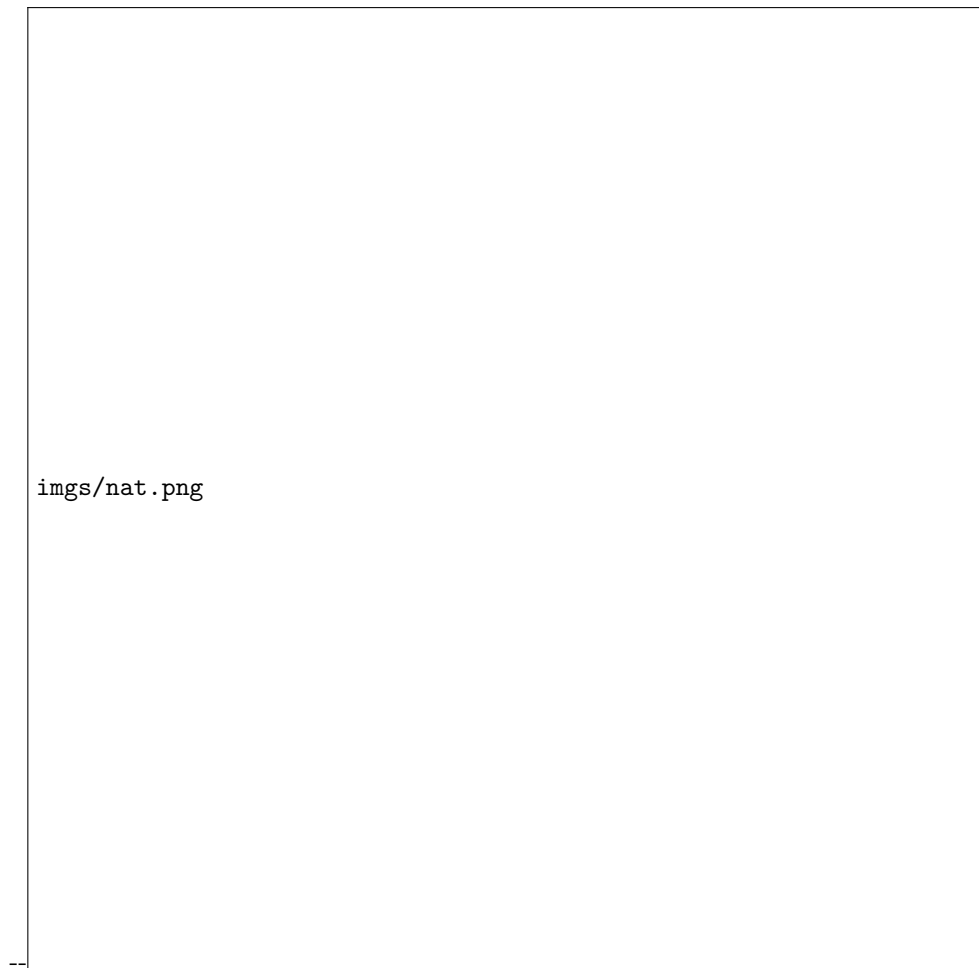


Figure 2: Colocación y funcionamiento de una caja NAT.

se conectan a una misma IP. Para evitar la ambigüedad en el retorno, se introdució la **Traducción de Direcciones por Puerto (PAT)**.

Siempre que un paquete **saliente** entra en la caja NAT (con PAT), además de modificarse la *dirección de origen*, se modifica el campo *Puerto de origen* TCP por un índice en la **tabla de traducción** de la caja NAT que almacena ambos valores.

Cuando un paquete **llega** a la caja NAT desde el ISP, el *Puerto de origen* en el encabezado TCP del paquete se extrae y se obtiene la dirección de origen dentro de la red utilizando el respectivo índice de la tabla de traducción de la caja NAT.

¿Por qué usa un número de índice en una tabla en vez de directamente el valor del puerto origen? Para evitar ambigüedades entre múltiples *hosts* que usan el mismo valor de puerto.

El esquema de PAT puede ser utilizado directamente por los ISP para aliviar el problema de la escasez de IPs, considerando a todos sus usuarios como una compañía grande, y brindándoles direcciones privadas.

2 La Capa de Aplicación

Las capas por debajo de la de **aplicación** están ahí para proporcionar **transporte confiable**, pero no hacen ningún trabajo verdadero para los usuarios. En la CdA se necesitan protocolos de apoyo que permitan el funcionamiento de las aplicaciones reales. Uno de ellos es el **DNS** (*Domain Name System*), que maneja la asignación de nombres dentro de Internet, facilitando el manejo de las direcciones de IP.

2.1 DNS - El Sistema de Nombres de Dominio

Dada a la dificultad de memorizar direcciones de red, y de la inflexibilidad de manejar cambios en las direcciones de los servidores al enviar mensajes a IPs, se introdujeron los nombres ASCII, con el fin de separar los **nombres de máquina** de las **direcciones de máquina**. Sin embargo, la red sólo comprende direcciones numéricas, por lo que se requieren algunos mecanismos para convertir las cadenas ASCII de nuevo a direcciones de red.

Una alternativa que surgió fue la de mantener un servidor con un archivo *hosts.txt* que listaba todos los *hosts* y sus direcciones de IP, y que se enviaba periódicamente a todos los *hosts* de la red.

Sin embargo, este método no sería práctico ante la presencia de miles de *hosts*, ya que no solo el tamaño del archivo crecería de manera considerable, sino que además ocurrirían conflictos constantes con los nombres de los *hosts* a menos de que dichos nombres se administraran **centralmente**, algo impensable en una red internacional enorme. Para resolver estos problemas, se inventó el **DNS**.

El DNS se basa en un esquema jerárquico que permite asignar nombres, basándose en el concepto de **dominio**, utilizando para su gestión una *base de datos* (*BD*) distribuida.

Para relacionar un nombre con una dirección IP, un programa de aplicación llama a un procedimiento de biblioteca llamado **resolvedor**, y le pasa el nombre como parámetro (**consulta DNS**). El resolvedor envía un paquete UDP a un servidor DNS local, que después busca el nombre y devuelve la dirección IP al resolvedor, que entonces lo remite al solicitante. Una vez que tiene la dirección IP, el programa puede establecer una conexión TCP con el destino, o enviarle paquetes UDP.

2.1.1 El espacio de nombres del DNS

Conceptualmente, Internet se divide en 200 **dominios** de nivel superior, cada uno de los cuales abarca muchos *hosts*. Cada dominio se divide en **subdominios**, los cuales, a su vez, también se dividen, y así sucesivamente. Puede verse como un árbol, cuyas hojas representan los dominios que no tienen subdominios, y que además pueden contener un solo *host*, o miles (si representa una organización).

Los dominios de nivel superior se dividen en dos categorías: **genéricos** y **de país** (o *geográficos*).


Cada dominio se nombra por la ruta hacia arriba desde él a la **raíz** (*sin nombre*). Los componentes se separan con puntos (ej, “*eng.sun.com.*”).

Los **nombres de dominio** pueden ser:

- **Absolutos:** terminan con punto (ej, “*eng.sun.com.*”).
- **Relativos:** tienen que interpretarse en algún contexto para determinar de manera única su significado verdadero (ej, “*eng.sun.com*”).

En ambos casos, un nombre de dominio hace referencia a un nodo específico del árbol y a todos los nodos por debajo de él, y no hacen distinción entre mayúsculas y minúsculas. Los nombres reflejan los **límites organizacionales**, no las redes físicas.

Para crear un nuevo dominio, se requiere el permiso del dominio en el que se incluirá, evitando así los conflictos de nombres y permitiendo a cada dominio llevar el registro de todos sus subdominios. Una vez que se ha creado y registrado un nuevo dominio, este puede crear subdominios, sin obtener el permiso de nadie más arriba en el árbol.



imgs/DNS_espacio_nombres.png

Figure 3: Parte del espacio de nombres de dominio de Internet.

2.1.2 Registros de recursos

Cada dominio, sea un *host* individual o un dominio de nivel superior, puede tener un grupo de **registros de recursos** (RRs) asociados a él. En un *host* individual, el RR más común es simplemente su dirección IP, pero también existen muchos otros tipos de RRs. Cuando un resolutor da un nombre de dominio al DNS, lo que recibe son los RRs asociados a ese nombre. Por lo tanto, DNS relaciona los dominios de nombres con los RRs.

Cada entrada en la tabla de un DNS contiene información, no sólo de las direcciones IP, si no de un RR, de cinco campos:

- **Nombre.dominio:** el dominio al que pertenece ese registro. Puede haber más de un registro por dominio. Si se omite, se usa por defecto el último nombre de dominio indicado.
- **TTL:** indica la estabilidad del registro.
 - + La información **altamente estable** tiene un valor grande (86.400 [seg], o sea, 1 día).
 - + La información **volátil** recibe un valor pequeño (60 [seg]).
- **Clase:** para la información de Internet, siempre es **IN**. Si se omite, se toma el último valor indicado.
- **Tipo:** indica el tipo de registro de que se trata.

- **Valor:** puede ser un número, un nombre de dominio o una cadena ASCII. La semántica depende del tipo de registro.

Principales tipos de RR DNS:

Tipo	Significado	Descripción
SOA	<i>Start of Authority</i>	Inicio de autoridad, identificando el dominio o la zona. Fija una serie de parámetros para esta zona.
A	<i>Address</i>	Dirección IP de un <i>host</i> en 32 bits. Si este tiene varias direcciones IP, habrá un registro diferente por cada una de ellas.
MX	<i>Mail eXchanger</i>	Especifica el nombre del dominio que está preparado para aceptar correo electrónico.
NS	<i>Name Server</i>	El nombre de dominio se hace corresponder con el nombre de una computadora de confianza para el dominio o servidor de nombres.
CNAME	<i>Canonical Name</i>	Es un alias que se corresponde con el nombre canónico verdadero.
PTR	<i>Pointer</i>	Apuntador, hace corresponder una dirección IP con el nombre de un sistema. Usado para asociar {dirección IP, nombre}, y realizar de esa forma búsquedas inversas .
HINFO	<i>Host Info</i>	Información del <i>host</i> , tipo y modelo de computadora y SO, en ASCII.
TXT	<i>Text</i>	Texto ASCII no interpretado. Permite agregar comentarios a la BD.
WKS	<i>Well-Known Services</i>	Servicios públicos. Puede listar los servicios de las aplicaciones disponibles en el ordenador.

2.1.3 Servidores de nombres

Un **solo** servidor de nombres podría contener toda la BD DNS y responder a todas las consultas dirigidas a ella, pero estaría tan sobrecargado que sería inservible. Más aún, si llegara a caerse, la Internet completa se vendría abajo.

Para evitar los problemas asociados a tener una sola fuente de información, el espacio de nombres DNS se divide en **zonas** no traslapantes. Una zona es una parte contigua del **árbol de nombres** que se administra como una unidad.

Cuando un resolutor tiene una consulta referente a un nombre de dominio, la pasa a uno de los servidores de nombres locales. Si el dominio que se busca cae bajo la jurisdicción del servidor de nombres, devuelve los registros de **recursos autorizados** -que provienen de la autoridad que administra el registro y, por lo tanto, siempre son correctos-. Los registros autorizados contrastan con los **registros en caché**, que podrían no estar actualizados.

Por otro lado, si el dominio es remoto y no hay información disponible localmente sobre el dominio solicitado, el servidor de nombres envía un mensaje de consulta al servidor de nombres de nivel superior en el que le solicita dicho dominio.

Este método de consultas conoce como **consulta recursiva**, puesto que cada servidor que no tiene toda la información solicitada la busca en algún otro lado y luego la proporciona. Algunos servidores no implementan este método y siempre devuelven el nombre del siguiente servidor a intentar (**consulta iterativa**).

Cuando un cliente DNS no recibe una respuesta antes de que termine su temporizador, por lo general probará con otro servidor la siguiente vez, asumiendo que el servidor probablemente esté inactivo.

Otro método es el de **búsqueda inversa**, que dado una dirección IP, devuelve el nombre. Para evitar una búsqueda exhaustiva por todo el espacio de nombres de dominio, se utiliza un dominio especial llamado **in-addr.arpa**. Cuando un cliente DNS desea conocer el nombre de dominio asociado a la dirección IP *w.x.y.z* realiza una pregunta inversa a **z.y.x.w.in-addr.arpa**.

Nota: la inversión de los **bytes** es necesaria debido a que los nombres de dominio son más genéricos por la derecha, al contrario que ocurre con las direcciones IP.

Tipos de servidores:

1. **Primarios (*Primary Name Servers*):** almacenan la información de su zona en una BD local. Son responsables de mantener la información actualizada y cualquier cambio debe ser notificado a este servidor.
2. **Secundarios (*Secondary Name Servers*):** obtienen los datos de su zona desde otro servidor que tenga autoridad para esa zona. El proceso de copia de la información se denomina **transferencia de zona**.
3. **Maestros (*Master Name Servers*):** transfieren las zonas a los servidores secundarios. Cuando un servidor secundario arranca busca un servidor maestro y realiza la *transferencia de zona*. Un servidor maestro para una zona puede ser a la vez un servidor primario o secundario de esa zona. Estos servidores extraen la información desde el servidor primario de la zona. Así se evita que los servidores secundarios sobrecargen al servidor primario con transferencias de zonas.
4. **Locales (*Caching-only servers*):** no tienen autoridad sobre ningún dominio; se limitan a contactar con otros servidores para resolver las peticiones de los clientes DNS. Estos servidores mantienen una memoria caché con las últimas preguntas contestadas. Cada vez que un cliente DNS le formula una pregunta, primero consulta en su memoria caché. Si encuentra la dirección IP solicitada, se la devuelve al cliente; si no, consulta a otros servidores, apuntando la respuesta en su memoria caché y comunicando la respuesta al cliente.
5. **Raíz (“.”):** se usa para consultar *hosts* externos, cuyas direcciones IP están presentes en un fichero de configuración del sistema y se cargan en el caché del DNS al iniciar el servidor. Proporcionan referencias directas a servidores de los dominios de segundo nivel (COM, EDU, geográficos, etc). Conocen a todos los servidores de dominios de primer nivel. Reciben consultas de servidores locales que no saben resolver un nombre. Hay trece servidores raíz a lo largo del mundo.

2.2 Correo Electrónico

Los primeros sistemas de correo electrónico simplemente consistían en protocolos de transferencia de archivos, conteniendo la primera línea del archivo la dirección del destinatario.

Algunas limitaciones que surgieron de este sistema:

- envió a grupos;
- sin notificación de entrega;
- sin estructura interna de envío.

En 1982 se publicaron las propuestas de correo electrónico del ARPANET:

- RFC 821. Protocolo de transmisión SMTP.
- RFC 822. Formato de mensaje.

El CCITT elaboró su recomendación X.400 para reemplazar al RFC 822, pero su excesiva complejidad lo hizo desaparecer, como le sucedió a la mayoría de las aplicaciones OSI.

2.2.1 Arquitectura y servicios

Los sistemas de correo electrónico normalmente consisten en dos subsistemas:

- Los **agentes de usuario**, que permiten leer y enviar correo electrónico. Son programas locales para interactuar con el sistema de correo electrónico. El usuario debe proporcionar el mensaje, la dirección de destino (en un formato que el agente de usuario pueda manejar, como direcciones DNS de la forma *example@dns-address*) y, posiblemente, algunos otros parámetros.

- Los **agentes de transferencia de mensajes**, que mueven los mensajes del origen al destino. Son por lo común *daemons* del sistema que operan en segundo plano y mueven correo electrónico a través del sistema. Se clasifican a su vez en:
 - + **De distribución.** (SMTP, ESMTP).
 - + **De entrega final.** (POP3, IMAP).

Por lo general, los sistemas de correo electrónico desempeñan cinco funciones básicas:

1. La **redacción** se refiere al proceso de crear mensajes y respuestas.
2. La **transferencia** se refiere a mover mensajes del remitente al destinatario. Requiere establecer una conexión con el destino o alguna máquina intermedia, enviar el mensaje y liberar la conexión, todo de forma automática.
3. La **generación del informe** notifica que ocurrió con el mensaje: *¿se entregó, rechazó o se perdió?*
4. La **visualización** de los mensajes es necesaria para que la gente pueda leer su correo electrónico.
5. La **disposición** tiene que ver con lo que el destinatario hace con el mensaje una vez que lo recibe.

En los sistemas de correo electrónico, existe una clara distinción entre el **sobre** (RFC 821) y su **contenido** (RFC 822). El sobre encapsula el mensaje; contiene toda la información necesaria para transportar el mensaje, como dirección de destino, prioridad y nivel de seguridad, la cual es diferente del mensaje mismo. Los agentes de transporte del mensaje usan el sobre para enrutar.

El mensaje dentro del sobre contiene dos partes: el **encabezado**, que contiene información de control para los agentes de usuario, y el **cuerpo**, que es por completo para el destinatario humano.

2.2.2 Formatos de mensaje

RFC 822:

Los mensajes consisten en un sobre primitivo, algunos campos de encabezado, una línea en blanco y el cuerpo del mensaje. Cada campo de cabecera consiste en una sola línea de texto ASCII que contiene el nombre del campo, dos puntos (:), y, para la mayoría de los campos, un valor.

El RFC 822 explícitamente indica que los usuarios pueden inventar cabeceras nuevas para uso privado siempre y cuando comiencen con la cadena X-.

MIME - Extensiones Multipropósito de Correo de Internet

La idea básica de MIME es continuar usando el formato RFC 822, pero agregar una estructura al cuerpo del mensaje y definir reglas de codificación para los mensajes no ASCII. De esa manera, nada cambia de la arquitectura anterior del RFC 822. Sólo afecta a los agentes de usuario, ya que para SMTP es totalmente transparente.

MIME define cinco nuevos encabezados de mensaje:

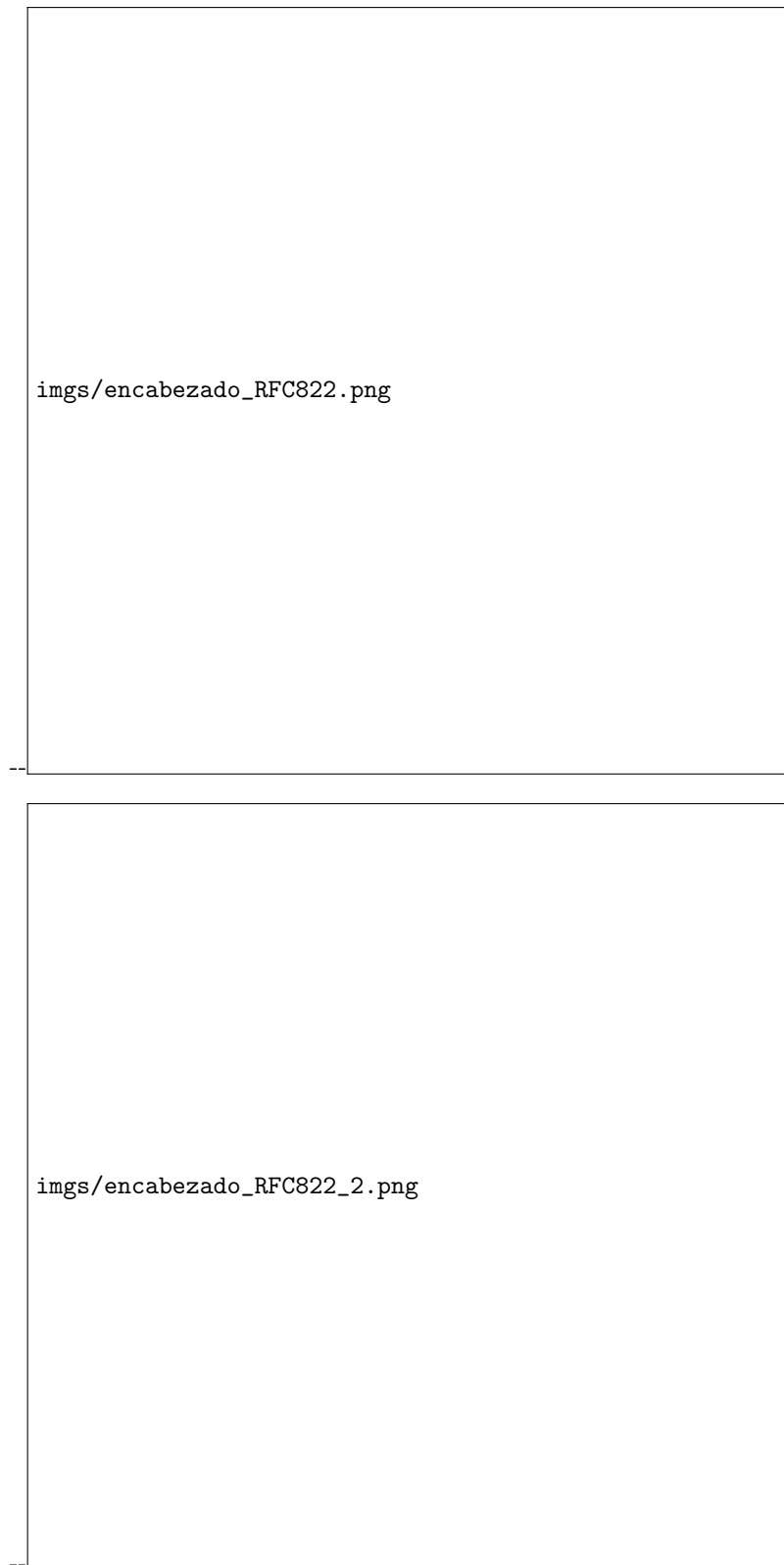


Figure 4: Campos de encabezado RFC 822: relacionados con el transporte de mensajes (*arriba*); usados por los agentes de usuario o destinatarios (*abajo*).

Encabezado	Significado
MIME-Version	Identifica la version de MIME. Si no existe se considera que el mensaje es texto normal en inglés.
Content-Description	Cadena de texto que describe el contenido. Esta cadena es necesaria para que el destinatario sepa si desea decodificar y leer el mensaje o no.
Content-Id	Identificador único, usa el mismo formato que el encabezado estándar Message-Id.
Content-Transfer-Encoding	Indica la manera en que está envuelto el cuerpo del mensaje. Existen

2.2.3 Transferencia de mensajes

SMTP - Protocolo Simple de Transporte de Correo

SMTP es un protocolo ASCII sencillo **cliente/servidor**. El correo electrónico se entrega al hacer que la máquina de origen establezca una conexión TCP con el **puerto 25** de la máquina de destino. Escuchando en este puerto está un *daemon* de correo electrónico que habla con el SMTP. Este *daemon* acepta conexiones de entrada y copia mensajes de estas a los buzones adecuados. Si no puede entregarse un mensaje, se devuelve al remitente un informe de error que contiene la primera parte del mensaje que no pudo entregarse.

Después de establecer la conexión TCP con el puerto 25, 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 está dispuesto**, el cliente libera la conexión y lo intenta después.
- Si **está dispuesto**, el cliente anuncia de quién proviene el mensaje, y a quién está dirigido. Si existe el destinatario en el destino, el servidor da al cliente permiso para enviar el mensaje. A continuación el cliente envía el mensaje y el servidor confirma su recepción. Una vez que todo el correo electrónico ha sido intercambiado en **ambas direcciones**, se libera la conexión.

Aunque el protocolo SMTP está bien definido, pueden surgir algunos problemas:

- La longitud del mensaje en implementaciones viejas de SMTP está limitada a 64 Kb.
- Si el cliente y el servidor tienen temporizaciones distintas, uno de ellos puede terminar mientras que el otro continúa trabajando, terminando inesperadamente la conexión.

Para superar algunos de estos problemas, se ha definido el **SMTP extendido (ESMTP)**.

2.2.4 Entrega final

Problema: acceso no permanente a internet, por ende, no se puede enviar correo directo entre *hosts*.

Solución: que un agente de transferencia de mensajes en una máquina ISP acepte correo electrónico para sus clientes y lo almacene en sus buzones en una máquina ISP con acceso permanente.

Un cliente puede consultar su buzón en la máquina ISP a través de los siguientes protocolos:

- **POP3 (Post Office Protocol):** 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. Su objetivo es obtener el correo electrónico del buzón remoto y almacenarlo en la máquina local del usuario para su lectura posterior. Usa el **puerto 110**.
- **IMAP (Interactive Mail Access Protocol):** la idea en que se basa 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 descarga el correo electrónico en la máquina personal del usuario dado que el usuario puede tener varias computadoras para consultar el correo. Además verifica si los correos ya han sido leídos con anterioridad. Usa el **puerto 143**.

2.3 World Wide Web

Acceso a documentos multimedia vinculados (**hipervínculos**) distribuidos en Internet. Posee un modelo **cliente/servidor**.

En el cliente:

- navegador (*browser*) para mostrar información recibida
- establece una conexión *telnet* al **puerto 80** del servidor y envía comandos para recuperar información.

En el servidor:

- proceso TCP de escucha en **puerto 80**.
- Una vez recibida una solicitud de un cliente, se envía respuesta y se libera la conexión.

Las **solicitudes** son en formato ASCII, y las **respuestas** son mensajes estructurados tipo MIME. Esto es según el protocolo **HTTP** (*HyperText Transfer Protocol*).

Cookies:

Una cookie es un pequeño archivo (o cadena, de a lo mucho 4 KB). Los navegadores almacenan *cookies* ofrecidas en un *directorio de cookies* en el disco duro de la máquina del cliente (a menos que el usuario las haya deshabilitado).

Una cookie puede contener hasta cinco campos:

- **Dominio:** indica de dónde viene la cookie.
- **Ruta:** es la ruta en la estructura del directorio del servidor que identifica qué partes del árbol de archivos del servidor podrían utilizar la cookie. Por lo general es (/), lo que significa el árbol completo.
- **Contenido:** toma la forma {nombre = valor}, que pueden ser los que el servidor desee.
- **Expira:** especifica cuándo caduca la cookie. Si este campo está ausente, el navegador descarta la cookie cuando sale (**cookie no persistente**). Si se proporciona una hora y una fecha, se mantiene hasta que expira (**cookie persistente**).
- **Seguro:** indica que el navegador podría simplemente regresar la cookie a un servidor seguro.

2.3.1 Protocolos FTP y Telnet

FTP (*File Transfer Protocol*):

Protocolo de transferencia de archivos, básico pero útil y fácil de usar. Disponible para muchos SO. Las funciones esenciales permiten:

- copiar archivos de un sistema a otro;
- ver listados de directorios;
- realizar tareas de gestión como cambiar de nombre o borrar archivos.

El **Trivial FTP (TFTP)** se utiliza en situaciones especiales como carga software (*SOs*, por ejemplo) en equipos sin disco duro (por ejemplo, *enrutadores*).

Telnet (*Terminal Networking*)

Protocolo de acceso a un ordenador desde otro ordenador sobre TCP. Disponible para muchos SO incluidos. Su acceso es a través del **puerto 23**. Aunque se ideó para comunicaciones *peer-to-peer* se usa para aplicaciones **cliente/servidor**. Por problemas de seguridad está siendo sustituido por el **SSH (Secure Shell)**.

2.3.2 Mejoras de desempeño

Almacenamiento en caché:

Una forma muy sencilla de mejorar el desempeño es guardar las páginas que han sido solicitadas en caso de que se utilicen nuevamente, es decir, **almacenarlas en caché**. Esta es una técnica en el **cliente**. El procedimiento común es que algún proceso, llamado **proxy**, mantenga el caché. Para utilizar el almacenamiento en caché, un navegador puede configurarse para que todas las solicitudes de páginas se le hagan a un proxy en lugar de al servidor real de la página. Si el proxy tiene la página, la regresa de inmediato. De lo contrario, la obtiene del servidor, la agrega al caché para uso posterior y la envía al cliente que la solicitó. Tiene que tener además una forma de determinar la obsolescencia de la página, y descartarla del caché luego de cierto tiempo (que dependerá de la estabilidad de la página).

Replicación del servidor:

Es una técnica en el **servidor**. El método más común que los servidores utilizan para mejorar el desempeño es replicar su contenido en múltiples ubicaciones separadas considerablemente (**espejeo** ó *mirroring*). De esa forma, se distribuye la carga en distintos servidores.