9. Capítulo 9: Servicios TCP/IP

A lo largo de los capítulos precedentes hemos desarrollados los fundamentos que aseguran la conectividad de todas las redes del mundo, y principalmente la "Red de Redes": Internet. Ahora bien, el crecimiento extraordinario que ha experimentado sobre todo Internet se debe a los servicios implementados en la capa de aplicación. Esas aplicaciones utilizan protocolos que permiten enviar correos electrónicos, acceder a páginas web, resolver los amigables nombres de dominio a direcciones ip, obtener direcciones ip en la nube en forma rápida, transparente y automática, entre muchos de los protocolos que en la actualidad forman parte de la gran red. Es momento de describir el funcionamiento de cuatro protocolos de los más importantes.

9.1. Protocolo de Configuración Dinámica de Host (DHCP, Dynamic Host Configuration Protocol)

El protocolo de configuración dinámica de host (DHCP, Dynamic Host Configuration Protocol) es una extensión de protocolo BOOTP que da más flexibilidad al administrar las direcciones IP. Este protocolo puede usarse para configurar dinámicamente los parámetros esenciales TCP/IP de los hosts (estaciones de trabajo y servidores) de una red. El protocolo DHCP tiene dos elementos:

- ✓ Un mecanismo para asignar direcciones IP y otros parámetros TCP/IP.
- ✓ Un protocolo para negociar y transmitir información específica del host.

El host TCP/IP que solicita la información de configuración TCP/IP se denomina cliente DHCP y el host que provee dicha información se llama servidor DHCP. El DHCP se describe en la norma RFC 2131.

9.1.1. Administración de Direcciones IP con DHCP

El protocolo DHCP usa los siguientes 3 métodos para asignar las direcciones IP:

1. Asignación manual o estática: El administrador de red configura manualmente la dirección IP del cliente DHCP en el servidor DHCP y la relaciona con una dirección MAC (MACAddress). El DHCP se usa para entregar al cliente DHCP que posee la respectiva MACAddress, el valor de la dirección IP configurada manualmente.
2. Asignación automática: No se requiere asignar manualmente direcciones IP. El servidor DHCP asigna al cliente DHCP, en el primer contacto, una dirección IP permanente que no podrá reutilizar ningún otro cliente DHCP. Este tipo de

asignación en ocasiones no es implementado por algunos servidores DHCP. De todas formas, se puede lograr el efecto asignando un tiempo de alquiler muy alto.

3. Asignación dinámica: El DHCP asigna una dirección IP al cliente DHCP por un tiempo determinado. Después que expire este lapso, se revoca la dirección IP y el cliente DHCP tiene que devolverla. Si el cliente aún necesita una dirección IP para efectuar sus operaciones, deberá solicitarla nuevamente.

Este protocolo permite la reutilización automática de una dirección IP. Si un cliente DHCP ya no necesita una dirección IP, como en el caso de una computadora apagada, ésta libera su dirección y la entrega al servidor DHCP. Éste puede reasignar dicha dirección a otro cliente que la pida.

El método de asignación dinámica es muy útil para clientes DHCP que necesitan una dirección IP para una conexión temporal a la red. Por ejemplo, se considera una situación en que 300 usuarios tengan computadoras portátiles conectadas a una red y ésta les ha asignado direcciones clase C. Este tipo de dirección permite a la red tener hasta 253 nodos (255 – 2 direcciones especiales = 253). Debido a que las computadoras que se conectan a una red usando el TCP/IP requieren tener una dirección única IP, entonces las 300 computadoras no podrían operar simultáneamente. Sin embargo, si como máximo hay 200 conexiones simultáneas a la red se puede configurar una dirección de clase C, y se reutilizan las direcciones IP no usadas. Usando el DHCP, en su método de asignación dinámica de direcciones IP, es posible reutilizar direcciones IP.

Sin importar cuál método se elija, aún puede configurarse otros parámetros IP (máscara, default Gateway, DNS, entre otros) de una única vez desde un servidor central, en lugar de repetir la configuración TCP/IP para cada computadora.

9.1.2. El proceso DHCP de adquisición de direcciones IP

Una vez que un cliente DHCP ha contactado con un servidor DHCP, a través de varios estados internos, negocia el uso y la duración de su dirección IP. La forma de adquisición de la dirección IP por el cliente DHCP se explica mejor en términos de un diagrama de procesos (Figura 9.1.2.1). Cuando se inicializa el cliente DHCP, éste comienza en el estado de inicialización INIT.

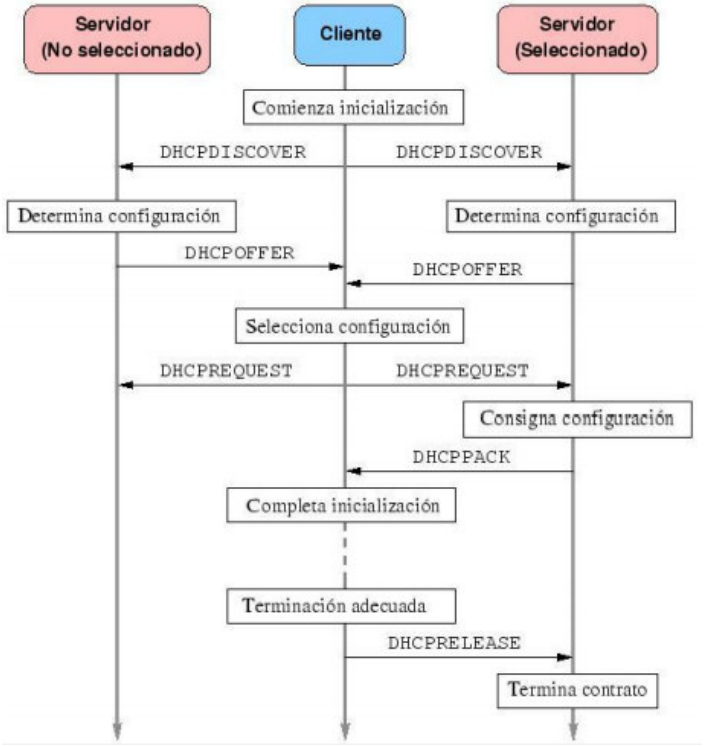


Figura 9.1.2.1: Proceso de adquisición de una dirección IP

El cliente DHCP desconoce sus parámetros IP y por eso envía un broadcast DHCPDISCOVER. El mensaje DHCPDISCOVER se encapsula en un paquete UDP. Se coloca el número 67 como puerto de destino UDP, el mismo utilizado por el servidor BOOTP, debido a que el protocolo DHCP es una extensión de este protocolo. El mensaje DHCPDISCOVER usa la dirección IP de broadcast de valor 255.255.255.255. Si no existe un servidor DHCP en la red local, el router IP debe tener un agente DHCP Relay o agente de reenvío que soporte la retransmisión de esta petición hacia las otras subredes. El agente DHCP relay se describe en la norma RFC 1542.

Antes de enviar el mensaje broadcast DHCPDISCOVER, el cliente DHCP espera por un tiempo aleatorio (entre 1 a 10 segundos) para evitar una colisión con otro cliente DHCP, para el caso que todos los clientes DHCP se inicialicen al mismo

Luego de enviar el mensaje broadcast DHCPDISCOVER, el cliente DHCP ingresa al estado SELECTING, donde recibe los mensajes DHCPOFFER de los servidores DHCP configurados para atenderlo. El tiempo que el cliente DHCP esperará por los mensajes DHCPOFFER depende de la implementación. Si el cliente DHCP recibe varias respuestas DHCPOFFER, elegirá una. En reacción, el cliente DHCP enviará un mensaje DHCPREQUEST para elegir un servidor DHCP, el que contestará con un DHCPACK.

Como opción, el cliente DHCP controla la dirección IP enviada en el DHCPACK para verificar si está o no está en uso. En una red con broadcast, el cliente

DHCP envía una petición ARP con la dirección IP sugerida para verificar que no esté duplicada. En caso de estarlo, el DHCPACK proveniente del servidor se ignora y se envía un DHCPDECLINE, con lo cual el cliente DHCP ingresa en estado INIT y vuelve a pedir una dirección IP válida que no esté en uso. Cuando la petición ARP se difunde sobre la red, el cliente usa su propia dirección de hardware en el campo de dirección fuente de hardware del ARP, pero coloca el valor de 0 en el campo de dirección fuente IP. Esta dirección de valor 0 se utiliza en lugar de la dirección IP sugerida, para no confundir a las memorias caché ARP de otros hosts.

Cuando se acepta el DHCPACK proveniente del servidor DHCP, se colocan tres valores de temporización y el cliente DHCP se mueve al estado BOUND (asociado).

- ✓ T1 es el temporizador de renovación de alquiler.
- ✓ T2 es el temporizador de reenganche.
- ✓ T3 es la duración del alquiler.

El DHCPACK siempre trae consigo el valor de T3 (configurable por el usuario). Los valores de T1 y T2 se configuran en el servidor DHCP; de no ser así, se usan los valores por defecto siguientes:

- ✓ T1 = 0,5 x T3.
- ✓ T2 = 0,875 x T3.

El tiempo real en que los temporizadores expiran se calcula añadiendo el valor del temporizador al tiempo en que se envió el mensaje DHCPREQUEST, el cual generó la respuesta DHCPACK.

Si este tiempo es T0, entonces los valores de expiración se calculan así:

- ✓ Expiración de T1 = T0 + T1
- ✓ Expiración de T2 = T0 + T2
- ✓ Expiración de T3 = T0 + T3

La RFC 2131 recomienda que se debe añadir un factor a T1 y T2 para evitar que varios clientes DHCP expiren sus temporizadores al mismo tiempo.

Después de la expiración del temporizador T1, el cliente DHCP se mueve del estado BOUND al estado RENEWING (renovación). En este último estado se debe negociar un nuevo alquiler para la dirección IP designada, entre el cliente DHCP y el servidor DHCP que originalmente le asignó la dirección IP. Si el servidor DHCP original, por algún motivo, no renueva el alquiler, le enviará un mensaje DHCPNACK y el cliente DHCP se moverá al estado INIT y intentará obtener una nueva dirección IP. En el caso contrario, si el servidor DHCP original envía un mensaje DHCPACK, éste contendrá la duración del nuevo alquiler. Entonces, el cliente DHCP coloca los valores de sus temporizadores y se moverá al estado BOUND.

Si el temporizador T2 (tiempo de reenganche) expira mientras el cliente DHCP está esperando en el estado RENEWING una respuesta sea DHCPACK o

DHCPNACK proveniente del servidor DHCP original, el cliente DHCP se moverá al estado REBINDING. El servidor original DHCP podría no haber respondido porque estaría apagado o porque el enlace con la red habría caído. Nótese en las ecuaciones previas que T2 es mayor que T1, de modo que el cliente DHCP espera que el servidor original DHCP renueve el alquiler por un tiempo igual a T2 – T1.

Al expirar el temporizador T2 (tiempo de reenganche), el cliente DHCP enviará un DHCPREQUEST a la red para contactar con cualquier servidor DHCP para extender el alquiler, con lo cual pasará al estado REBINDING.

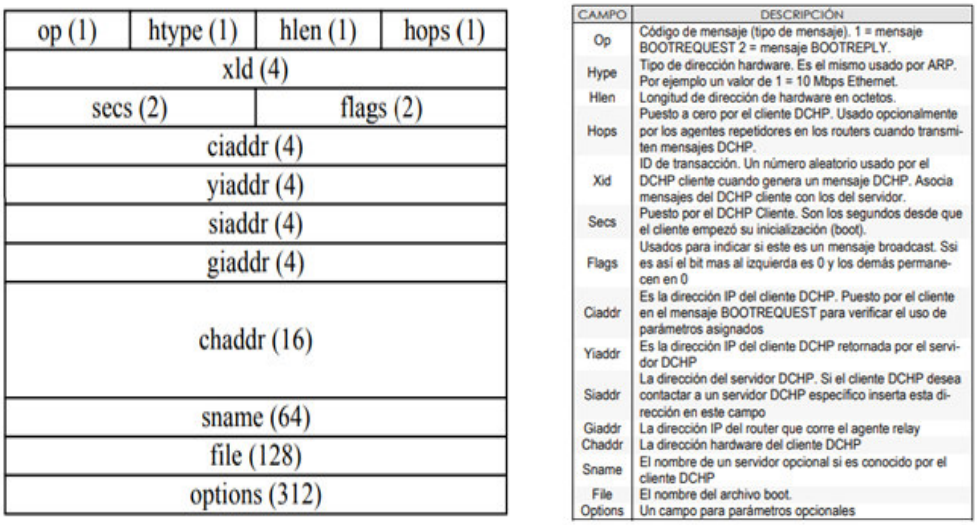
El cliente DHCP envía este mensaje broadcast DHCPREQUEST porque presume que, luego de haber esperado T2 – T1 segundos en el estado RENEWING, el servidor DHCP original no está disponible, por lo cual tratará de contactar con otro servidor DHCP para que le responda.

Si un servidor DHCP responde con un DHCPACK, el cliente DHCP renueva su alquiler (T3), coloca los temporizadores T1 y T2 y retorna al estado BOUND. Si no hay servidor DHCP disponible para renovar alquiler luego de expirar el temporizador T3, el alquiler cesa y el cliente DHCP pasa al estado INIT. Nótese que el cliente DHCP intentó renovar el alquiler primero con el servidor original y luego con cualquier otro servidor en la red. Al acabar el alquiler (T3 expira), el cliente DHCP debe devolver su dirección IP y cesar toda acción con dicha dirección IP en la red.

El cliente DHCP no siempre tiene que esperar la expiración del alquiler para terminar el uso de una dirección IP. Éste puede renunciar voluntariamente a una dirección IP, cancelando su alquiler. Por ejemplo, el usuario de un computador portátil podría conectarse a la red para una actividad particular. El servidor DHCP de la red podría colocar la dirección del alquiler por una hora. Suponiendo que el usuario acabe su tarea en 30 minutos, entonces se desconectará de la red al cabo de dicho lapso. Cuando el usuario se libera armoniosamente, el cliente DHCP enviará un mensaje DHCPRELEASE al servidor DHCP para cancelar el alquiler. La dirección IP ahora estará disponible.

9.1.3. Formato de un paquete DHCP

La Figura 9.1.3.1 a la izquierda ilustra el formato del paquete DHCP, el cual es un formato fijo para todos los campos, excepto para las opciones que tienen un mínimo de 312 octetos. A la derecha de la misma se proporciona una explicación de los campos del protocolo DHCP.



op (1)	htype (1)	hlen (1)	hops (1)
xld (4)			
secs (2)		flags (2)	
ciaddr (4)			
yiaddr (4)			
siaddr (4)			
giaddr (4)			
chaddr (16)			
sname (64)			
file (128)			
options (312)			

CAMPO	DESCRIPCIÓN
Op	Código de mensaje (tipo de mensaje). 1 = mensaje BOOTREQUEST 2 = mensaje BOOTREPLY.
Hype	Tipo de dirección hardware. Es el mismo usado por ARP. Por ejemplo un valor de 1 = 10 Mbps Ethernet.
Hlen	Longitud de dirección de hardware en octetos.
Hops	Puesto a cero por el cliente DCHP. Usado opcionalmente por los agentes repetidores en los routers cuando transmiten mensajes DCHP.
Xid	ID de transacción. Un número aleatorio usado por el DCHP cliente cuando genera un mensaje DCHP. Asocia mensajes del DCHP cliente con los del servidor.
Secs	Puesto por el DCHP Cliente. Son los segundos desde que el cliente empezó su inicialización (boot).
Flags	Usados para indicar si este es un mensaje broadcast. Ssi es así el bit mas al izquierda es 0 y los demás permanecen en 0
Ciaddr	Es la dirección IP del cliente DCHP. Puesto por el cliente en el mensaje BOOTREQUEST para verificar el uso de parámetros asignados
Yiaddr	Es la dirección IP del cliente DCHP retornada por el servidor DCHP
Siaddr	La dirección del servidor DCHP. Si el cliente DCHP desea contactar a un servidor DCHP específico inserta esta dirección en este campo
Giaddr	La dirección IP del router que corre el agente relay
Chaddr	La dirección hardware del cliente DCHP
Sname	El nombre de un servidor opcional si es conocido por el cliente DCHP
File	El nombre del archivo boot.
Options	Un campo para parámetros opcionales

Figura 9.1.3.1: Formato de paquete DHCP y significado de sus campos

El campo de opciones es variable en longitud, con el tamaño mínimo extendido a 312 octetos, de tal manera que el tamaño mínimo de un mensaje DHCP es de 576 octetos, por lo cual es el tamaño mínimo de datagrama IP que un host debe aceptar. Si el cliente DHCP necesita usar tamaño de mensajes más grandes, éste puede negociar esto con la opción de tamaño máximo de mensaje DHCP (Maximum DHCP message size).

9.1.4. Planificación de una implementación DHCP

Toda implementación de un esquema de entrega de direcciones IP en forma automática necesita de la definición de ciertos aspectos que se describen a continuación:

- ✓ Equipamiento que no puede recibir direcciones IP en forma automática o que deben ser configurados con direcciones estáticas (servidores) y las direcciones IP que se utilizaran para cada uno de ellos. Esas direcciones IP deben ser excluidas del conjunto de direcciones IP que se entregarán en alquiler
- ✓ Identificación de los equipos que deben recibir siempre una misma dirección IP y su respectiva dirección MAC, por ejemplo, impresoras de alto volumen. Estas direcciones van a ser reservadas junto a la MACAddress de cada uno de los equipos que la van a alquilar
- ✓ Definición del conjunto de direcciones que se van alquilar junto con el tiempo de vida de alquiler de cada una de ellas. Esto es lo que se denomina *"ámbito de direcciones"*.

- ✓ Determinar la cantidad de servidores DHCP que se van a desplegar y cuidar que los ámbitos de direcciones definidos en cada uno de ellos no se superpongan, puesto que, de no ser así, se alquilar direcciones IP duplicadas. Los servidores DHCP no dialogan entre ellos, por lo tanto, no chequean la posibilidad de que tengan configuradas para alquilar direcciones IP iguales, pero en distintos ámbitos.
- ✓ Definición de la cantidad de "ámbito de direcciones" basados en las distintas subredes en las cuales se debe entregar direcciones IP en forma automática.
- ✓ Configuración de parámetros complementarios a la dirección IP y máscara de subred que se pueden entregar en forma automática, por ejemplo, direcciones de *"default Gateway"*, *"servidor DNS"*, *"servidor WINS"*, entre otros.
- ✓ Identificación del tipo de routers que forman parte de la red. Existen routers que dejan pasar el tráfico DHCP (es un tráfico de broadcast) y son conocidos como routers compatibles con el RFC 1542.
- ✓ Configuración de Agentes de reenvío (DHCP Relay Agent). Los routers que no son compatibles RFC 1542 dividen las redes DHCP, lo que obliga a configurar un servidor DHCP por cada red. En caso de que existan segmentos de red con clientes DHCP, pero que no cuentan con un servidor de DHCP, es necesario que definan en ese segmento un *"Agente de Reenvío"*, el cual va a redirigir los pedidos de direcciones IP de los clientes hacia un servidor DHCP que se encuentre en otro segmento de la red.

9.2. Servicio de Resolución de Nombres (DNS, Domain Name System)

El sistema de nombres de dominio (Domain Name System, DNS), es un sistema de nomenclatura jerárquica estandarizado en la RFC 1034 y 1035 que permite asociar información con nombres de dominio. Su uso principal permite asociar nombres a direcciones IP con el fin de que los mismos puedan ser localizados más fácilmente usando un lenguaje más elegible para los humanos. Debido a que las direcciones IP son difíciles de memorizar, esta técnica permite asignar a las direcciones nombres significativos. Por ejemplo, para poder acceder al sitio de la Facultad de Ciencias Exactas y Tecnología de la Universidad Nacional de Tucumán, debería conocerse la dirección IP que aloja dicho servicio, que en este caso es 200.45.169.78. El sistema DNS asigna el nombre www.facet.unt.edu.ar a dicho sitio y permite que se pueda realizar de modo sencillo la traducción de dicho nombre a la dirección IP correspondiente. Del mismo modo ocurre con cualquier otro sitio o recurso.

En los comienzos de InterNet se utilizaba una única tabla centralizada de traducción de nombres a direcciones. En el año 1970 ARPANET estaba formada por unos cientos de máquinas y un único archivo, HOSTS.TXT, que contenía toda la información que se necesitaba sobre esas máquinas. El centro de información de red del Departamento de Defensa de Estados Unidos, disponía de la versión maestra de la tabla y otros sistemas realizaban una copia regularmente. Con el paso del tiempo y a medida que los protocolos

TCP/IP se utilizaban más asiduamente, este método presento serios inconvenientes entre los que pueden destacarse los siguientes:

- ✓ El tráfico y la carga de red para la máquina que contenía la tabla que hacía posible el mapeo era excesivo.
- ✓ La consistencia del archivo era muy difícil de mantener.
- ✓ No se podía garantizar la no duplicidad de nombres, dado que mantener una administración central en una red Internacional era algo muy complicado.
- ✓ A medida que la red crecía, el tamaño del archivo también lo hacía.
- ✓ Si la máquina central salía de servicio, toda la red quedaba inutilizable.
- ✓ El método era claramente poco escalable.

En al año 1984 surgió un nuevo sistema de resolución de nombres llamado Domain Name System (DNS). Las premisas básicas del nuevo sistema fueron solucionar los principales inconvenientes que se venían acarreando con el sistema anterior.

9.2.1. Funcionamiento

DNS es un sistema de nomenclatura jerárquica estandarizado en la RFC 1034 y 1035 que permite asociar información con nombres de dominio. Su uso principal permite asociar nombres a direcciones IP con el fin de que los mismos puedan ser localizados más fácilmente usando un lenguaje más elegible para los humanos. Para dar soporte al servicio, se utiliza una base de datos jerárquica y distribuida que permite almacenar la información de resolución. Asimismo, el servicio permite por su naturaleza descentralizada que cada porción de la base de datos sea administrada por la entidad u organización a la que fue delegada, permitiendo de esta manera que cada administrador solo pueda manipular los datos de su propia base de datos.

Cuando una aplicación cliente necesita resolver un nombre, enviará el requerimiento de resolución a algún servidor de nombres, del cual esperará recibir como resultado la dirección IP asociada a dicho nombre. Una vez obtenida la dirección IP, se procederá a la comunicación estándar que defina cada protocolo dependiendo de cada necesidad. Con el fin de brindar una alta disponibilidad al sistema se definieron 13 servidores DNS raíz, los cuáles se encuentran geográficamente dispersos en el mundo, y permiten que, ante una eventual falla en alguno de ellos, otros puedan tomar el control del sistema de resolución. La lista de los 13 servidores raíz para el sistema de nombres, puede verse en la Figura 9.2.1.1 (www.iana.org):

HOSTNAME	IP ADDRESSES	OPERATOR
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	Verisign, Inc.
b.root-servers.net	199.9.14.201, 2001:500:200::b	University of Southern California, Information Sciences Institute
c.root-servers.net	192.33.4.12, 2001:500:2::c	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10, 2001:500:a8::e	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4, 2001:500:12::d0d	US Department of Defense (NIC)
h.root-servers.net	198.97.190.53, 2001:500:1::53	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	Verisign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:9f::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project

Figura 9.2.1.1: Servidores Raíz (Org., 2021)

9.2.2. Espacio de nombres de dominio

La base de datos del sistema de nombres DNS se indexa utilizando nombres de dominios. Cada nombre de dominio está representado por un camino en un árbol invertido de nombres, al cual se lo conoce como espacio de nombres. Se denomina árbol invertido, ya que los nombres dentro del sistema se comienzan a leer desde las hojas (nodos inferiores) hacia la raíz. Dicho árbol posee una estructura jerárquica similar a la que se puede encontrar en un sistema de archivos de un sistema operativo. El árbol posee una única raíz denotada por el carácter "." (Punto). A partir de dicha raíz, la base de datos se estructura jerárquicamente como se muestra en la Figura 9.2.2.1. Como puede apreciarse, el primer nodo del árbol se corresponde con el nodo raíz. Es de allí desde donde parte la estructura de resolución. Los servidores responsables de la administración de dicha porción de la base de datos son los 13 servidores raíz, quienes a su vez delegan porciones de la base de datos a otros nodos:

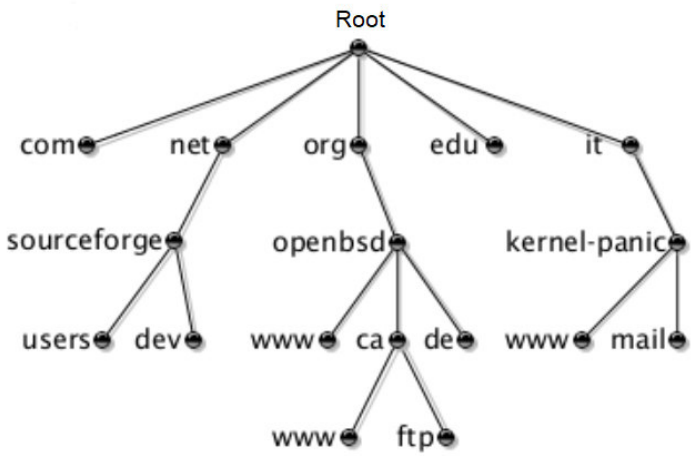


Figura 9.2.2.1: Estructura del espacio de nombres de dominio

Cada nodo en el árbol de la estructura posee una etiqueta que puede tener una longitud máxima de 63 caracteres. Los nodos intermedios generalmente denotan un nombre de dominio, mientras que los nodos inferiores (las hojas) hacen referencia a los recursos de resolución dentro del dominio. Un nombre completo de dominio está representado por el conjunto de etiquetas que van desde las hojas del árbol hasta la raíz del mismo, delimitadas por el carácter "." (Punto). Los nodos que dependen directamente de un padre (denominados también hermanos) no pueden contener en su etiqueta el mismo nombre. De este modo los caminos son unívocos. Si existe independencia entre los caminos, los nombres pueden repetirse.

Tal es el caso de la etiqueta www, que generalmente se repite como nombre de nodo en las hojas, pero sobre caminos distintos del árbol en la jerarquía de nombres. La Figura 9.2.2.2 grafica dicha situación. En la misma puede verse un caso de repetición de nombres sobre el mismo camino, lo cual no está permitido; y un caso de repetición del mismo nombre sobre un camino distinto. Este último solapamiento de nombres está permitido dado que se cumple la independencia de caminos:

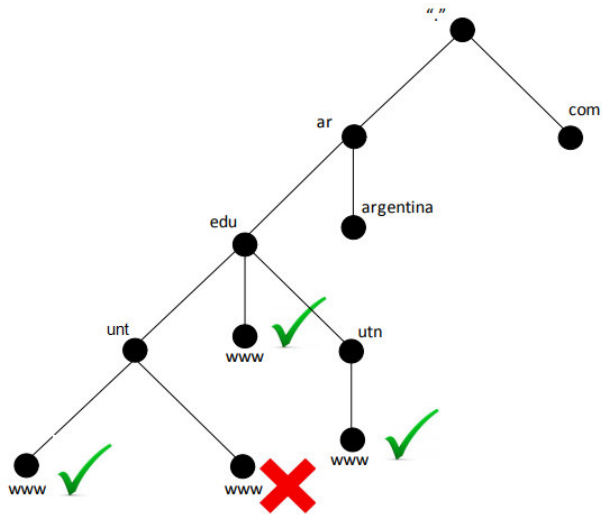


Figura 9.2.2.2: Control de inconsistencias en nombres DNS

Los nombres de dominio, incluyendo los nombres de servidores, se estructuran siguiendo una organización lógica a diferencia de la organización del direccionamiento IP que generalmente sigue una organización física o geográfica. Este tipo de organización permite que dos hosts dependientes del mismo dominio (por ejemplo, www1 y www2 dentro del dominio nombre1.com.ar) puedan residir en redes distintas, inclusive geográficamente distantes una de otra. Adicionalmente, de cada dominio se pueden desprender otros dominios, conocidos como sub dominios. Esta estructuración define la forma de árbol del sistema de nombres. Una forma simple de determinar si un dominio forma parte de otro, es a través de la estructura jerárquica de nombres. Si en el recorrido hacia la raíz se encuentra algún punto en común, entonces se

puede inferir que ambos subdominios forman parte de un dominio general. Por ejemplo, el dominio nombre1.com.ar es un subdominio del dominio com.ar.

A cada dominio, también se lo puede conocer como un subdominio de nivel N, donde N determina la distancia medida en cantidad de nodos del árbol que se deben atravesar hasta llegar a la raíz a partir de un nodo dado. Por consiguiente, se determina que la raíz "." (Punto) posee nivel 0. De la raíz se desprenden los dominios de nivel 1, que son un número acotado. Este tipo de dominio generalmente se utiliza para delegar administración de la base de datos de resolución a países, organizaciones, y otros fines que engloben lógicamente nombres relacionados a algún tipo de actividad. Si bien el sistema de nombres no impone reglas acerca del modo en el que deben llamarse los dominios, existen ciertos dominios de nivel 1 que se han tomado como referencia para la construcción del árbol (también se conocen como TLD, Top Level Domain) . Algunos de ellos son:

- ✓ com.: Agrupa organizaciones comerciales como por ejemplo google.com, hotmail.com y otras
- ✓ edu.: Agrupa organizaciones educativas como por ejemplo berkeley.edu, purdue.edu y otras.
- ✓ gov.: Agrupa organizaciones gubernamentales como por ejemplo nasa.gov y nsf.gov entre otras.
- ✓ net.: Originalmente agrupaba organizaciones proveedoras de infraestructura de red, como NSFNET y UUNET. En el año 1996, se modificó el criterio para permitir que organizaciones comerciales también pudieran registrar subdominios de .net, de modo similar a lo que se hace con .com.

Otros dominios de nivel 1 son ar, pe, uy, entre otros; que agrupan a las organizaciones de cada país. Este tipo de dominios son delegados a las autoridades de registro de Argentina, Perú o Uruguay, por ejemplo, de modo tal que cada una agrupe los nombres de las solicitudes generadas ante cada entidad registrante. En Argentina, la entidad responsable de administrar el subdominio ar. es NIC Argentina, perteneciente a la Dirección Nacional de Registros de Dominios de Internet, quien a su vez subdivide la base

de datos de resolución en otros dominios. Dentro de los subdominios más comunes que se pueden registrar en Argentina y a través del sitio se encuentran:

- ✓ com.ar.: Este subdominio agrupa organizaciones comerciales o de índole genérico. De acuerdo a la normativa vigente, podrá registrar dominios en la zona ".com.ar" cualquier persona física o jurídica argentina o extranjera.
- ✓ gob.ar.: Subdominio que agrupa a organizaciones gubernamentales
- ✓ mil.ar.: Sólo podrán registrar dominios en la zona ".mil.ar" entidades pertenecientes a las Fuerzas Armadas de la República Argentina.
- ✓ net.ar.: Sólo podrán registrar dominios en la zona ".net.ar" las entidades argentinas o extranjeras que sean proveedoras de servicios de Internet y tengan

licencia de la Comisión Nacional de Comunicaciones para prestar servicios de valor agregado en la República Argentina.

- ✓ org.ar.: Sólo podrán registrar dominios en la zona ".org.ar" las entidades que sean organizaciones sin fines de lucro argentinas o extranjeras
- ✓ tur.ar.: Sólo podrán registrar dominios en la zona ".tur.ar" las empresas de viajes y turismo, agencias de turismo o agencias de pasajes que se encuentren habilitadas por el Ministerio de Turismo.
- ✓ edu.ar.: Este subdominio agrupa organizaciones educativas como por ejemplo unlp.edu.ar. Para el caso especial del subdominio edu.ar, la ARIU (Asociación de Redes de Interconexión Universitaria) es la entidad en la que NIC Argentina delegó la responsabilidad de la operación estable y confiable de la base de datos.

Existe también un dominio de nivel superior o de nivel 1 especial, denominado arpa. El mismo fue originalmente utilizado para realizar la transición de la tabla HOSTS.TXT utilizada en la red ARPAnet hacia el sistema de nombres DNS. De allí su nombre. Es utilizado con el fin de realizar resoluciones inversas, es decir, determinar que nombre tiene asociado una determinada dirección IP. Entonces por ejemplo la dirección IP 200.45.169.78 es mapeada al nombre 78.169.45.200.in-addr.arpa, el cual a su vez podría resolverse a un nombre como por ejemplo www.facet.unt.edu.ar.

La estructuración de nombres, pensando en una agrupación lógica y utilizando como criterios ubicaciones geográficas o tipos de organización permite inferir los nombres en base a su estructuración. Este esquema permite que, dado un nombre de dominio, se pueda inferir a qué tipo de equipo o servicio hace referencia el mismo. Por ejemplo, sobre el nombre de dominio www.facet.unt.edu.ar, se puede inferir de modo sencillo que se trata de un nombre que hace referencia a un sitio o servicio que tiene referencia con el país Argentina dado que se encuentra en el subdominio de nivel superior ar; que referencia a dicho país. Luego podemos determinar que se trata de una organización educativa, a través del subdominio .edu. Si bien el nombre *"unt"* no responde a ninguna de las reglas impuestas, se puede inferir que son las siglas de la Universidad Nacional de Tucumán, lo mismo pasa con la sigla *"facet"*, la cual, luego de una breve búsqueda de información se deduce que significa *"Facultad de Ciencias Exactas y Tecnología"*. Por último, la sigla *"www"* que generalmente se encuentra asociada al servicio World Wide Web. Utilizando el mismo esquema de razonamiento, se puede inferir la índole de casi cualquier nombre de DNS.

9.2.3. Delegación de zonas

Una de las premisas principales en la implementación del servicio DNS fue que el mismo pudiera operar de forma descentralizada, de modo tal que los distintos dominios puedan ser administrados independientemente e inclusive puedan ser servidos desde distintos puntos de la red. Este principio se logra a través del concepto denominado delegación. Cada dominio de DNS es subdividido en otros, a los cuáles se los conoce como subdominio. De este modo, el dominio raíz "." (Punto), es subdividido en los

dominios de nivel superior o nivel 1 (com, edu, ar entre otros). A estos últimos los llamamos subdominios de la raíz ".", los cuáles a su vez se subdividen en otros como por ejemplo edu.ar y com.ar. Cada uno de estos dominios posee una administración descentralizada y permite que sea administrado por diferentes organizaciones y alojado en distintos servidores.

Cada organización es responsable por la administración de la porción de la base de datos (subdominio) que le fue delegada. La responsabilidad en la administración incluye la definición de registros dentro de la zona, o bien la delegación de nuevos subdominios a otras organizaciones o administradores. Por ejemplo, los administradores del dominio edu.ar delegan la administración del subdominio unt.edu.ar a la Universidad Nacional de Tucumán, quien, a su vez a través de los responsables administradores, define por un lado, sus propios registros de resolución como por ejemplo *"www.unt.edu.ar"* y por otro lado, también delega porciones de la base de datos a las distintas Facultades, como por ejemplo la Facultad Ciencias Exactas y Tecnología, quién a su vez es responsable de responder por el subdominio facet.unt.edu.ar. El esquema de delegación puede apreciarse de modo gráfico en la siguiente Figura 9.2.3.1:

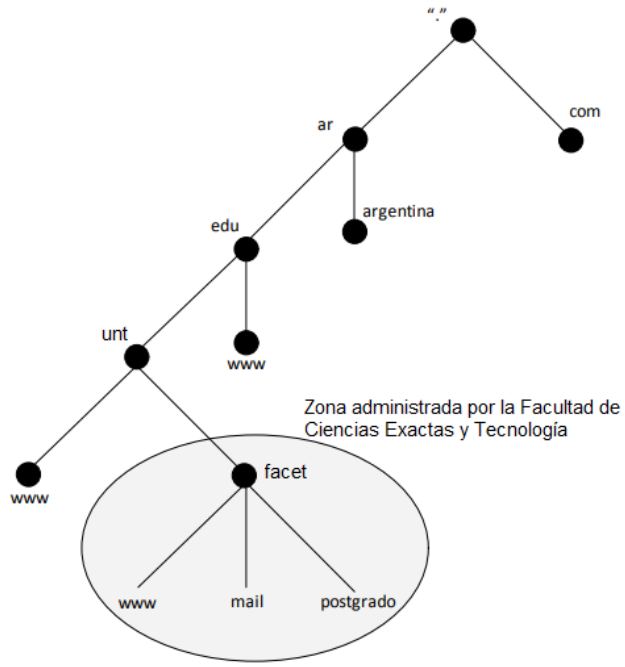


Figura 9.2.3.1: Delegación de zonas de administración

9.2.4. Servidores DNS y zonas

Toda la información acerca de zonas, nombres de dominio y registros de resolución, es gestionada por los servidores de nombre de dominio. Cada servidor de nombres contiene en la mayoría de los casos una porción de la base de datos de resolución conocida como zona de DNS, por la cual el mismo es responsable. Se dice que un servidor

es autoritativo para determinada zona, si el mismo gestiona la porción de la base de datos que se corresponde con el dominio representado por la zona. Un servidor DNS puede ser autoritativo para una o más zonas. Adicionalmente posee información que lo enlaza a otros servidores de resolución, lo cual le permite resolver otros nombres para los cuáles el mismo no es autoritativo o no posee información en su propia base de datos. Cada dominio de DNS puede ser subdividido en diversas zonas.

Cada una de las zonas se corresponde con un subdominio del dominio anterior. Adicionalmente, las zonas pueden ser gestionadas por el mismo servidor DNS, el cuál será autoritativo para las mismas, o bien pueden ser delegadas a nuevos servidores DNS, los cuales administrarán independientemente la porción de la base de resolución delegada y serán autoritativos para la resolución de nombres de dicha porción. La Figura 9.2.4.1 muestra un ejemplo acerca del esquema de dominios y delegación de zonas:

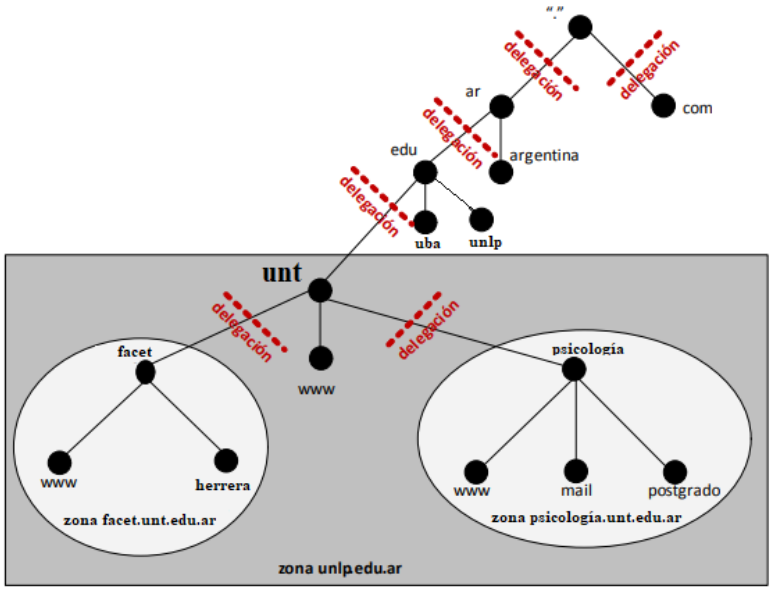


Figura 9.2.4.1: Zonas, dominios, delegación

La especificación de DNS, establece dos tipos de servidores de nombres: los servidores primarios y los secundarios. Un servidor DNS primario, posee la información de resolución de una determinada zona, ya que el administrador de dicho servicio ha ingresado la configuración manualmente en el mismo. Es el servidor donde se realizan los cambios para los distintos registros de resolución. El servidor secundario aprende la información de resolución a través del servidor primario. En este último, el administrador no actualiza información de resolución, sino que la configuración es aprendida del primario a través del concepto conocido como transferencia de zona. Cada vez que en el servidor primario se realice alguna modificación en la zona, la misma será notificada al servidor secundario para que el mismo actualice la información de resolución transfiriendo la porción de la base de datos que ha cambiado desde el servidor

primario. Adicionalmente, el servidor secundario consultará a intervalos regulares al servidor primario con el fin de determinar si hubo algún cambio en la zona para el cual no haya sido notificado. De producirse esta situación, el servidor secundario transferirá desde el primario la zona que haya sufrido modificaciones.

Pueden existir tantos servidores secundarios como se quiera configurar. El único requerimiento entre ellos es que exista comunicación a través del protocolo UDP y TCP en el puerto 53 de modo tal que puedan transferir información de actualización de zonas entre ellos. La posibilidad de agregar nuevos servidores secundarios y sincronizar la información entre ellos provee alta disponibilidad. Ante la caída de un servidor los otros responderán a los requerimientos. También provee escalabilidad ya que permite repartir la carga de resolución entre varios equipos. El concepto "servidor primario" o "servidor secundario" puede resultar confuso, ya que no se aplica a la totalidad del servidor DNS. Un servidor puede ser primario para un conjunto de zonas y secundario para otras, de modo tal que el concepto no aplica a la totalidad del servicio que corre sobre cierto equipo, sino que aplica a cada zona configurada. Tanto los servidores primarios como los secundarios son autoritativos para el conjunto de zonas que tengan configuradas.

La información de resolución dentro de un servidor, puede ser almacenada utilizando distintas técnicas, dependiendo del software que se utilice. La empresa Microsoft provee software para dar soporte al servicio de DNS, la cual almacena las configuraciones en el registro de Windows. Otra implementación del protocolo DNS es la provista por el software BIND (BIND, s.f.). En BIND, la información es almacenada en archivos dentro del sistema de archivos que corre el servicio. Cada archivo representa una zona de resolución, y los mismos tienen un esquema similar al que se muestra en la siguiente Figura 9.4.2.2:

```
ejemplo.com.ar.          IN SOA  dns1.ejemplo.com.ar. root.ejemplo.com.ar. (

199609260  ; serial

28800      ; refresh (8 hours)

7200       ; retry (2 hours)

2419200    ; expire (4 weeks)

86400      ; minimum (1 day)

)

IN       NS        dns1

@                              IN       MX        5 mail

dns1                           IN       A         192.168.1.1

mail                           IN       A         192.168.1.2

www                            IN       CNAME     mail
```

Figura 9.2.4.2: Registros definidos en el archivo de configuración de Zona

Los tipos más comunes de registro que pueden definirse son:

- ✓ SOA (Start Of Authority o Autoridad de la zona): Proporciona información sobre el servidor DNS primario de la zona, el correo electrónico del administrador del dominio, el número de serie del dominio, y los tiempos de refresco o actualización.
- ✓ NS (Name Server o Servidor de Nombres): Define la asociación que existe entre un nombre de dominio y los servidores de nombres que almacenan la información de dicho dominio. Cada dominio se puede asociar a uno o más servidores de nombres. Indica cuál o cuáles son los servidores de nombre autoritativos para la zona.
- ✓ MX (Mail Exchange o registro de intercambio de correo): Asocia un nombre de dominio a una lista de servidores de intercambio de correo para ese dominio. Permite realizar un balanceo de carga y definir prioridades para el uso de uno o más servicios de correo.
- ✓ A (Address o dirección): Este registro se usa para traducir nombres de servidores a direcciones IPv4. En IPv6 el registro es AAAA.
- ✓ CNAME (Canonical Name o Nombre canónico): Se usa para crear nombres de servidores o alias para los servidores de un dominio. Es usado cuando se están corriendo múltiples servicios (como por ejemplo mail y web) en un servidor con una sola dirección IP. Cada servicio tiene su propia entrada de DNS (como mail.ejemplo.com.ar y www.ejemplo.com.ar).
- ✓ PTR (Pointer o indicador): También conocido como registro reverso, funciona a la inversa del registro A, traduciendo direcciones IP en nombres de dominio

Existen otros tipos de registro definidos en la RFC, que permiten ampliar el esquema de resolución de nombres como por ejemplo SPF para indicar los hosts autorizados para enviar correo para un dominio determinado, o TXT que se utiliza para autenticación de correo. Adicionalmente se definen registros para el protocolo IPV6 como por ejemplo AAAA que tiene la misma función que el registro A de IPv4.

9.2.5. Proceso de resolución: consultas recursivas e iterativas

Cuando un servidor DNS recibe un requerimiento de resolución, verifica su propia base de datos con el fin de analizar si puede satisfacer el requerimiento con alguna de las zonas que en él se encuentran definidas. Si el requerimiento de resolución es sobre un registro de una zona para la cual el servidor no es autoritativo, entonces comenzará el proceso de resolución. Para ello, el servidor DNS realizará la consulta en nombre del cliente, hasta obtener el dato solicitado o bien un mensaje de error indicando que el nombre no puede ser resuelto. El servidor DNS realizará tantas consultas como sean necesarias a la base de datos jerárquica, con el fin de alcanzar la porción de la base que pueda devolver la solicitud completa. Suponiendo el nombre www.ejemplo.com.ar, la primera acción que se realizará será consultar al servidor raíz "." (Punto), la dirección

del servidor responsable de responder por el subdominio “ar”. Una vez obtenida dicha dirección, se consultará la dirección del servidor responsable por responder el subdominio “com.ar”. Por último, se consultará la dirección del servidor responsable del subdominio “ejemplo.com.ar”, al cual se le realizará una última consulta por el registro “www”. Este último servidor, conocido como autoritativo para el dominio, responderá al primer servidor DNS, quién se encargará de entregar la respuesta final al cliente. La respuesta obtenida, estará compuesta por el nombre resuelto, la dirección IP asociada a dicho nombre y el tiempo por el cual dicho nombre será válido previo a que deba realizarse una nueva consulta para validarlo. La Figura 9.2.5.1 muestra el proceso de resolución descripto:

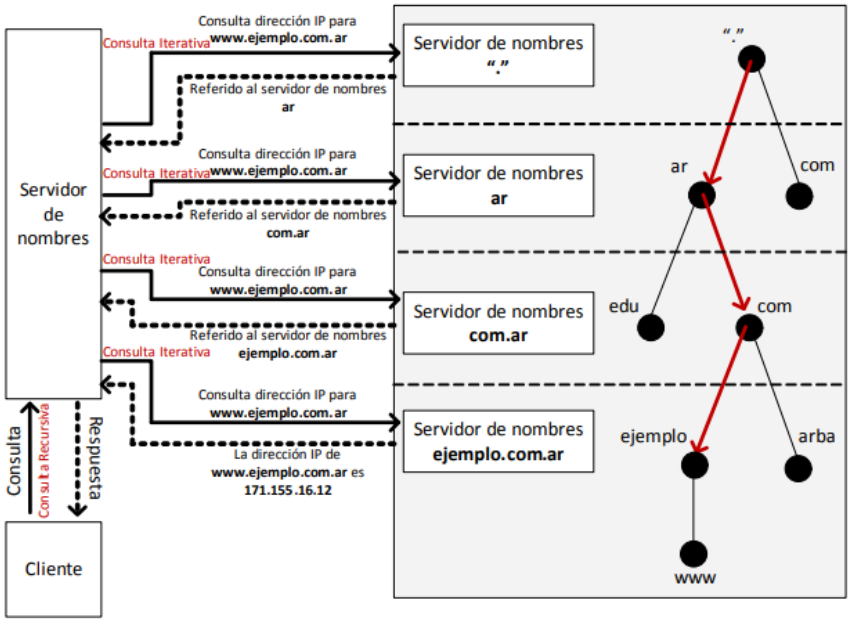


Figura 9.5.2.1: Proceso de resolución

Las consultas que un cliente realiza a un servidor DNS pueden ser recursivas o iterativas. Las consultas recursivas, o el proceso de resolución recursivo, es el nombre que se le da al proceso que realiza un servidor DNS cuando recibe un pedido de resolución. Se conoce como recursiva, ya que el servidor DNS realizará el mismo procedimiento, consultando a otros servidores DNS, hasta que el mismo obtenga la respuesta deseada y pueda ser pasada al cliente. El resultado de una consulta recursiva, es siempre el resultado final esperado (la asociación de la dirección IP al nombre), o bien un mensaje de error indicando que el registro consultado no existe. La resolución iterativa, sin embargo, hace referencia al proceso de resolución utilizado por el servidor de nombres cuando recibe una consulta iterativa. Cuando un servidor recibe una consulta iterativa, el mismo no está obligado a dar la respuesta final, sino que lo que puede hacer es referir al cliente a otro servidor DNS con el fin de que se realice una nueva consulta y determinar si el segundo puede dar la respuesta buscada. Si el segundo servidor referido no puede

dar la respuesta buscada, entonces devolverá el nombre de otro servidor para que el proceso de resolución continúe, hasta encontrar a algún servidor autoritativo que pueda dar una respuesta final por el registro solicitado. La Figura 9.5.2.1 muestra el proceso de resolución completo, diferenciando las consultas iterativas de las recursivas para la resolución del nombre www.ejemplo.com.ar.

9.2.6. Caching (Mantenimiento de los nombres resueltos en memoria temporal)

El proceso de resolución de nombres es un proceso complejo que en la mayoría de los casos implica realizar numerosas consultas a diversos servidores, con el fin de servir la consulta de un cliente. Cada una de las consultas intermedias, agrega retardos medidos en tiempo a la respuesta final. De este modo, si la consulta que realiza un cliente a su servidor DNS requiere tres consultas adicionales a otros servidores con el fin de resolver el nombre solicitado, el tiempo total de respuesta estará dado por la sumatoria de los tiempos de respuesta de los 3 servidores consultados. La técnica conocida como *"caching"* tiene como finalidad disminuir los tiempos anteriores a través del almacenamiento temporal de las consultas ya realizadas. De este modo si dos clientes consultan el mismo nombre a un servidor DNS en un lapso de tiempo acotado, el primer requerimiento de resolución deberá esperar hasta que se complete la totalidad del proceso. El segundo requerimiento será servido directamente por el servidor DNS sin necesidad de realizar nuevamente el procedimiento de búsqueda, ya que el mismo ha almacenado temporalmente en su base de datos (cache) el resultado del requerimiento previo.

Cada vez que un servidor DNS recibe un requerimiento de resolución recursiva, el mismo debe realizar diversas consultas a otros servidores con el fin de buscar una respuesta a la consulta original. Cada consulta que realiza, le permite descubrir información acerca del espacio de nombres en el árbol de resolución. Cada vez que el servidor es referido a otro servidor DNS, el mismo almacena la información acerca de los servidores autoritativos para cierta porción del espacio de nombres y sus correspondientes direcciones IP. La información almacenada le permitirá acelerar las búsquedas a futuro para nuevos requerimientos de resolución. El tiempo de validez que tendrá un recuso resuelto, dependerá del valor establecido en el campo TTL para la respuesta. Una vez expirado el tiempo de vida, el servidor deberá repetir el proceso ante un nuevo requerimiento de resolución. También son aprendidos y almacenados temporalmente por el servidor DNS, los requerimientos de resolución cuyo resultado han arrojado un error, ya sea porque el servidor DNS referido no se encuentra disponible o porque el registro solicitado no existe. A este último concepto se lo conoce como *"negative caching"*, y permite almacenar por lapsos de tiempo acotados este tipo de resoluciones con el fin de brindar más rápidamente los resultados.

El almacenamiento de las consultas realizadas por parte del servidor, implica que cada vez que se desee resolver un nombre, el servidor DNS buscará en su propia base de datos de resolución temporal (caché) si posee el resultado a la próxima consulta a realizar. Si el resultado se encuentra en la caché y es válido, entonces será servido inmediatamente, caso contrario el servidor realizará la consulta de modo habitual.

Esta técnica principalmente utilizada por los servidores DNS, ha sido extendida a los sistemas operativos y aplicaciones. Los sistemas operativos de los clientes que realizan consultas también almacenan las resoluciones realizadas, así como también lo hacen las aplicaciones. El objetivo es brindar una mejor experiencia al usuario disminuyendo los tiempos de respuesta en las consultas realizadas.

Las "caché" deben poseer un mecanismo de expiración que permita determinar cuándo un dato es válido o no. Este mecanismo permite mantener los datos temporales actualizados con el fin de mantener la coherencia en las respuestas. Para ello, se utiliza el valor TTL (Time To Live) de las respuestas, para almacenar los resultados temporalmente por el lapso de tiempo indicado en este valor. Cuando el administrador de una determinada zona configura el valor TTL para la misma o un registro particular, indica a otros servidores DNS cuál es el lapso de tiempo que los mismos tienen autorizado a almacenar temporalmente los recursos resueltos. Luego de este tiempo los datos almacenados ya no tendrán validez, debiendo descartarse y resolverse nuevamente. Un valor de TTL demasiado alto, causará que los recursos sean almacenados en caches intermedias por largos lapsos de tiempo, lo cual puede causar que una actualización de datos no se vea reflejada inmediatamente para los clientes. Adicionalmente, el hecho de permitir que los datos se cacheen temporalmente en otros servidores, causará que el servidor DNS autoritativo reciba una menor cantidad de consultas para determinada zona. En contrapartida, un valor de TTL bajo causará que los recursos se almacenen por cortos lapsos de tiempo asegurando un alto grado de actualización en las respuestas; pero causará que el servidor autoritativo tenga que contestar una mayor cantidad de consultas.

Si bien se determina que los registros resueltos deben ser almacenados o cacheados únicamente por el lapso de tiempo indicado en el campo TTL de la respuesta, algunos sistemas operativos e inclusive aplicaciones no respetan esta imposición. Esto se debe a que el proceso de resolución de nombres implica un retardo al momento de establecer una conexión. Previo a establecer la comunicación, debe realizarse la traducción del nombre dado a una dirección IP. En el ejemplo de acceso a un recurso en un servidor web, cuando el usuario ingresa una dirección en su navegador la primera acción que se realiza es la resolución del nombre solicitado. Esto disparará una serie de consultas a servidores DNS, de modo tal de obtener la dirección IP a la cual deberá realizarse la conexión utilizando el protocolo HTTP. Debido a que las resoluciones implican un retardo en la comunicación, los navegadores web implementan su propia caché de resolución, del mismo modo que lo hace el sistema operativo cliente y los servidores no autoritativos.

El principal problema de la implementación de caches de resolución por parte de las aplicaciones, reside en la necesidad de solicitar la resolución de los registros de DNS a través del sistema operativo en el que corren. Es el sistema operativo quien realizará la resolución en nombre de la aplicación y luego le devolverá el resultado a la misma. Con el fin de proveer un sistema de comunicación estándar entre el sistema operativo y las aplicaciones para la resolución de nombres, se utilizan las funciones estándar *"gethostbyname"* y *"gethostbyaddr"*. Las mismas permiten traducir nombres a direcciones IP y direcciones IP a nombres respectivamente. Estas funciones han sido utilizadas por años para realizar el proceso de resolución. En la actualidad, dichas

funciones han sido reemplazadas por las funciones *"getnameinfo"* y *"getaddrinfo"* respectivamente. Esto se debe a que las funciones anteriores no hacían un correcto manejo de la resolución sobre la pila de IP versión 6. La principal particularidad de las 4 funciones nombradas, es que, si bien se encargan de enmascarar el proceso de resolución de nombres, cuando devuelven el resultado a la aplicación no pasan el valor del campo TTL devuelto en la respuesta por parte del servidor DNS.

9.2.7. Actualización de Zonas

Los registros de recursos dentro de una zona determinada deben ser administrados. A menudo surgen cambios que implican modificar la base de datos de resolución, de modo tal que un nombre definido con ciertos valores comience a resolverse con otros. Tal es el caso del cambio de dirección IP de un servidor, el cual deberá reflejarse en el servicio DNS para que cuando los usuarios quieran acceder al mismo utilizando su nombre, el servidor DNS devuelva la nueva dirección IP. Adicionalmente, puede resultar necesario agregar nuevos registros de resolución en una determinada zona.

Los cambios en la base de datos de resolución pueden realizarse editando los archivos con un editor de texto, o bien a través de alguna interfaz definida, como por ejemplo NSUPDATE que permite la realización de actualizaciones dinámicas sobre una zona dada. La primera metodología, implica editar el archivo de zona con un editor de texto, modificar el registro deseado, incrementar el número de serie de la zona y salvar los cambios al archivo. Por último, debe indicarse al proceso BIND que relea nuevamente la configuración de la zona, y en caso de que la misma posea algún servidor secundario, notifique los cambios a dicho servidor. Es muy importante modificar el número de serie de la zona, por un número de serie superior al que se encuentra configurado al momento de realizar el cambio. Para ello, se utiliza la convención AAAAMMDDNN, donde AAAA es el año, MM representa al mes, DD representa al día y NN representa a un número auto incremental de 00 a 99. Si el número de serie no es actualizado, los servidores DNS no transferirán la zona modificada, e inclusive otros servidores DNS no serán notificados del cambio.

Otra opción para la realización de modificaciones en las zonas de DNS, es la utilización de la interfaz NSUPDATE. Se trata de una utilidad incluida a partir de la versión 8 del servidor BIND, que permite realizar modificaciones a los archivos de zona de manera sencilla. La utilidad permite agregar o quitar un registro de recurso de una zona de DNS, sin la necesidad de editar los archivos manualmente. A través de la ejecución del comando, se pueden enviar requerimientos de actualización dinámica de acuerdo al estándar definido en la RFC 2136. Cuando una zona es administrada a través de actualizaciones dinámicas, no debe ser editada su configuración manualmente.

9.2.8. Extensiones de Seguridad DNSSEC

Las Extensiones de Seguridad para el Sistema de Nombres de Dominio (DNSSEC) o Domain Name System Security Extensions son un conjunto de especificaciones de la IETF (Internet Engineering Task Force) que agregan funciones de seguridad al protocolo DNS. El hecho de que la funcionalidad se agregue al protocolo ya existente, permite que DNSSEC y DNS sean compatibles entre sí. Las extensiones, agregan las características de integridad y autenticidad a DNS, utilizando técnicas de criptografía de clave pública. De este modo es posible determinar que una traducción de nombre de dominio a IP o viceversa es legítima, está autorizada por la entidad que debe responder por la zona de resolución y no ha sido modificada en el camino. Las extensiones fueron creadas con el fin de proteger a los clientes de datos falsificados tales como los que se producen por envenenamientos de cache o suplantación de DNS.

DNSSEC utiliza para garantizar la legitimidad de las zonas, la misma jerarquía de delegación que utiliza el servidor DNS. Los servidores de nivel 1 firman digitalmente sus zonas y los subdominios que se generan sobre las mismas. De este modo cada responsable de un subdominio determinado podrá realizar la misma acción con los subdominios que de él dependan. Es importante destacar que las extensiones de seguridad están orientadas a brindar autenticidad de los datos y no cifrado o encripción de los mismos.

Otra funcionalidad de DNSSEC es la de brindar autenticación. Históricamente se ha utilizado el concepto de autenticar a los usuarios que pueden realizar modificaciones sobre una zona de DNS utilizando su dirección IP. Es de este modo, que generalmente para que un servidor primario pueda comunicar cambios de zonas a un secundario o viceversa, en la configuración para la definición de la zona de DNS se indica cuál es la dirección IP de la que se permite realizar modificaciones. En la Figura 9.2.8.1 muestra un ejemplo de definición de zona que permite que la misma se actualizada desde una determinada dirección IP:

```
zone "ejemplo.com.ar" in {
type master;
file "/etc/bind/zonas/ejemplo.com.ar.conf";
allow-update  { 192.168.1.254 };
};
```

Figura 9.2.8.1: Transferencia de zona sólo permitida desde una única IP

Utilizando la funcionalidad provista por las extensiones de seguridad, la definición anterior puede modificarse de modo tal que la actualización sea permitida a aquel que posea la llave criptográfica adecuada para realizar la actualización. La Figura 9.2.8.2 muestra el ejemplo de configuración:

```
key “ejemplo” {
algorithm HMAC-MD5;
secret “xqaD2Hg0DsDdDFfF9Gr8r3j4rGgG09Gerf2SgSe3erA4E62ee3q4Eq2dq3q4GfD9==”
}

zone "ejemplo.com.ar" in {
type master;
file "/etc/bind/zonas/ejemplo.com.ar.conf";
allow-update  { key ejemplo };
};
```

Figura 9.2.8.2: Actualización de zona basada en certificados digitales

9.2.9. Formato del mensaje DNS

Los mensajes DNS utilizan un formato único, que se muestra en la Figura 9.2.9.1. Hay cinco secciones posibles en un mensaje DNS: cabecera, pregunta, respuesta, autoridad y registros adicionales.

La sección de cabecera siempre está presente y consta de los siguientes campos:

- ✓ Query Response: Indica si este mensaje es una consulta o una respuesta.
- ✓ Opcode: Indica si se trata de una consulta estándar, una consulta inversa (dirección a nombre), o una solicitud de estado del servidor. Este valor lo establece el emisor y se copia en la respuesta.
- ✓ Authoritative Answer: Válido en una respuesta e indica si el servidor de nombres que responde es una autoridad para el nombre de dominio en cuestión.
- ✓ Truncated: Indica si el mensaje de respuesta fue truncado debido a longitud superior a la permitida en el canal de transmisión. Si es así, el solicitante utilizará una conexión TCP para reenviar la consulta.
- ✓ Recursion Desired: Si se establece, indica al servidor que persiga la consulta de forma recursiva.
- ✓ Recursion Available: Se establece o se desactiva en una respuesta para indicar si la consulta recursiva está disponible en el servidor de nombres.
- ✓ Response Code: Los valores posibles son: sin error, error de formato (el servidor no puede interpretar consulta), fallo del servidor, error de nombre (el nombre de dominio no existe), no implementado (no se admite este tipo de consulta), y rechazado (por razones de política).
- ✓ QDcount: Número de entradas en la sección de preguntas (cero o más).
- ✓ ANcount: Número de RR en la sección de respuestas (cero o más).
- ✓ NScount: Número de RR en la sección de autoridad (cero o más).
- ✓ ARcount: Número de RR en la sección de registros adicionales (cero o más).

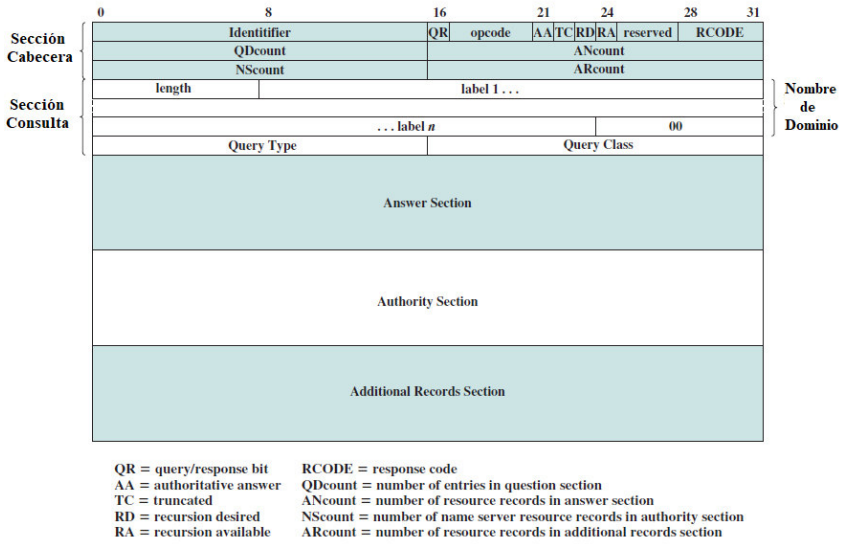


Figura 9.2.9.1: Mensaje DNS

La sección de preguntas contiene las consultas para el servidor de nombres. Si está presente, contiene normalmente una sola entrada. Cada entrada está compuesta de los siguientes campos:

- ✓ Domain Name: Un nombre de dominio representado como una secuencia de etiquetas, donde cada etiqueta consiste en un octeto de longitud seguido de ese número de octetos. El nombre de dominio nombre de dominio termina con el octeto de longitud cero de la etiqueta nula de la raíz.
- ✓ Query Type: Indica el tipo de consulta. Los valores de este campo incluyen todos los valores válidos para el campo Tipo en el formato RR, junto con algunos códigos más generales que coinciden con más de un tipo de RR.
- ✓ Query Class: Especifica la clase de consulta, normalmente Internet.

La sección de respuesta contiene los RR's que responden a la pregunta; la sección de autoridad contiene RR's que apuntan a un servidor de nombres autoritativo; la sección de registros adicional contiene RR's que se relacionan con la consulta pero que no son estrictamente respuestas a la pregunta.

9.3. Correo Electrónico (SMTP, Simple Mail Transfer Protocol) y MIME (Multi-purpose Internet Mail Extension)

La aplicación más utilizada virtualmente en cualquier sistema distribuido es el correo electrónico. El protocolo simple de transferencia de correo (SMTP, Simple Mail Transfer Protocol) ha sido siempre el caballo de batalla del conjunto de protocolos TCP/IP. Sin embargo, SMTP ha estado tradicionalmente limitado a la distribución de mensajes sencillos de texto. Desde hace algunos años, ha surgido la demanda de distribuir

correo con capacidad de contener distintos tipos de datos, incluyendo voz, imágenes y secuencias de vídeo. Para satisfacer estos requisitos se ha definido un nuevo estándar sobre la base de SMTP: las extensiones multipropósito de correo electrónico (MIME, Multi-Purpose Internet Mail Extension). En esta sección se describirán SMTP y luego MIME.

9.3.1. Correo Electrónico

Para implementar la arquitectura de correo de Internet, es necesario recurrir a un conjunto de normas. De ellas se destacan cuatro estándares:

- ✓ Protocolo de oficina de correos (POP3): POP3 permite a un cliente de correo electrónico (agente de usuario) descargar un correo electrónico de un servidor de correo electrónico (MTA, Message Transfer Agent). Los agentes de usuario POP3 se conectan a través de TCP/IP al servidor (normalmente el puerto 110). El agente de usuario introduce un nombre de usuario y contraseña (almacenados internamente para mayor comodidad o introducidos cada vez por el usuario para mayor seguridad). Tras la autorización, el UA (User Agent) puede ejecutar comandos POP3 para recuperar y eliminar el correo.
- ✓ Protocolo de Acceso al Correo de Internet (IMAP): Al igual que POP3, IMAP también permite un cliente de correo electrónico para acceder al correo en un servidor de correo electrónico. IMAP también utiliza TCP/IP, con el puerto TCP 143 del servidor. IMAP es más complejo que POP3. IMAP proporciona autenticación más fuerte que POP3 y proporciona otras funciones no soportadas por POP3.
- ✓ Protocolo simple de transferencia de correo (SMTP): Este protocolo se utiliza para la transferencia de correo de un agente de usuario a un MTA y de un MTA a otro.
- ✓ Extensiones de Correo de Internet Multipropósito (MIME): MIME complementa a SMTP y permite encapsular mensajes multimedia (no textuales) dentro de un mensaje dentro de un mensaje SMTP estándar.

Protocolo simple de Transferencia de Correo (SMTP)

SMTP es el protocolo estándar para la transferencia de correo entre computadores en la familia de protocolos TCP/IP. SMTP está definido en el RFC 821. Aunque los mensajes transferidos por SMTP normalmente siguen el formato definido en el RFC 822, que se describirá más adelante, a SMTP no le atañe ni el formato ni el contenido de los mensajes transferidos, con dos excepciones. Se hace referencia a este concepto diciendo que SMTP utiliza la información escrita en el sobre del correo (cabecera del mensaje), pero que no examina el contenido (cuerpo del mensaje) del sobre. Las dos excepciones son las siguientes:

1. SMTP normaliza el conjunto de caracteres del mensaje al conjunto ASCII de 7 bits.

2. SMTP incorpora información al comienzo del mensaje transferido que indica el camino que ha seguido el mismo.

Funcionamiento básico del correo electrónico

La Figura 9.3.1.1 muestra el flujo general del correo en un sistema típico. Aunque gran parte de esta actividad se encuentra fuera del ámbito de SMTP, la figura muestra el contexto dentro del que opera normalmente SMTP.

Para empezar, el correo lo crea un programa agente de usuario en respuesta a una entrada de usuario. Cada mensaje creado consta de una cabecera que incluye la dirección de correo electrónico del destinatario junto con otra información y un cuerpo que contiene el mensaje a enviar. Estos mensajes se sitúan de alguna forma en una cola de espera y se suministran como entrada a un programa emisor SMTP, que normalmente es un programa servidor siempre presente en el computador.

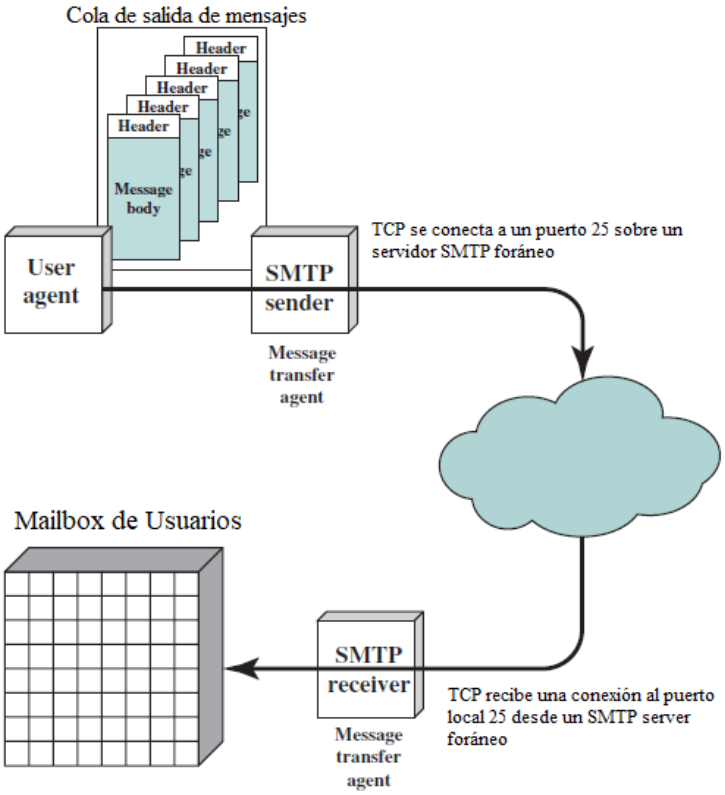


Figura 9.3.1.1: Flujo de un sistema de correo electrónico

Aunque la estructura de la cola de salida de correo será diferente según el sistema operativo del computador, cada mensaje en cola tiene conceptualmente dos partes:

1. El texto del mensaje, compuesto por:
   - ✓ La cabecera RFC 822: ésta constituye el sobre del mensaje e incluye una indicación del destinatario o destinatarios deseados.
   - ✓ El cuerpo del mensaje, escrito por el usuario.
2. Una lista de destinos de correo.

La lista de destinos de correo para los mensajes la obtiene el agente de usuario a partir de la cabecera RFC 822 del mensaje. En algunos casos, el destino o destinos se especifican literalmente en la cabecera del mensaje. En otros casos, el agente de usuario puede necesitar expandir los nombres de la lista de correo, eliminar duplicados y reemplazar nombres mnemónicos por los nombres de buzones de correo reales. Si se indica alguna copia de carbón oculta (BCC, Blind Carbon Copy), el agente de usuario necesita preparar mensajes para ajustarse a este requisito. La idea básica consiste en que los distintos formatos y estilos preferidos por los humanos en la interfaz de usuario se reemplacen por una lista normalizada adecuada para el programa de envío SMTP.

El programa de envío SMTP toma los mensajes de la cola de salida de correo y los transmite al computador destino adecuado mediante transacciones SMTP a través de una o más conexiones TCP al puerto 25 del computador objetivo. Un computador puede tener varios programas de envío SMTP activos simultáneamente si tiene una gran cantidad de volumen de correo de salida, y también debe tener la capacidad de crear receptores SMTP bajo demanda para que el correo que provenga de un computador no retarde el correo de otro.

Cuando el emisor SMTP completa la entrega de un mensaje específico a uno o más usuarios de un computador concreto, el emisor elimina los correspondientes destinatarios de la lista de destinos del mensaje. Cuando se han procesado todos los destinos de un mensaje concreto, éste se elimina de la cola. En el procesamiento de una cola, el emisor SMTP puede llevar a cabo varias optimizaciones. Si un mensaje determinado se envía a distintos usuarios de un único computador, sólo es necesario enviar el texto del mensaje una vez. Si hay listos para enviar varios mensajes al mismo computador, el emisor SMTP puede abrir una sola conexión TCP, transferir los múltiples mensajes y cerrar la conexión, en lugar de abrir y cerrar una conexión para cada mensaje.

El emisor SMTP debe hacer frente a diversos tipos de errores. El computador destino puede estar fuera de alcance, no encontrarse en funcionamiento o puede fallar la conexión TCP mientras se está transfiriendo el correo. El emisor puede volver a poner en cola el correo para efectuar la transferencia más tarde, pero renunciando a intentarlo otra vez tras un período de tiempo determinado en lugar de mantenerlo en la cola indefinidamente. Un error común consiste en indicar una dirección de destinatario errónea, debido a un error en la entrada del usuario o a que el destino deseado tiene una nueva dirección en un computador diferente. Si es posible, el emisor SMTP debe redirigir el mensaje o devolver una notificación de error al que originó el mensaje.

El protocolo SMTP se utiliza para transferir un mensaje desde el emisor SMTP al receptor SMTP a través de una conexión TCP. SMTP intenta proporcionar un funcionamiento fiable, pero no garantiza la recuperación de mensajes perdidos. No hay

confirmación "extremo a extremo" de la entrega con éxito del mensaje para el que lo originó y tampoco se garantiza la notificación de los errores. Sin embargo, el sistema de correo basado en SMTP es generalmente considerado fiable.

El receptor SMTP acepta cada mensaje que llega y lo sitúa en el buzón de correo del usuario adecuado o lo copia en la cola local de correo de salida para reenviarlo si es necesario. El receptor SMTP debe ser capaz de verificar los destinos locales de correo y atender los errores, incluyendo los errores de transmisión y de falta de espacio para almacenamiento.

El emisor SMTP es responsable del mensaje hasta el momento en el que el receptor SMTP indica que la transferencia se ha completado. Sin embargo, esto sólo significa que el mensaje ha llegado al receptor SMTP, no que el mensaje haya sido entregado y recogido por el destinatario final deseado. Las responsabilidades del receptor SMTP sobre al tratamiento de errores se restringen generalmente a abandonar conexiones TCP que fallen o que estén inactivas por largos períodos de tiempo. Por ello, la mayor parte de las responsabilidades en cuanto a la recuperación de los errores se sitúa en el emisor. Los errores que se produzcan durante la indicación de la finalización pueden producir la duplicación de mensajes, pero no su pérdida.

En la mayoría de los casos, los mensajes van directamente desde la máquina que origina el mensaje hasta la máquina destino a través de una única conexión TCP. Sin embargo, el mensaje pasará ocasionalmente por máquinas intermedias mediante la capacidad de reenvío de SMTP, en cuyo caso el mensaje debe atravesar una sucesión de conexiones TCP entre la fuente y el destino. Uno de los casos en que esto se produce consiste en que el emisor especifique una ruta al destino mediante una secuencia de servidores. Un caso más usual es el del reenvío requerido debido al traslado de un usuario.

Es importante señalar que el protocolo SMTP se limita a la conversación que tiene lugar entre el emisor SMTP y el receptor SMTP. La función principal de SMTP es la transferencia de mensajes, aunque existan algunas funciones auxiliares para la verificación y procesado del destino del correo.

Visión general de SMTP

El funcionamiento de SMTP consiste en una serie de órdenes y respuestas intercambiadas entre el emisor y receptor SMTP. La iniciativa la lleva el SMTP emisor, quien establece la conexión TCP. Una vez que se ha establecido la conexión, el emisor SMTP envía órdenes al receptor a través de la conexión. Cada orden genera exactamente una respuesta del receptor SMTP.

La Tabla 9.3.1.2 muestra las órdenes SMTP. Cada orden consta de una única línea de texto que comienza con un código de orden de 4 letras, seguido en algunos casos por un campo de argumento. La mayoría de las repuestas constan de una sola línea, aunque son posibles respuestas de varias líneas. En la tabla se señalan aquellas órdenes que todos los receptores deben ser capaces de reconocer. Las otras órdenes son opcionales y pueden ser ignoradas por el receptor.

Las respuestas SMTP se muestran en la Tabla 9.3.1.3. Cada respuesta comienza con un código de tres dígitos, pudiendo ir seguida por información adicional. El primer dígito indica la categoría de la respuesta:

- ✓ Respuesta de finalización positiva: la acción solicitada se ha completado satisfactoriamente. Puede iniciarse una nueva solicitud.
- ✓ Respuesta intermedia positiva: la orden ha sido aceptada, pero la acción solicitada se encuentra suspendida, pendiente de la recepción de información adicional. El emisor SMTP debe enviar otra orden especificando está información. Esta respuesta se utiliza en grupos de secuencias de órdenes.
- ✓ Respuesta de finalización negativa transitoria: la orden no se aceptó y la acción solicitada no se llevó a cabo. Sin embargo, la condición de error es temporal y puede solicitarse la acción de nuevo.

Tabla 9.3.1.2: Órdenes SMTP

Nombre	Formato de la orden	Descripción
HELO	HELO ⟨SP⟩ ⟨dominio⟩ ⟨CRLF⟩	Envía identificación
MAIL	MAIL ⟨SP⟩ FROM: ⟨camino inverso⟩ ⟨CRLF⟩	Identifica al que origina el correo
RCPT	RCPT ⟨SP⟩ TO: ⟨camino al destino⟩ ⟨CRLF⟩	Identifica al destinatario del correo
DATA	DATA ⟨CRLF⟩	Transfiere el texto del mensaje
RSET	RSET ⟨CRLF⟩	Aborta la transacción del correo en curso
NOOP	NOOP ⟨CRLF⟩	Operación nula
QUIT	QUIT ⟨CRLF⟩	Cierra la conexión TCP
SEND	SEND ⟨SP⟩ FROM: ⟨camino inverso⟩ ⟨CRLF⟩	Envía correo al terminal
SOML	SOML ⟨SP⟩ FROM: ⟨camino inverso⟩ ⟨CRLF⟩	Envía correo al terminal si es posible. En caso contrario, lo envía al buzón
SAML	SAML ⟨SP⟩ FROM: ⟨camino inverso⟩ ⟨CRLF⟩	Envía correo al terminal y al buzón
VRFY	VRFY ⟨SP⟩ ⟨cadena⟩ ⟨CRLF⟩	Confirma el nombre del usuario
EXPN	EXPN ⟨SP⟩ ⟨cadena⟩ ⟨CRLF⟩	Devuelve la lista de miembros de una lista de correo
HELP	HELP [⟨SP⟩ ⟨cadena⟩] ⟨CRLF⟩	Envía documentación específica del sistema
TURN	TURN ⟨CRLF⟩	Intercambia el rol del emisor y el receptor

⟨CRLF⟩ = retorno de carro, salto de línea
⟨SP⟩ = espacio
Los corchetes indican elementos opcionales.
Los comandos sombreados son opcionales en implementaciones conformes a SMTP.

- ✓ Respuesta de finalización negativa permanente: la orden no se aceptó y la acción solicitada no se realizó.

La operación básica de SMTP se produce en tres fases: establecimiento de la conexión, intercambio de uno o más pares orden-respuesta y cierre de la conexión. A continuación, se examina cada una de las fases.

Establecimiento de la conexión

Un emisor SMTP intentará establecer una conexión TCP con un computador destino cuando tenga uno o más mensajes de correo para entregarle. La secuencia es bastante sencilla:

- ✓ El emisor abre una conexión TCP con el receptor.
- ✓ Una vez que se ha establecido la conexión, el receptor se identifica a sí mismo con la respuesta *"220 servicio preparado"*.
- ✓ El emisor se identifica a sí mismo con la orden HELO.
- ✓ El receptor acepta la identificación del emisor mediante la respuesta *"250 OK"*.

Si el servicio de correo no está disponible en el destino, el computador destino devuelve la respuesta *"421 servicio no disponible"* en el paso 2 y finaliza el proceso.

Tabla 9.3.1.3: Respuestas SMTP

Código	Descripción
	Respuesta de finalización positiva
211	Estado del sistema o respuesta de ayuda del sistema.
214	Mensaje de ayuda. (Información de cómo utilizar el receptor o el significado de una orden particular no estándar. Esta respuesta es sólo útil al usuario humano)
220	⟨dominio⟩ Servicio preparado
221	⟨dominio⟩ Servicio cerrando el canal de transmisión
250	Acción de correo solicitada correcta, completada
251	Usuario no local. Se reenviará a ⟨camino al destino⟩
	Respuesta intermedia positiva
354	Comenzar el texto del correo. Acabar con ⟨CRLF⟩.⟨CRLF⟩
	Respuesta de finalización negativa transitoria
421	⟨dominio⟩ Servicio no disponible; perdiendo canal de transmisión (ésta puede ser la respuesta a cualquier orden si el servicio sabe que debe apagarse)
450	Acción de correo solicitada no ejecutada; buzón de correo no disponible (por ejemplo, buzón ocupado)
451	Cancelada acción solicitada; error local en el procesamiento
452	Acción solicitada no ejecutada; almacenamiento del sistema insuficiente
	Respuesta de finalización negativa permanente
500	Error de sintaxis; orden no reconocida (esto puede incluir errores, como línea de orden demasiado larga)
501	Error de sintaxis en los parámetros o los argumentos
502	Orden no implementada
503	Secuencia de órdenes incorrecta
504	Parámetro de orden no implementado
550	Acción solicitada no ejecutada; buzón de correo no disponible (por ejemplo, buzón no encontrado o no se accedió)
551	Usuario no local, por favor, pruebe con ⟨camino al destino⟩
552	Acción de correo solicitada cancelada; excedida la asignación de espacio de almacenamiento
553	Acción solicitada no ejecutada; nombre del buzón de correo no permitido (por ejemplo, sintaxis de correo incorrecta)
554	Transacción fallida

Transferencia de correo

Una vez que se ha establecido una conexión, el emisor SMTP puede enviar uno o más mensajes al receptor SMTP. Existen tres fases lógicas en la transferencia de un mensaje:

- ✓ Una orden MAIL identifica al que originó el mensaje.
- ✓ Una o más órdenes RCPT identifican a los destinatarios de este mensaje.
- ✓ Una orden DATA transfiere el texto del mensaje.

La orden MAIL proporciona el camino inverso, que puede utilizarse para informar de errores. Si el receptor está preparado para aceptar mensajes de esta fuente, entonces devuelve una respuesta *"250 OK"*. En otro caso devuelve una respuesta indicando un fallo al ejecutar la orden (códigos 451, 452, 552) o un error en la orden (códigos 421, 500 y 501).

La orden RCPT identifica a un destinatario individual de los datos del correo. Se puede especificar varios destinatarios mediante el uso múltiple de esta orden. Se devuelve una respuesta distinta por cada orden RCPT, con una de las siguientes posibilidades:

1. El receptor acepta el destinatario con una respuesta *"250"*. Esto indica que el buzón de correo designado se encuentra en el sistema del receptor.

2. El destino necesitará el reenvío del correo que será efectuado por el receptor (251).
3. El destino requerirá una operación de reenvío, pero el receptor no lo reenviará. El emisor debe volver a enviarlo a la dirección de reenvío (551).
4. No existe un buzón de correo en este computador para este destinatario (550).
5. Se rechaza el destino debido a algún fallo de ejecución (códigos 450, 451, 452 y 552 y 553) o a un error en la orden (códigos 421, 500, 501 y 503).

La ventaja de utilizar una fase separada para RCPT es que el emisor no enviará el mensaje hasta que esté seguro de que el receptor está dispuesto a recibirlo para al menos un destinatario, evitando de este modo la sobrecarga que constituye enviar un mensaje completo para averiguar que se desconoce al destinatario. Una vez que el receptor SMTP está de acuerdo en recibir el mensaje de correo para al menos un destinatario, el emisor SMTP utiliza la orden DATA para iniciar la transferencia del mensaje. Si el receptor SMTP sigue dispuesto a recibir el mensaje, devuelve una respuesta *"354"*. En otro caso, el receptor devuelve una respuesta indicando que hubo un fallo al ejecutar la orden (códigos 451 o 554) o un error en la orden (códigos 421, 500, 501 o 503). Si se devuelve la respuesta «354», el emisor SMTP procede a enviar el mensaje sobre la conexión TCP en forma de una secuencia de líneas ASCII. El fin del mensaje se indica con una línea que contiene únicamente un punto (*". "*). El receptor SMTP responde con una respuesta *"250 OK"* si se acepta el mensaje o con el código de error apropiado (451, 452, 552 o 554) en caso contrario.

El siguiente ejemplo, tomado del RFC 821, muestra el proceso:

S:MAIL FROM:<Smith@Alpha.ARPA>

R: 250 OK

S:RCPT TO:<Jones@Beta.ARPA>

R: 250 OK

S:RCPT TO:<Green@Beta.ARPA>

R: 550 No such user here

S:RCPT TO:<Brown@Beta.ARPA>

R: 250 OK

S:DATA

R: 354 Start mail input; end with <CRLF>.<CRLF>

770 Comunicaciones y redes de computadores

S: Bla bla bla ...

S: ... etc. etc. etc.

S: <CRLF>.<CRLF>

R: 250 OK

El emisor SMTP está transmitiendo un correo creado por el usuario Smith@Alpha.ARPA. El mensaje va dirigido a tres usuarios en la máquina Beta.ARPA, llamados Jones, Green y Brown. El receptor SMTP indica que tiene buzones de correo para Jones y Brown, pero que no tiene información sobre Green. Ya que al menos uno de los destinatarios deseados ha sido verificado, el emisor procede a enviar el mensaje de texto.

Cierre de la conexión

El emisor SMTP cierra la conexión en dos pasos. Primero, el emisor envía una orden QUIT y espera una respuesta. El segundo paso consiste en iniciar una operación de cierre de la conexión TCP. El receptor inicia su cierre TCP después de enviar su respuesta a la orden QUIT.

RFC 822

El RFC 822 define un formato para los mensajes de texto que se envían utilizando el correo electrónico. El estándar SMTP adopta el RFC 822 como formato a utilizar en la construcción de mensajes para su transmisión a través de SMTP. En el contexto del RFC 822, los mensajes se componen de un sobre y un contenido. El sobre contiene toda la información necesaria para llevar a cabo la transmisión y la entrega. El contenido está compuesto por el objeto que ha de entregarse al destinatario. El estándar RFC 822 se aplica solamente al contenido. Sin embargo, el estándar del contenido incluye un conjunto de campos de cabecera que puede utilizar el sistema de correo para crear el sobre, con la finalidad de facilitar a los programas la adquisición de esa información.

Un mensaje RFC 822 consta de una secuencia de líneas de texto y utiliza una plantilla general. Es decir, un mensaje consta de varias líneas de cabecera que siguen un formato rígido, seguidas de una sección correspondiente al cuerpo compuesto por texto arbitrario.

Una línea de cabecera consta normalmente de una palabra clave, seguida por dos puntos (*":"*) y los argumentos de la palabra clave. Este formato permite que una línea larga sea fragmentada en varias líneas. Las palabras clave más frecuentemente utilizadas son: *"From"* (*"de"*), *"To"* (*"a"*), *"Subject"* (*"asunto"*) y *"Date"* (*"fecha"*). Aquí se muestra un ejemplo de mensaje:

Date: Tue, 16 Jan 1996 10:37:17 (EST)

From: «William Stallings» aws@host.comb

Subject: La sintaxis en el RFC 822

To: Smith@Other-host.com

Cc: Jones@Yet-Another-Host.com

Hola. Esta parte constituye el comienzo real del cuerpo del mensaje, separado de la cabecera del mensaje por una linea1 en blanco.

Otro campo que se encuentra a menudo en la cabecera RFC 822 es "Message-ID" (*"identificador del mensaje"*). Este campo contiene un identificador único asociado a este mensaje.

9.3.2. Extensiones Multipropósito de Correo Electrónico (MIME)

MIME (Multi purpose Internet Mail Extensions) es una extensión del marco de trabajo establecido en el RFC 822 que pretende abordar algunos de los problemas y limitaciones del uso de SMTP y del RFC 822 para el correo electrónico. (Rodriguez A., 2002) señala las siguientes limitaciones del esquema SMTP/822:

1. SMTP no puede transmitir ficheros ejecutables u otros objetos binarios. Se utilizan diversos esquemas para convertir ficheros binarios a formato texto de forma que los sistemas de correo de SMTP puedan utilizarlos, incluyéndose el popular esquema de UNIX UUencode/UUdecode. Sin embargo, ninguno de estos procedimientos de conversión constituye un estándar, ni siquiera de facto.
2. SMTP no puede transmitir datos de texto que incluyan caracteres de lenguajes nacionales, ya que éstos se representan por códigos de 8 bits con valores de 128 en decimal o superiores, y SMTP está limitado a caracteres ASCII de 7 bits.
3. Los servidores SMTP pueden rechazar mensajes de correo que superen un cierto tamaño.
4. Las pasarelas de SMTP que traducen caracteres ASCII a código EBCDIC no utilizan un conjunto consistente de correspondencias, lo que da lugar a problemas en la traducción.
5. Las pasarelas de SMTP a redes de correo electrónico X.400 no pueden manejar los datos no textuales incluidos en mensajes X.400.
6. Algunas implementaciones no se adhieren completamente al estándar de SMTP definido en el RFC 821. Éstos son algunos de los problemas comunes que aparecen:
    - ✓ La eliminación, incorporación o reordenación de caracteres retorno de carro y de salto de línea.
    - ✓ El truncado o el solapamiento de las líneas de longitud mayor a 76 caracteres.
    - ✓ La eliminación de espacios en blanco finales (caracteres tabuladores y espacios).
    - ✓ El completado de las líneas de un mensaje para conseguir la misma longitud en todas.

- ✓ La conversión de los caracteres tabuladores en varios caracteres de espacio.

MIME pretende resolver estos problemas de forma que resulte compatible con las implementaciones existentes del RFC 822. La especificación se encuentra en los RFC del 2045 al 2049.

Visión general

La especificación de MIME incluye los siguientes elementos:

1. Se definen cinco campos nuevos de la cabecera del mensaje, los cuales pueden incluirse en una cabecera RFC 822. Estos campos proporcionan información acerca del cuerpo del mensaje.
2. Se definen varios formatos para el contenido, normalizando así las representaciones que dan soporte al correo electrónico multimedia.
3. Se definen esquemas de codificación de transferencia, posibilitando así la conversión de cualquier formato de contenido a un formato protegido contra las alteraciones que efectúe el sistema de correo.

En esta subsección se describen los cinco nuevos campos de la cabecera del mensaje. En las dos siguientes se examinan los formatos de contenido y los esquemas de codificación de transferencia.

Los cinco campos de cabecera definidos en MIME son:

- ✓ MIME-Version (*"versión de MIME"*): el valor de parámetro debe ser *"1.0"*. Este campo indica que el mensaje cumple los RFC.
- ✓ Content-Type (*"tipo de contenido"*): describe los datos contenidos en el cuerpo del mensaje con suficiente detalle, de tal manera que el agente de usuario receptor pueda escoger un agente o un mecanismo apropiado para presentar los datos al usuario o, en otro caso, ocuparse de los datos de forma adecuada.
- ✓ Content-Transfer-Encoding (*"esquema de codificación de transferencia del contenido"*): indica el tipo de transformación que se ha utilizado para representar el cuerpo del mensaje de modo que sea aceptable para el transporte del correo.
- ✓ Content-ID (*"identificador del contenido"*): utilizado para identificar de forma unívoca entidades MIME en múltiples contextos.
- ✓ Content-Description (*"descripción del contenido"*): una descripción en texto nativo del objeto incluido en el cuerpo. Esto es útil cuando el objeto no se puede visualizar (por ejemplo, datos de audio).

En una cabecera RFC 822 normal pueden aparecer todos o alguno de estos campos. Una implementación conforme debe permitir los campos MIME-Version, Content-Type y Content-Transfer-Encoding. Los campos Content-ID y Content-Description son opcionales y pueden ser ignorados por la implementación del destinatario.

Tipos de contenido MIME

El grueso de la especificación de MIME está relacionado con la definición de varios tipos de contenidos. Esto refleja la necesidad de proporcionar formas normalizadas de tratar una gran variedad de representaciones de información en un entorno multimedia.

La Tabla 9.3.2.1 enumera los tipos de contenido. Existen siete tipos de contenidos principales diferentes y un total de 14 subtipos. En general, un tipo de contenido declara el tipo general de los datos y el subtipo especifica un formato particular para ese tipo de datos.

Para un cuerpo del tipo text (*"texto"*), no se requiere un software especial para obtener el significado completo del texto, aparte de soportar el conjunto de caracteres indicado. El único subtipo definido es el texto nativo, que consiste simplemente en una cadena de caracteres ASCII o caracteres ISO 8859. Una versión anterior de la especificación MIME incluía el subtipo richtext (*"texto enriquecido"*), que admite una mayor flexibilidad en el formateo. Se espera que este subtipo vuelva a aparecer en un RFC posterior.

El tipo multipart (*"multiparte"*) indica que el cuerpo del mensaje contiene múltiples partes independientes. El campo Content-Type de la cabecera incluye un parámetro, llamado delimitador, que define la marca utilizada para delimitar las distintas partes del cuerpo. Este delimitador no debe aparecer en ninguna de las partes del mensaje. Cada límite comienza en una línea nueva y consta de dos guiones seguidos por el valor del delimitador. El delimitador final, que indica el fin de la última parte, tiene además dos guiones como sufijo. Dentro de cada parte puede existir una cabecera opcional MIME ordinaria.

Tabla 9.3.2.1: Tipos de contenido MIME

Tipo	Subtipo	Descripción
Texto	Nativo	Texto no formateado; puede ser ASCII o ISO 8859
Multipart («multiparte»)	*Mixed* («mixto»)	Las diferentes partes son independientes pero van a ser transmitidas juntas. Se deben presentar al receptor en el mismo orden en que aparecen en el mensaje de correo.
	Parallel («paralelo»)	Difiere del subtipo *mixed* solamente en que no se define orden para la entrega de las partes al receptor.
	Alternative («Alternativo»)	Las diferentes partes son versiones alternativas de la misma información. Están ordenadas en fidelidad creciente al original y el sistema de correo destino debe mostrar la mejor versión para el usuario.
	Digest («resumen»)	Similar al subtipo *mixed*, pero el tipo/subtipo por defecto para cada parte es message/rfc822.
Message («mensaje»)	rfc822	El propio cuerpo es un mensaje encapsulado que cumple con el RFC 822.
	Partial («parcial»)	Utilizado para permitir la fragmentación de elementos de correo grandes de forma que sea transparente al destino.
	External-body («cuerpo-externo»)	Contiene un puntero a un objeto que existe en otra parte.
Image («Imagen»)	jpeg	La imagen está en formato JPEG, codificación JFIF.
	gif	La imagen está en formato GIF.
Video («Vídeo»)	mpeg	Formato MPEG.
Audio	*Basic* («Básico»)	Codificación en ley-*mu* de RDSI, con un canal de 8 bits.
Application («aplicación»)	*Postscript*	Postscript de Adobe.
	octet-stream («flujo de octetos»)	Datos binarios generales compuestos por bytes de 8 bits.

Esquemas de codificación de transferencia de MIME

El otro componente principal de la especificación MIME, además de la especificación del tipo de contenido, es la definición de la codificación de transferencia de los cuerpos de los mensajes. Su objetivo es proporcionar una entrega fiable a través del mayor número de entornos diversos.

El estándar MIME define dos métodos para codificar los datos. El campo Content-Transfer-Encoding en realidad puede tomar seis valores, como se muestra en la Tabla 9.3.2.2. Sin embargo, tres de estos valores (7bit, 8bit y binary) indican que no se ha efectuado ninguna codificación, pero en cambio proporcionan información sobre la naturaleza de los datos. Para la transferencia SMTP, es seguro utilizar la forma 7bit. Las formas 8bit y binary se pueden utilizar en otros contextos de transporte de correo. Otro valor de esquema de codificación de contenido es x-token (*"esquema x"*), que indica que se utiliza algún otro esquema de codificación del que se proporcionará el nombre. Éste podría ser un esquema específico de un fabricante o de una aplicación concreta. Los dos esquemas de codificación reales definidos son el quoted-printable (*"imprimible textualmente"*) y el base64. Los dos esquemas se definen para ofrecer la posibilidad de escoger entre una técnica de transferencia que es esencialmente legible por humanos y otra que es segura para todos los tipos de datos en una forma razonablemente compacta.

Tabla 9.3.2.2: Esquemas de codificación de transferencia MIME

7bit	Todos los datos se representan por líneas cortas de caracteres ASCII.
8bit	Las líneas son cortas, pero puede haber caracteres no ASCII (octetos con el bit de orden más alto establecido).
Binary («binario»)	Además de incluir caracteres no ASCII, las líneas pueden no ser lo suficientemente cortas para el transporte SMTP.
quoted-printable («imprimible textualmente»)	Codifica los datos de tal forma que si la mayoría de los datos que se codifican son texto ASCII, el texto codificado permanece en gran medida reconocible por los usuarios humanos.
base64	Codifica los datos convirtiendo bloques de 6 bits en bloques de 8 bits, todos ellos caracteres ASCII imprimibles.
x-token («esquema x»)	Una codificación no estándar.

La codificación de transferencia quoted-printable es útil cuando los datos son en buena parte octetos que corresponden a caracteres ASCII imprimibles. En esencia, representa caracteres no seguros por medio de la representación en hexadecimal de sus códigos e introduce rupturas de líneas reversibles (auxiliares) para limitar la longitud de las líneas del mensaje a 76 caracteres. Las reglas de codificación son las siguientes:

1. Representación general de 8 bits: esta regla se va a utilizar cuando no se aplique ninguna de las otras reglas. Cualquier carácter se representa por un signo igual seguido de una representación en hexadecimal de dos dígitos del valor del octeto. Por ejemplo, el carácter de salto de página ASCII, que tiene un valor en decimal de 8 bits de *"12"*, se representa por *"=0C"*.

2. Representación literal: cualquier carácter en el rango decimal comprendido entre 33 (*"!"*) y 126 (*"V"*), exceptuando el decimal 61 (*"%"*), se representa por el propio carácter ASCII.

3. Espacio en blanco: los octetos con valor 9 y 32 se pueden representar como los caracteres ASCII de tabulador y espacio respectivamente, excepto al final de la línea. Cualquier espacio en blanco (tabulador o espacio) al final de una línea se debe representar según la regla 1. Al decodificar se suprime cualquier espacio en blanco al final de la línea. Esto elimina cualquier espacio en blanco incorporado por agentes de transporte intermedios.

4. Ruptura de línea: cualquier ruptura de línea, independientemente de su representación inicial, se representa por la ruptura de línea del RFC 822, que consiste en la combinación de un retorno de carro y un salto de línea (<CRLF>).

5. Ruptura de línea reversible: si una línea codificada va a tener una longitud mayor que 76 caracteres (excluyendo <CRLF>), se debe insertar una ruptura de línea reversible antes de la posición 76. Una ruptura de línea reversible consiste en la secuencia en hexadecimal 3D0D0A, que es el código ASCII para el signo igual seguido del retorno de carro y el salto de línea.

La codificación de transferencia base64, también conocida como codificación radix-64, es una técnica común para codificar datos binarios arbitrarios de forma que sean invulnerables al procesamiento de los programas de transporte de correo. Esta técnica convierte una entrada binaria arbitraria en una salida de caracteres imprimibles. Esta forma de codificación tiene las siguientes características relevantes:

1. El rango de la función es un conjunto de caracteres representable universalmente en todos los sitios, no una codificación binaria específica de ese conjunto de caracteres. Así, los propios caracteres se pueden codificar en cualquier forma que sea necesaria por un sistema determinado. Por ejemplo, el carácter *"E"* se representa en un sistema basado en ASCII por el valor hexadecimal 45 y en un sistema basado en EBCDIC por el valor hexadecimal C5.
2. El conjunto de caracteres consta de los 65 caracteres imprimibles, uno de los cuales se utiliza para el relleno. Con $2^6 = 64$ caracteres disponibles, cada carácter se puede utilizar para representar 6 bits de entrada.
3. No se incluyen caracteres de control en el conjunto. Así, un mensaje codificado en base64 puede atravesar sistemas de tratamiento de correo que comprueben el flujo de datos para encontrar caracteres de control.
4. El carácter de guion (*"–"*) no se utiliza. Este carácter tiene significado en el formato RFC 822, por lo que debe evitarse.

La Tabla 9.3.2.3 muestra la equivalencia de los valores de 6 bits de entrada con los caracteres. El conjunto de caracteres consta de los caracteres alfanuméricos más *"+"* y *"/"*. El carácter *"="* se utiliza como carácter de relleno.

Tabla 9.3.2.4: Esquema de codificación radix-64

Valor de 6 bits	Codificación del carácter	Valor de 6 bits	Codificación del carácter	Valor de 6 bits	Codificación del carácter	Valor de 6 bits	Codificación del carácter
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/
						(relleno)	=

La Figura 9.3.2.4 muestra un esquema de conversión sencillo. La entrada binaria se procesa en bloques de 3 octetos o 24 bits. Cada conjunto de 6 bits del bloque de 24 bits se convierte en un carácter. En la figura, los caracteres se muestran codificados como valores de 8 bits. En este caso típico, cada entrada de 24 bits se expande en una salida de 32 bits.

9.3.3. Protocolo de Oficina Postal v.3 (POP3, Post Office Protocol)

El Protocolo de Oficina Postal Versión 3 de POP, identificado como POP3, es un estándar de Internet definido en el RFC 1939. POP3 soporta las funciones básicas de descarga y borrado para la recuperación de correo electrónico. Para realizar una función del cliente (MUA, Message User Application, por ejemplo Zimbra[1]) al servidor (MS, Message Store), el MUA establece una conexión TCP con el MS, utilizando el puerto 110.

A continuación, la interacción pasa por tres estados distintos:

- ✓ Estado de autenticación: Durante este estado, el cliente debe autenticarse ante el usuario. Esto se suele hacer con una simple combinación de ID de usuario/contraseña, aunque existen opciones más sofisticadas.
- ✓ Estado de transacción: Una vez que el servidor autentifica con éxito al cliente, éste puede cliente puede acceder al buzón para recuperar y eliminar mensajes.
- ✓ Estado de actualización: Durante este estado, el servidor ejecuta todos los cambios solicitados por los comandos del cliente y luego cierra la conexión.

9.3.4. Protocolo de Acceso a Mensajes de Internet v.4 (IMAP4, Internet Message Access Protocol)

Protocolo de Acceso a Mensajes de Internet La versión 4 de IMAP está definida por el RFC 3501. Al igual que el POP, los servidores IMAP4 almacenan mensajes para múltiples usuarios para ser recuperados a petición del cliente, pero el modelo IMAP4 proporciona más funcionalidad a los usuarios que el modelo POP, incluyendo las siguientes características:

- ✓ Los clientes pueden tener múltiples buzones remotos desde los que se pueden recuperar los mensajes.
- ✓ Los clientes también pueden especificar criterios para la descarga de mensajes, como no transferir mensajes grandes a través de enlaces lentos.
- ✓ IMAP siempre mantiene los mensajes en el servidor y replica las copias a los clientes.
- ✓ IMAP4 permite a los clientes realizar cambios tanto cuando están conectados como cuando están desconectados. Cuando se desconecta (lo que se conoce como cliente desconectado), los cambios realizados en el cliente surten efecto en el servidor mediante la resincronización periódica del cliente y el servidor.

[1] Zimbra es un programa de código libre y gratuito que se puede usar como cliente de correo electrónico

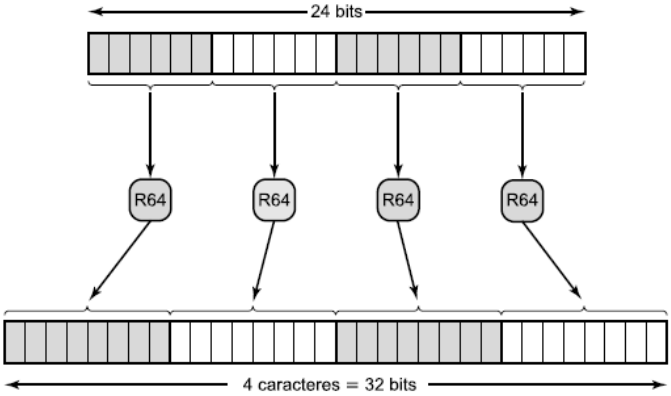


Figura 9.3.2.4: Conversión datos de radix-64 a caracteres de 8 bit

9.4. Protocolo de Transferencia de Hipertexto (HTTP, Hypertext Transfer Protocol)

El protocolo de transferencia de hipertexto (HTTP, Hypertext Transfer Protocol) es el protocolo base de la telaraña mundial (WWW, World Wide Web) y puede utilizarse en cualquier aplicación cliente-servidor que suponga la utilización de hipertextos. El nombre es más bien confuso, ya que HTTP no es un protocolo para transferir hipertexto, sino un protocolo para transmitir información con la eficiencia necesaria para efectuar saltos de hipertexto. Los datos transferidos por el protocolo pueden ser texto nativo, hipertexto, audio, imágenes o cualquier información accesible a través de Internet.

9.4.1. Descripción general

HTTP es un protocolo cliente/servidor orientado a transacciones. El uso más habitual de HTTP se produce entre un navegador y un servidor web. Para proporcionar fiabilidad, HTTP hace uso de TCP. No obstante, HTTP es un protocolo *"sin estados"*: cada transacción se trata independientemente. Por consiguiente, una implementación típica creará una conexión nueva entre el cliente y el servidor para cada transacción y cerrará la conexión tan pronto como se complete la transacción, aunque la especificación no impone esta relación uno a uno entre la transacción y la duración de la conexión.

La naturaleza de protocolo sin estados de HTTP es la adecuada para su aplicación habitual. Una sesión normal de un usuario con un navegador web supone obtener una secuencia de páginas y documentos web. La secuencia se lleva a cabo

idealmente de forma rápida y las localizaciones de las distintas páginas y documentos pueden encontrarse en varios servidores ampliamente distribuidos.

Otra característica importante de HTTP es que es flexible en cuanto a los formatos que puede manejar. Cuando un cliente emite una solicitud a un servidor, puede incluir una lista priorizada de formatos con los que puede operar, respondiendo el servidor con el formato adecuado. Por ejemplo, un navegador Lynx no puede operar con imágenes, por lo que el servidor no necesita transmitir ninguna imagen de las existentes en las páginas web. Esta disposición evita la transmisión de información innecesaria y proporciona la base para ampliar el conjunto de formatos con nuevas especificaciones estándares y propietarias.

La Figura 9.4.1.1 muestra tres ejemplos del funcionamiento de HTTP. El caso más sencillo es aquel en el que un agente de usuario establece una conexión directa con el servidor origen. El agente de usuario es el cliente que inicia la solicitud, como es el caso de un navegador web actuando de parte de un usuario final. El servidor origen es el servidor en el que se encuentra el recurso de interés. Un ejemplo de esto lo constituye un servidor web en el que reside la página web de inicio deseada. Para este caso, el cliente abre una conexión TCP extremo a extremo entre el cliente y el servidor. El cliente emite entonces una solicitud HTTP. La solicitud consta de una orden concreta, denominada método, una URL y un mensaje de tipo MIME que contiene los parámetros de la solicitud, información acerca del cliente y, tal vez, alguna información adicional del contenido.

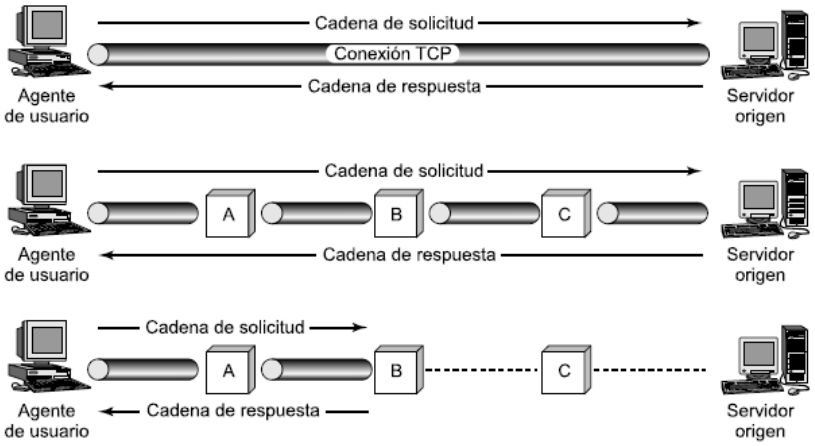


Figura 9.4.1.1: Funcionamiento de HTTP

Cuando el servidor recibe la solicitud, intenta llevar a cabo la acción solicitada y después devuelve una respuesta HTTP. La respuesta incluye información de estado, un código de éxito o error y un mensaje de tipo MIME que contiene información sobre el servidor, información sobre la misma respuesta y un posible cuerpo con el contenido. A continuación, se cierra la conexión TCP.

En la parte central de la Figura 9.4.1.1 se muestra un caso en que no existe una conexión TCP extremo a extremo entre el agente de usuario y el servidor origen. En su lugar, existen uno o más sistemas intermedios con conexiones TCP entre sistemas

lógicamente adyacentes. Cada sistema intermedio actúa como un retransmisor, de forma que una solicitud iniciada por el cliente se retransmite a través de los sistemas intermedios hasta el servidor y la respuesta del servidor se retransmite de vuelta al cliente.

Se definen tres tipos de sistemas intermedios en la especificación HTTP, ilustrados en la Figura 9.4.1.2: representante (proxy), pasarela (gateway) y túnel (tunnel).

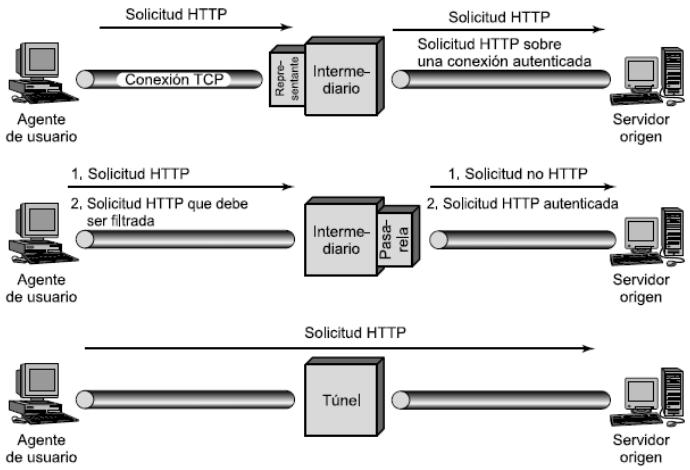


Figura 9.4.1.2: Sistemas HTTP intermedios

Proxy

Un proxy actúa en nombre de otros clientes y presenta las solicitudes de éstos a un servidor. El proxy actúa como servidor cuando interactúa con un cliente, y como cliente cuando interactúa con un servidor. Existen dos escenarios que requieren el uso de un representante:

- ✓ Intermediario de seguridad: el cliente y el servidor pueden estar separados por un intermediario de seguridad, como es el caso de un cortafuego, con el proxy en el lado del cliente. Normalmente, el cliente forma parte de una red protegida por un cortafuego y el servidor es externo a esta red protegida. En este caso, el servidor se debe autenticar al cortafuego para establecer una conexión con el proxy. Este acepta respuestas después de haber atravesado el cortafuego.
- ✓ Diferentes versiones de HTTP: si el cliente y el servidor ejecutan diferentes versiones de HTTP, el representante puede implementar ambas versiones y realizar las traducciones necesarias.

En resumen, un proxy es un agente de reenvío que recibe solicitudes de objetos URL, modifica las solicitudes y las reenvía hacia el servidor identificado en la URL.

Gateway

Un gateway es un servidor que se presenta al cliente como si se tratase de un servidor origen. Actúa en nombre de otros servidores que no pueden comunicarse directamente con un cliente. Existen dos escenarios en los que se pueden utilizar gateways:

- ✓ Intermediario de seguridad: el cliente y el servidor pueden estar separados por un intermediario de seguridad, como es el caso de un cortafuego, con la pasarela en el lado del servidor. Normalmente, el servidor está conectado a la red protegida por un cortafuego y el cliente es externo a esta red. En este caso, el cliente se debe autenticar en el gateway, que puede pasar entonces la solicitud al servidor.
- ✓ Servidor no HTTP: los navegadores web tienen incorporada la capacidad de contactar con servidores de otros protocolos distintos de HTTP, como servidores de FTP o Gopher. Esta capacidad también la puede proporcionar un gateway. El cliente realiza una solicitud HTTP a un servidor gateway. El servidor gateway contacta con el servidor FTP o Gopher pertinente para obtener el resultado deseado. Este resultado se convierte entonces a un formato adecuado para HTTP y se transmite de vuelta al cliente.

Túnel

A diferencia del proxy y el gateway, el túnel no realiza operaciones sobre las solicitudes y respuestas HTTP. En su lugar, un túnel es simplemente un punto de retransmisión entre dos conexiones TCP y los mensajes HTTP se transfieren sin modificaciones, como si hubiera una única conexión HTTP entre el agente de usuario y el servidor origen. Los túneles se utilizan cuando deba existir un sistema intermediario entre el cliente y el servidor, pero no sea necesario para ese sistema comprender el contenido de los mensajes. Un ejemplo de este caso lo constituye un cortafuego, en el que un cliente o un servidor externo a la red protegida puede establecer una conexión autenticada y después mantener esa conexión con objeto de realizar las transacciones HTTP.

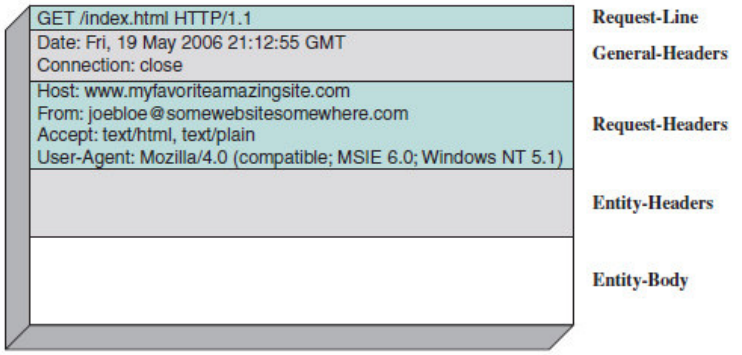
Caché

Volviendo a la Figura 9.4.1.1, su parte inferior muestra un ejemplo de una caché. Una caché es un servicio que puede almacenar solicitudes y respuestas previas para tratar las nuevas solicitudes. Si llega una solicitud nueva que es igual a una solicitud almacenada, entonces la caché puede proporcionar directamente la respuesta en lugar de acceder al recurso indicado en la URL. La caché puede operar en un cliente, en un servidor o en un sistema intermedio que no sea un túnel. En la Figura, en la última sección, el intermediario B ha almacenado una transacción de solicitud/respuesta, de forma que para una nueva solicitud correspondiente del cliente no se necesite recorrer la cadena completa hasta el servidor origen, sino que se procese por B.

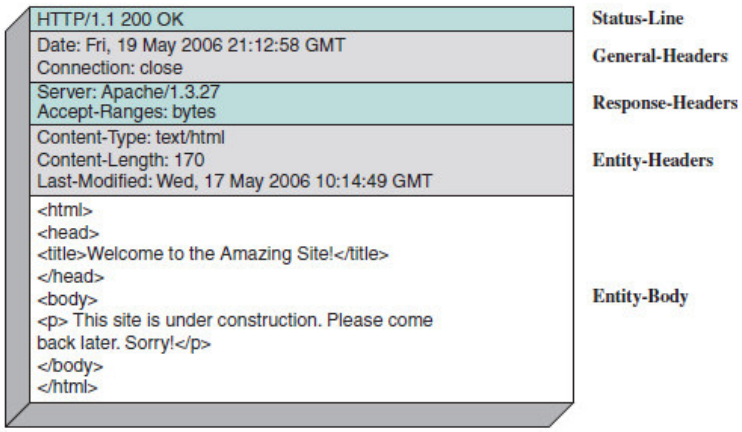
No todas las transacciones se pueden almacenar, pudiendo un cliente o un servidor indicar que ciertas transacciones pueden almacenarse sólo durante un determinado periodo de tiempo.

9.4.2. Mensajes

La mejor forma de describir la funcionalidad de HTTP es describir los elementos individuales del mensaje HTTP. HTTP consta de dos tipos de mensajes: solicitudes de los clientes a los servidores y respuestas de los servidores a los clientes. La estructura general de estos mensajes se muestra en la Figura 9.4.2.1. Más formalmente, utilizando la notación BNF (Backus-Naur Form) (véase Tabla 9.4.2.2).



(a) HTTP request (solicitud)



(b) HTTP response (respuesta)

Figura 9.4.2.1: Ejemplo de formatos de mensaje

Los mensajes «Simple-Request» y «Simple-Response» fueron definidos en HTTP/0.9. La solicitud consiste en una orden sencilla GET con la URL solicitada. La respuesta es simplemente un bloque que contiene la información identificada en la URL. En HTTP/1.1 se desaconseja la utilización de estos formatos simples, ya que impiden al

cliente realizar la negociación del contenido y al servidor identificar el tipo de contenido de la entidad devuelta.

En las solicitudes y respuestas completas se utilizan los siguientes campos:

- ✓ Request-Line: identifica el tipo de mensaje y el recurso solicitado.
- ✓ Response-Line: proporciona información de estado sobre esta respuesta.
- ✓ General-Header: contiene campos aplicables a los mensajes de solicitud y de respuesta, pero que no se aplican a la entidad que está siendo transferida.
- ✓ Request-Header: contiene información acerca de la solicitud y el cliente.
- ✓ Response-Header: contiene información sobre la respuesta.
- ✓ Entity-Header: contiene información acerca del recurso identificado por la solicitud e información sobre el cuerpo de la entidad.
- ✓ Entity-Body: el cuerpo del mensaje.

Todas las cabeceras de HTTP constan de una secuencia de campos que siguen el mismo formato genérico que el RFC 822 (descrito anteriormente). Cada campo comienza en una línea nueva y se compone del nombre del campo seguido por dos puntos y el valor del campo.

Aunque el mecanismo básico de transacción es simple, existe un gran número de campos y parámetros definidos en HTTP. Éstos se muestran en la Tabla 9.4.2.3 En el resto de esta sección examinaremos los campos de la cabecera general. En las siguientes secciones se describirán las cabeceras de solicitud, las cabeceras de respuesta y las entidades.

Tabla 9.4.2.2: Notación BNF utilizada por HTTP y URL

- Las palabras en minúsculas representan variables o nombres de reglas.
- Una regla tiene la forma

  nombre = definición

- DIGIT es cualquier dígito decimal; CRLF es el retorno de carro o salto de línea; SP representa uno o más espacios.
- Las comillas encierran texto literal
- Los ángulos «‹» «›», se pueden utilizar dentro de una definición para delimitar un nombre en aras de una mayor claridad.
- Los elementos separados por una barra «|» son alternativos.
- Los paréntesis ordinarios se utilizan para agrupar.
- El carácter «*» que preceda a un elemento indica repetición. La forma completa es

  ‹I›*‹J›elemento

  indicando al menos I y como mucho J ocurrencias del elemento. *elemento admite cualquier número de repeticiones, incluyendo 0; 1*elemento requiere al menos un elemento; 1*2elemento admite 1 o 2 elementos; ‹N›elemento significa exactamente N elementos.
- Los corchetes «[» «]», encierran elementos opcionales.
- La construcción «≠» se utiliza para definir, con el siguiente formato:

  ‹I›≠‹J›elemento

  indicando al menos I y como mucho J elementos, cada uno separado por una coma y espacios opcionales.
- Un punto y coma «;» a la derecha de una regla indica el comienzo de un comentario que continúa hasta el final de la línea.

Campos de la cabecera general

Los campos de la cabecera general pueden utilizarse en los mensajes de solicitud y de respuesta. Estos campos se aplican en ambos tipos de mensajes y contienen información que no se aplica directamente a la entidad que se está transfiriendo. Los campos son:

- ✓ Cache-Control (*"control de caché"*): especifica las directivas que ha de obedecer cualquier mecanismo que implemente una caché a lo largo de la cadena solicitud/respuesta. Su propósito es el de evitar que una caché interfiera de forma adversa sobre esta solicitud o respuesta concreta.
- ✓ Connection (*"conexión"*): contiene una lista de palabras clave y nombres de campos de cabecera que sólo se aplican a esta conexión TCP entre el emisor y el receptor más cercano que no sea un túnel.
- ✓ Date (*"Fecha"*): fecha y hora en la que se originó el mensaje.
- ✓ Forwarded (*"reenviado"*): utilizado por las pasarelas y representantes para indicar pasos intermedios a lo largo de la cadena de solicitud o respuesta. Cada pasarela o representante que procese un mensaje puede incorporar un campo «Forwarded» donde indique su URL.
- ✓ Keep-Alive (*"mantener activo"*): puede estar presente si existe la palabra clave *"keep-alive"* en un campo *"Connection"* recibido, para proporcionar información al solicitante sobre la conexión persistente. Este campo puede indicar la duración máxima que el emisor mantendrá abierta la conexión esperando la siguiente solicitud o el número máximo de solicitudes adicionales que se permitirán sobre la conexión persistente actual.
- ✓ MIME-Version (*"versión de MIME"*): indica que el mensaje cumple la especificación de la versión de MIME indicada.
- ✓ Pragma: contiene directivas específicas de implementación que pueden aplicarse a cualquier receptor a lo largo de la cadena de solicitud/respuesta.
- ✓ Upgrade (*"actualizar"*): se utiliza en una solicitud para especificar qué protocolos adicionales admite el cliente que querría utilizar. Se emplea en una respuesta para indicar qué protocolo será empleado.

Tabla 9.4.2.3: Todos los mensajes HTPP

TODOS LOS MENSAJES

Campos de la cabecera general	**Campos de la cabecera de entidad**
Cache-Control («control de caché») *Connection* («conexión») *Date* («fecha») *Forwarded* («reenviado») *Keep-Alive* («mantener activo») *MIME-Version* («versión de MIME») *Pragma* *Upgrade* («actualizar»)	*Allow* («admitir») *Content-Encoding* («esquema de codificación del contenido») *Content-Language* («lenguaje del contenido») *Content-length* («longitud del contenido») *Content-MD5* («MD5 del contenido») *Content-Range* («rango del contenido») *Content-Type* («tipo de contenido») *Content-Version* («versión del contenido») *Derived-From* («derivada de») *Expires* («expiración») *Last-Modified* («última modificación») *Link* («enlace») *Title* («título») *Transfer-Encoding* («esquema de codificación de transferencia») *URL-Header* («cabecera de URL») *Extension-header* («cabecera de extensión»)

MENSAJES DE SOLICITUD

Métodos de solicitud	**Campos de la cabecera de solicitud**
OPTIONS («opciones») *GET* («obtener») *HEAD* («cabecera») *POST* («enviar») *PUT* («poner») *PATCH* («parchear») *COPY* («copiar») *MOVE* («mover») *DELETE* («eliminar») *LINK* («enlazar») *UNLINK* («desenlazar») *TRACE* («trazar») *WRAPPED* («empaquetado») *extension-method* («método de extensión»)	*Accept* («aceptar») *Accept-Charset* («aceptar conjunto de caracteres») *Accept-Encoding* («aceptar esquema de codificación») *Accept-Language* («aceptar lenguaje») *Authorization* («autorización») *From* («de») *Host* («computador») *if-Modified-Since* («si ha sido modificado desde») *Proxy-Authorization* («autorización del representante») *Range* («rango») *Referrer* («remitente») *Unless* («a menos que») *User-Agent* («agente de usuario»)

MENSAJES DE RESPUESTA

Códigos de estado de respuesta	**Campos de la cabecera de respuesta**
Continue («continuar») *Switching Protocol* («cambiando de protocolo») *OK* («correcto») *Created* («creado») *Accepted* («aceptada») *Non-Authoritative Information* («información no acreditada») *No Content* («sin contenido») *Reset Content* («reiniciar contenido») *Partial Content* («contenido parcial»)	*Location* («localización») *Proxy-Authenticate* («autenticación del representante») *Public* («Público») *Retry-After* («intentar después de») *Server* («servidor») *WWW-Authenticate* («autenticar WWW»)

MENSAJES DE RESPUESTA

Códigos de estado de respuesta

Moved Temporarily («trasladado temporalmente») *See Other* («probar otro») *Not Modified* («no modificado») *Use Proxy* («utilizar representante») *Bad Request* («solicitud incorrecta») *Unauthorized* («desautorizado») *Payment Required* («pago requerido») *Forbidden* («prohibido») *Not Found* («no encontrado») *Method Not Allowed* («método no admitido») *None Acceptable* («ninguno aceptable») *Proxy Authentication Required* («autenticación de representante requerida»)	*Request Timeout* («expiración de solicitud») *Conflict* («conflicto») *Gone* («desaparecido») *Length Required* («longitud requerida») *Unless True* («condición verdadera») *Internal Server Error* («error interno del servidor») *No Implemented* («no implementada») *Bad Gateway* («error en pasarela») *Server Unavailable* («servidor no disponible») *Gateway Timeout* («expiración de pasarela») *extension code* («código de extensión»

9.4.3. Flujo de HTTP

Cuando el cliente quiere comunicarse con el servidor, tanto si es directamente con él, o a través de un proxy intermedio, realiza los siguientes pasos:

1. Abre una conexión TCP: la conexión TCP se usará para hacer una petición, o varias, y recibir la respuesta. El cliente puede abrir una conexión nueva, reusar una existente, o abrir varias a la vez hacia el servidor.
2. Hace una petición HTTP (Figura 9.4.3.1): Los mensajes HTTP (previos a HTTP/2) son legibles en texto plano. A partir de la versión del protocolo HTTP/2, los mensajes se comprimen, haciendo que no sean directamente interpretables, aunque el principio de operación es el mismo.

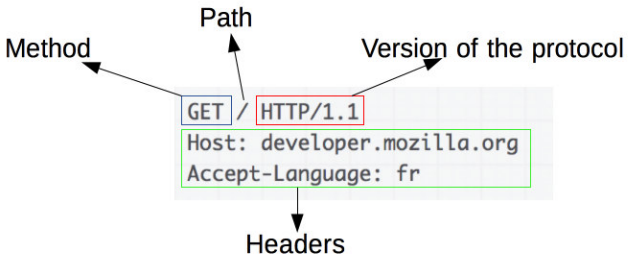


Figura 9.4.3.1: Petición HTTP

Una petición de HTTP, está formada por los siguientes campos:

- ✓ Un método HTTP, normalmente pueden ser un verbo, como: GET, POST o un nombre como: OPTIONS (en-US) o HEAD (en-US), que defina la operación que el cliente quiera realizar. El objetivo de un cliente, suele ser una petición de recursos, usando GET, o presentar un valor de un formulario HTML, usando POST, aunque en otras ocasiones puede hacer otros tipos de peticiones.
- ✓ La dirección del recurso pedido; la URL del recurso, sin los elementos obvios por el contexto, como pueden ser: sin el protocolo (http://), el dominio (aquí developer.mozilla.org), o el puerto TCP (aquí el 80).
- ✓ La versión del protocolo HTTP.
- ✓ Cabeceras HTTP opcionales, que pueden aportar información adicional a los servidores.
- ✓ Un cuerpo de mensaje, en algún método, como puede ser POST, en el cual envía la información para el servidor.

3. Lee la respuesta enviada por el servidor (Figura 9.4.3.2):

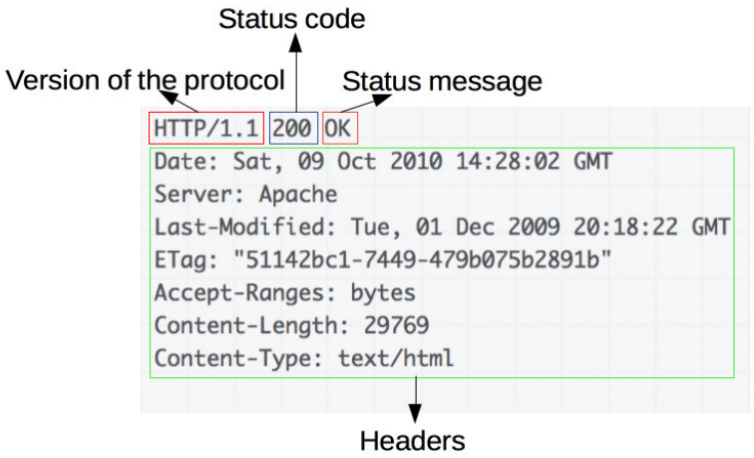


Figura 9.4.3.2: Respuesta HTTP

Las respuestas están formadas por los siguientes campos:

- ✓ La versión del protocolo HTTP que están usando.
- ✓ Un código de estado, indicando si la petición ha sido exitosa, o no, y debido a que.
- ✓ Un mensaje de estado, una breve descripción del código de estado.
- ✓ Cabeceras HTTP, como las de las peticiones.
- ✓ Opcionalmente, el recurso que se ha pedido.

4. Cierre o reutilización de la conexión para futuras peticiones.

Si está activado el HTTP pipelining[2], varias peticiones pueden enviarse sin tener que esperar que la primera respuesta haya sido satisfecha. Este procedimiento es difícil de implementar en las redes de computadores actuales, donde se mezclan software antiguos y modernos. Así que el HTTP pipelining ha sido substituido en HTTP/2 por el multiplexado de varias peticiones en una sola trama.

[2] Pipelining HTTP es una técnica donde se envían peticiones sin esperar respuestas, su implementación significó grandes mejoras en el tráfico HTTP 1 y 1.1, sobre todo en las conexiones satelitales.