Protocolo DHCP

Redes

UNIVERSIDAD SIGLO 21 | MIEMBRO DE LA RED ILUMNO

Protocolo DHCP

Todo dispositivo que forma parte de una red debe tener asignada una dirección IP para poder comunicarse en el nivel 3. Cuando se utiliza el protocolo IPv4, las direcciones IP pueder ser públicas o privadas. Las públicas son aquellas que pueden enrutarse en Internet. En ambos casos, un administrador de red debe asignar las direcciones correctamente a los dispositivos de cada red.

La asignación de direcciones puede ser manual o automática. Cuando se realiza de forma manual, cada dispositivo debe configurarse a mano con su dirección IPv4 y su máscara de subred. Otros parámetros son la puerta de enlace predeterminada y los servidores DNS que debe utilizar el dispositivo para comunicarse con otros fuera de su red o resolver nombres de dominio.

Cuando las redes son de tamaño reducido o cuando los dispositivos deben tener la misma dirección IP permanentemente, la asignación manual es posible; en cambio cuando las redes son de gran tamaño o cuando los dispositivos que acceden rotan de forma continua, es necesario implementar un método automático de asignación de direcciones. De esto último se encarga el protocolo DHCP (Dynamic host configuration protocol ó Protocolo de configuración dinámica de host).

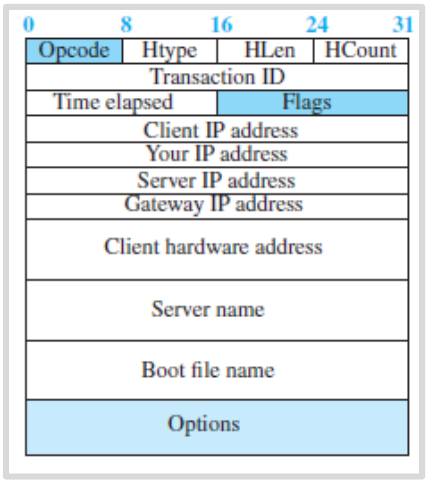
Formato de los mensajes DHCP

El protocolo DHCP pertenece a la capa de aplicación. Como todo protocolo de redes, está formado por un encabezado que contiene diferentes campos los cuales realizan diferentes funciones.

En la figura 1 se observa el formato de encabezado DHCP con sus respectivos campos. Los principales campos se explican a continuación.

- Opcode: es el código de operación que diferencia una solicitud de una respuesta.
- Transaction ID: identifica al cliente que solicita una dirección IP y diferencia de otras solicitudes
- Client IP Address: cuando el cliente no tiene dirección IP, este campo es puesto en cero.
- Your IP Address: La dirección IP enviada por el servidor
- Options: campo de 64 bits con dos propósitos

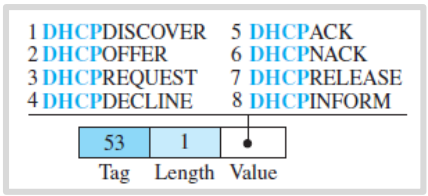
Figura 1: Formato de mensaje DHCP



Fuente: Forouzan, 2013, p. 540

Cuando el campo options lleva en la denominada “Magic Cookie” el número 99.130.83.99, el cliente interpreta que los siguientes 60 bytes son opciones. Estas opciones están compuestas por 3 campos como se observa en la figura 2. Cuando el campo Tag tiene el valor 53, significa que se utilizará alguno de los 8 mensajes DHCP. Otros Tags son utilizados por los fabricantes.

Figura 2: Options



Fuente: Forouzan, 2013, p. 540

Operación DHCP

El objetivo del protocolo es asignar direcciones IP a los clientes que la soliciten. Para que esto sea posible, un servidor DHCP debe estar configurado. La operación consiste en el intercambio de mensajes entre el cliente y el servidor como se observa en la figura 3.

Paso 1: mensaje DHCPDISCOVER

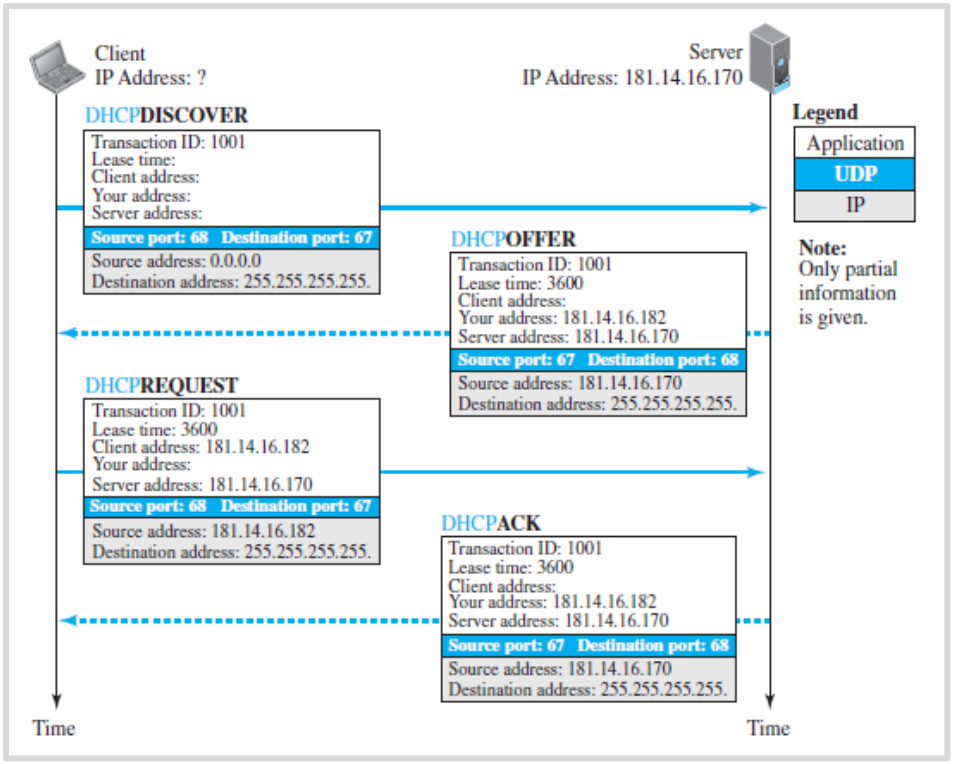
Cuando un cliente inicia el procedimiento de obtensión de dirección IP envía un mensaje DHCPDISCOVER con un número de transacción aleatorio. Cómo el cliente no tiene dirección IP ni conoce la dirección del servidor, utiliza la dirección IP de destino 255.255.255.255 (broadcast) y la dirección IP de

origen 0.0.0.0. Antes de encapsularse en un datagrama IP, el mensaje DHCP es primero encapsulado en un datagrama de usuario UDP utilizando el puerto de origen 68 y el puerto de destino 67.

Paso 2: mensaje DHCPOFFER

Cuando un servidor DHCP recibe un paquete IP broadcast cuyo datagrama de usuario UDP tiene como puerto de destino 67, y cuyo mensaje DHCP es un DHCPDISCOVER lo procesa y responde con un mensaje DHCPOFFER.
El mensaje DHCP contendrá ahora una dirección IP ofrecida para el cliente y la dirección IP del serivodor. Sin embargo, el paquete IP tendrá como dirección IP de destino 255.255.255.255 ya que el cliente aún no tiene asignada una dirección IP e igualmente debe recibir el mensaje. Los puertos utilizados en el datagrama de usuario UDP son los mismos que en el mensaje anterior, pero invertidos.

Figura 3: Operación DHCP



Fuente: Forouzan, 2013, p. 541

Paso 3: mensaje DHCPREQUEST

Un cliente puede recibir más de una oferta y puede elegir la que considere mejor entre todas. Para solicitar la oferta más conveniente, utiliza el mensaje DHCPREQUEST. En este caso, cuando el mensaje se encapsula en

IP, el cliente ya utiliza la dirección de origen que le ofreció el servidor DHCP, pero continúa utilizando la dirección IP de destino broadcast para que el resto de los servidores DHCP (si los hay) se enteren que su oferta fue rechazada.

Paso 4: DHCPACK

El paso final consiste en la confirmación o no por parte del servidor de la dirección IP. Cuando el servidor confirma utiliza un mensaje DHCPACK; cuando rechaza la oferta el mensaje es un DHCPNACK. Ambos son enviados utilizando la IP de destino broadcast para que el resto se entere.

Lease time

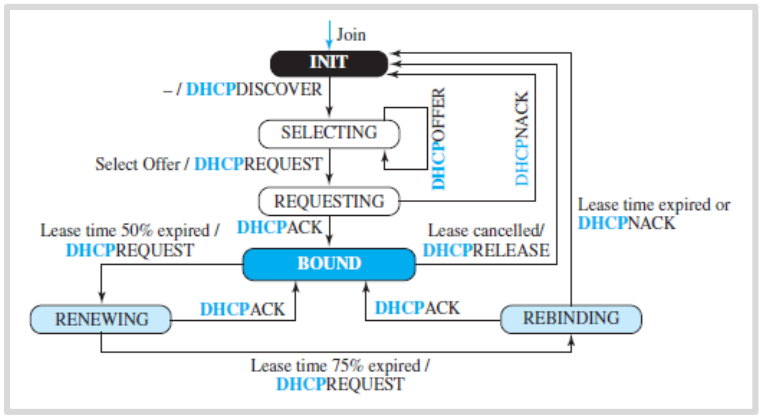
DHCP permite 3 tipos de asignaciones: estáticas, automáticas y dinámicas. En una asignación automática, el administrador configura al servidor para que asigne siempre la misma dirección IP a un determinado host cuando este se conecta a la red.
En una asignación dinámica, el cliente obtiene una dirección IP ofrecida por el servidor y la puede conservar por cierta cantidad de tiempo. La cantidad de tiempo se denomina Lease Time y es un parámetro configurable.

En este momento seguramente surge la pregunta ¿Cuánto tiempo se debe configurar? La respuesta es que dependerá de cada situación. Si utilizamos asignación dinámica en un hogar con cierta cantidad de dispositivos como por ejemplo computadoras, celulares y tablets, el tiempo puede ser grande ya que no hay mucha rotación de dispositivos y la cantidad de direcciones disponibles es suficiente.
En cambio si la asignación de direcciones se produce en una red pública, por ejemplo un campus universitario o aeropuerto, la rotación va a ser muy alta. Las personas permanece cierta cantidad de tiempo (no mucho mas de algunas horas) y se van. Si asignamos direcciones IP por 24 horas en esta red, muchas direcciones IP no estarán disponibles al estar prestadas a dispositivos que ya no las están utilizando.

¿Cómo se controla el Lease time? En la figura 4 se observan diferentes estados por lo que pasa un cliente DHCP.

Una vez que el cliente obtiene una dirección IP pasa al estado BOUND. El cliente activa un timer y cuando llega al 50% del Lease time que le asignó el servidor, deberá enviar un DHCPREQUEST solicitando renovación de tiempo al servidor. Este último puede confirmar la renovación con un DHCPACK volviendo el timer del cliente a cero. Este proceso continuará indefinidamente hasta que el cliente decida cancelar el préstamo enviando un mensaje DHCPRELEASE.

Figura 4: Estados en un cliente DHCP



Fuente: Forouzan, 2013, p. 543

En el caso de que el servidor no responda, el timer continúa hasta el 75% del lease tiempo. En este momento, el cliente cambia al estado REBINDING y podrá volver al estado BOUND si el servidor confirma con un DHCPACK. En caso contrario, volverá al estado inicial y deberá solicitar otra dirección IP.

Cambio de lease time en servidores DHCP

Para cambiar el Lease Time en un servidor DHCP implementado en Router Cisco, utilizar el comando lease dentro de la configuración dhcp-config. El lease time por defecto es de 1 día.

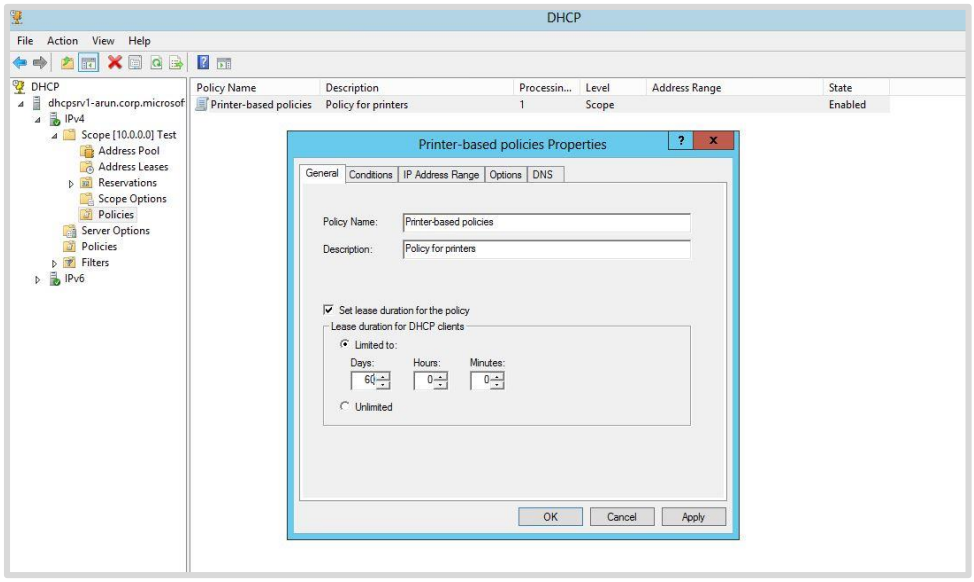
```
Router(dhcp-config)# lease {days[hours][minutes] | infinite}
```

Para cambiar el Lease Time en un Router Mikrotik utilizar el siguiente comando:

/ip dhcp-server set dhcp1 lease-time=

En Microsoft Windows 2012, es possible definer políticas para asignar diferentes Lease Time. En el ejemplo de la figura 5, se asigna un lease time de 60 días para impresoras.

Figura 5: Lease time en Microsoft Windows 2012



Fuente: Imagen sin título sobre lease time. Recuperado de: https://blogs.technet.microsoft.com/teamdhcp/2012/09/22/using-dhcp-policies-to-set-different-lease-durations-for-different-device-types/

Referencias

Forouzan, B (2013). Network Layer en *Data Communications AND Networking.* Estados Unidos: McGraw-Hill

Comer, D (2013). Boostrap and Autoconfiguratoin en *Internetworking with TCP/IP.* Estados Unidos: Pearson

IETF (1997). RFC2131. Dynamic Host Configuration Protocol. Estados Unidos. https://tools.ietf.org/html/rfc2131

Resolución de nombres

Redes

UNIVERSIDAD SIGLO 21 | MIEMBRO DE LA RED ILUMNO

Resolución de nombres de dominio

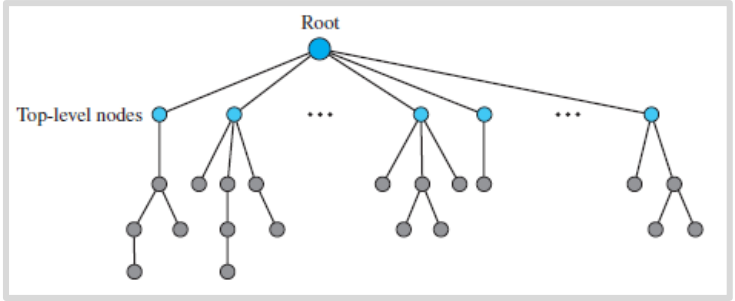
Para que dos dispositivos se comuniquen entre si se requiere una dirección IP de origen y otra de destino. Para utilizar servicios de Internet, por ejemplo el servicio de páginas web, las personas deberían recordar cuál es la dirección IP de cada servidor donde se almacena la página web que desea visitar.

Esto podría resultar molesto pero viable si solo existieran un par de sitios web que visitar. Actulamente existen millones de sitios y recordar todas las direcciones IP no tiene el menor sentido para las personas. Para ellos se ideo el protocolo DNS, el cual resuelve el inconveniente asociados nombres reconocibles por las personas con la dirección IP correspondiente.

Jerarquía de nombres

En Internet, los nombres no pueden estar repetidos ya que se generaría confusión y problemas para determinar cuál es realmente el servidor al que se desea acceder. Por cada dirección IP habrá un nombre único. En Internet se utiliza una jerarquía de nombres para poder descentralizar el control. En la figura 1 se observa la estructura en donde un servidor de nombre raiz se divide en denominados “top level domains”, los cuales a su vez se subdividen en los diferentes nombres utilizados por empresas, universidades, particulares, organizaciones, etc.
Cada top level domain server es encargado de administrar solo los nombres bajo su órtibta. Por ejemplo un topo level domain “edu”, solo es encargado de gestionar los nombres que terminen con .edu y no deberá preocuparse por el resto (.com, .org, .gov, .ar, etc).

Figura 1: Estructura jerárquica



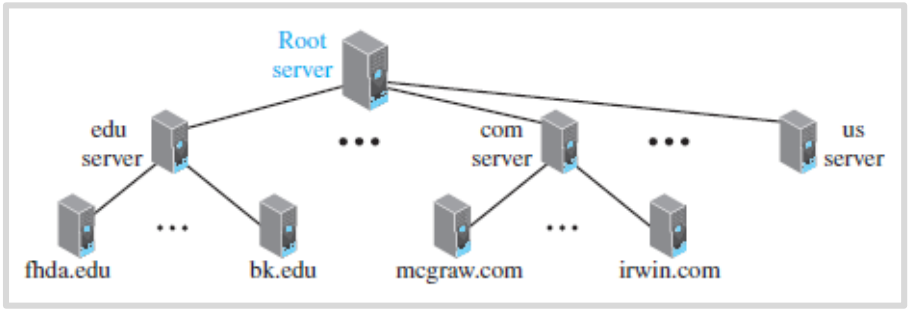
Fuente: Forouzan, 2013, p. 912

Si bien los nombres no pueden repetirse bajo un nivel de la jerarquía, si podrían hacerlo en niveles inferiores. Por ejemplo, los nombres verde.empresa1.com y verde.empresa2.com son válidos y únicos, ya que verde en cada caso es un subdominio de los dominios empresa1.com y empresa2.com.
El máximo de niveles permitidos es de 128 siendo 0 el nivel root. Las palabras que se utilizan en cada nodo se denominan label o etiquetas y tienen un máximo de 63 caracteres.
Un nombre de dominio es una secuencia de labels separados por un punto. Así el nombre ues21.edu.ar está formado por 3 labels separados por puntos. El top level domain es "ar", el siguiente label es "edu" y luego "ues21". Luego dentro de ues21 podrían continuar creandose subdominios, por ejemplo pc128.laboratorio1.ues21.edu.ar.

Almacenamiento de la información

Debido a la gran cantidad de información que debería manejar un servidor DNS (todos los dominios del mundo) se ha optado por una estructura descentralizada como la que se observa en la figura 2.

Figura 2: Servidores DNS



Fuente: Forouzan, 2013, p. 914

Cada servidor de la estructura es responsable de gestionar información de su dominio. Por ejemplo el servidor "edu server" solo posee información sobre los dominios del top level edu. Bajo su órbita hay además dos servidores: fhda.edu y bk.edu. Cada uno de estos dos servidores maneja informació sobre los subdominios fhda y bk respectivamente.
Si alguien ubicado en la red de bk.edu quisiera acceder a una computadora ubicada en mcgraw.com, la consulta deberá se redireccionada hacia el servidor mcgraw.com a través del "com server".

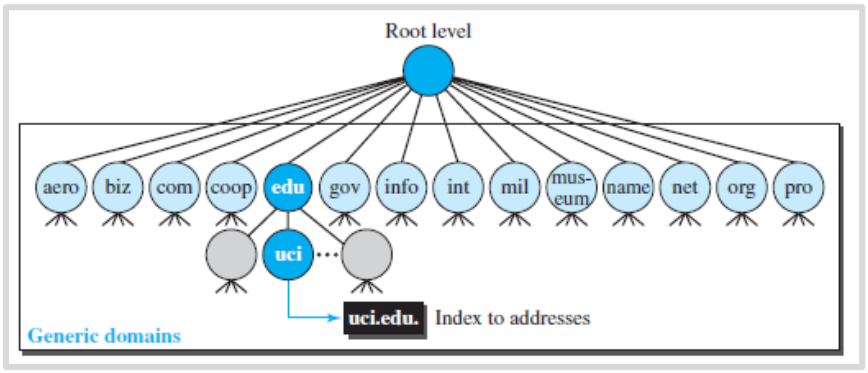
El servidor root es el que conoce cuales son todos los top level domains y de esta forma puede indicar a donde deben realizarse las consultas, pero no posee información sobre nombres ya que delega ese trabajo en los top level domain servers.

Aunque en la imagen 2 se observe solo un servidor DNS, existen 13 servidores root diferentes nombrados con letras (a.root-server.net , b,root-server.net, etc). A su vez, existen múltiples servidores distribuidos por todo el mundo por cada root server.
A modo de ejemplo, en la ciudad de Córdoba, Argentina, hay 3 servidores root pertenecientes a j.root-server.com, d.root-server.com y e.root-server.com

Dominios en Internet

En la figura 3 se observan diferentes dominios utilizados en Internet. Cada tipo de dominio describe el tipo de organización que lo utiliza. Por ejemplo los dominio .edu son utilizados por organizaciones dedicadas a la educación. Los .gov para organismos gubernamentales.
Además de estos dominios, existe un top level domain por cada país del mundo. Para argentina es .ar, para brasil .br.

Figura 3: Dominios



Fuente: Forouzan, 2013, p. 915

Resolución de nombres

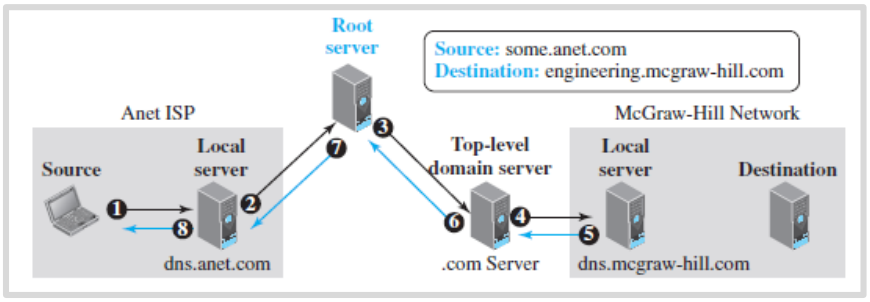
Resolver un nombre significa obtener la dirección IP. Un cliente que necesita resolver un nombre en una IP o viceversa debe acceder al servidor DNS más próxima para poder cumplir su objetivo. El servidor DNS podrá cumplir la tarea o requerir de otro servidor DNS.
Las resoluciones pueden ser recursivas o iterativas.

Resolución recursiva

Para entender la resolución recursiva analicemos el ejempo de la figura 4. El dispositivo some. anet.com quiere obtener la IP del dispositovivo engineering.mcgraw-hill.com para poder enviarle un mensaje. La aplicación dentro de “some” deberá llamar al cliente DNS para que consulte al servidor DNS, en este caso dns.anet.com.

Como el servidor dns.anet.com no conoce la IP del host de destino, envía una solicitud al Root Server. Lo servidores root no tienen mapeos nombre-IP pero si conocen los servidores top level domain, por lo que envia una solicitud al top level .com.
Como el servidor .com tampoco conoce la IP del nombre solicitado envía la consulta al servidor DNS de mcgraw-hill quién finalmente es el que conoce qué IP tiene el host engineering. La respuesta seguirá el mismo camino hasta llegar al host “some” (pasos 5, 6, 7 y 8) quién de esta forma pudo obtener la dirección IP que necesitaba y podrá enviar su mensaje.

Figura 4: Resolución recursiva

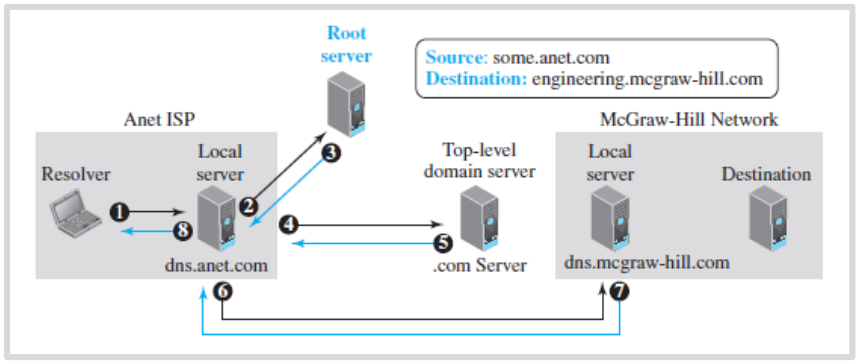


Fuente: Forouzan, 2013, p. 917

Resolución iterativa

Las resolucione iterativas cumplen el mismo objetvo que las recursivas pero trabajan de otra manera. Cuando los servidores no conocen la dirección IP del nombre solicitado pero si conocen la IP del siguiente servidor que debería consultarse, devuelven esa información al servidor DNS que inició la consulta, el cual luego utiliza dicha información para enviar la solicitud al siuguiente servidor.
En la figura 5 se muestra el mismo escenario que el utilizado para la resolución recursiva.
En el paso 2, el servidor DNS local consulta al Root server quien le devuelve la IP del top level .com.
En el paso 4, el servidor DNS local consulta al top level .com quien le devuelve la IP del servidor DNS de McGraw-Hill.
Finalmente, el servidor DNS localenvía la consulta al servidor DNS dns.mcgraw-hill y obtiene como respuesta la IP que necesitaba.

Figura 5: Resolución iterativa



Fuente: Forouzan, 2013, p. 918

Tipos de registros

Los servidores DNS almacenan información en una base de datos sobre los nombres de dominio utilizando registros de recursos. Cada registro de recursos se compone de:

- Nombre de dominio: identifica al registro
- Tipo: indica cómo debe interpretarse el valor.
- Clase: define el tipo de red (IN para Internet)
- TTL: indica por cuantos segundos la información es válida
- Valor: indica la información almacenada sobre el nombre

Los tipos más comunes son los A (dirección IPv4 de host), AAAA (dirección IPv6 de host), MX (Mail Exchange), NS (Servidor de nombres). Para conocer la toalidad de los tipos, consultar la RFC1034 y la bibliogafía básica.

Referencias

Tanenbaum, A (2012). La capa de aplicación en Redes *de Computadoras.* Madrid: Editorial Pearson Education

Forouzan, B (2013). Application Layer en *Data Communications AND Networking.* Estados Unidos: McGraw-Hill

IETF (1987). RFC1034. DOMAIN NAMES - CONCEPTS AND FACILITIES. Estados Unidos. https://tools.ietf.org/html/rfc1034

IETF (1987). RFC1035. DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION. Estados Unidos. https://tools.ietf.org/html/rfc1035

Correo electrónico

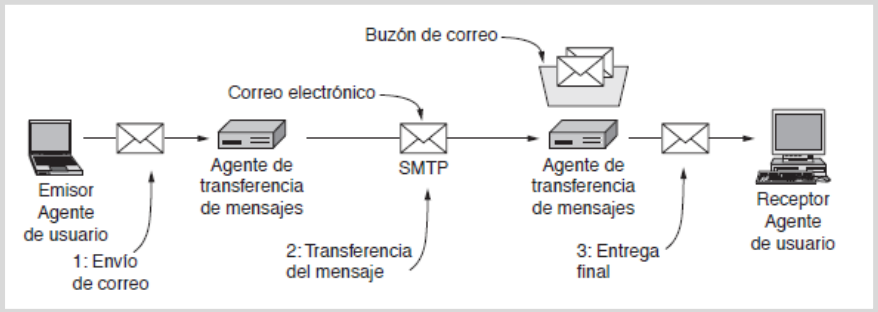
Redes

UNIVERSIDAD SIGLO 21 | MIEMBRO DE LA RED ILUMNO

Correo electrónico

El correo electrónico es uno de los servicios más importantes de la capa de aplicación. Su uso ha revolucionado las comunicaciones empresariales y personales relegando al correo tradicional.
En los comienzos de Internet, el correo electrónico o e-mail era la aplicación por excelencia para comunicación entre personas; si bien actualmente existen innumerables métodos como aplicaciones de mensajería instantánea, blogs o redes sociales, la mayoría requiere que las personas se registren y para ello siempre es requerida una dirección de correo electrónico, por lo que la continuidad del servicio está garantizada. Además, el correo electrónico no tiene como objetivo ser un servicio en tiempo real a diferencia de otras aplicaciones actuales lo que le brinda mas flexibilidad en las comunicaciones a las personas que lo utilizan.

Figura 1: Arquitectura de un sistema de correo electrónico



Fuente: Tanenbaum, 2012, p. 536

En la figura 1 se observa la arquitectura básica de un sistema de correo electrónico. El emisor, denominado agente de usuario utiliza una aplicación de correo, que puede estar instalada en su dispositivo o disponible en una página de Internet, para enviar su correo a un agente de transferencia de mensajes también denominado comúnmente servidor de correo.

Son ejemplos de agentes de usuarios las aplicaciones instalables Microsoft Outlook o Mozila Thunderbird; en el caso de correo web, los más utilizados son Gmail, Outlook y Yahoo.

Para transferir mensajes de correo entre un agente de usuario y un servidor de correo, o entre servidores de correo el protocolo que se utiliza actuamente se llama SMTP (Simple Mail Transfer Protocol o Protocolo simple de transferencia de correos). Para la entrega final, los protocolos utilizados son POP e IMAP.

Protocolo SMTP

Este protocolo define como ciertos comandos y respuestan deben manejarse entre dos dispositivos. Los comandos se envían desde un cliente que puede ser un agente de usuario o un agente de transferencia de correo. En el otro extremo se ubicará un agente de transferencia de correo servidor que enviará respuestas a los comandos recibidos.

En la tabla 1 se pueden apreciar 3 comandos de SMTP. Para conocer la lista completa, leer la RFC821, sección 4.1

Tabla: Comandos SMTP

Palabra Clave	Argumento	Descripción
HELO	Nombre del host del remitente	Nombre del host del transmisor
MAIL FROM	Remitente del mensaje	Identifica al remitente
DATA	Cuerpo del mensaje	Envía el mensaje real

Fuente: adaptado de Forouzan, 2013.

Para enviar un correo electrónico, la aplicación del emisor se intentará conectar al servidor utilizando el puerto 25 TCP. Luego de establecida la coneixón, el servidor envía un código 220 indicando que está listo o 421 en caso de que el servicio no esté disponile.
Si está listo el cliente envía un comando HELO informando el dominio del remitente. La respuesta 250 confirma que el comando fue completado.
Para enviar un mensaje, el cliente enviará el comando MAIL FROM con su dirección de correo; luego de la confirmación del servidor enviará un RCPT TO indicando la dirección del destinatario y finalmente un DATA para iniciar la transferencia. Estos últimos dos pasos se repiten por cada destinatario.
A partir de este momento el cliente enviará el mensaje el líneas consecutivas.

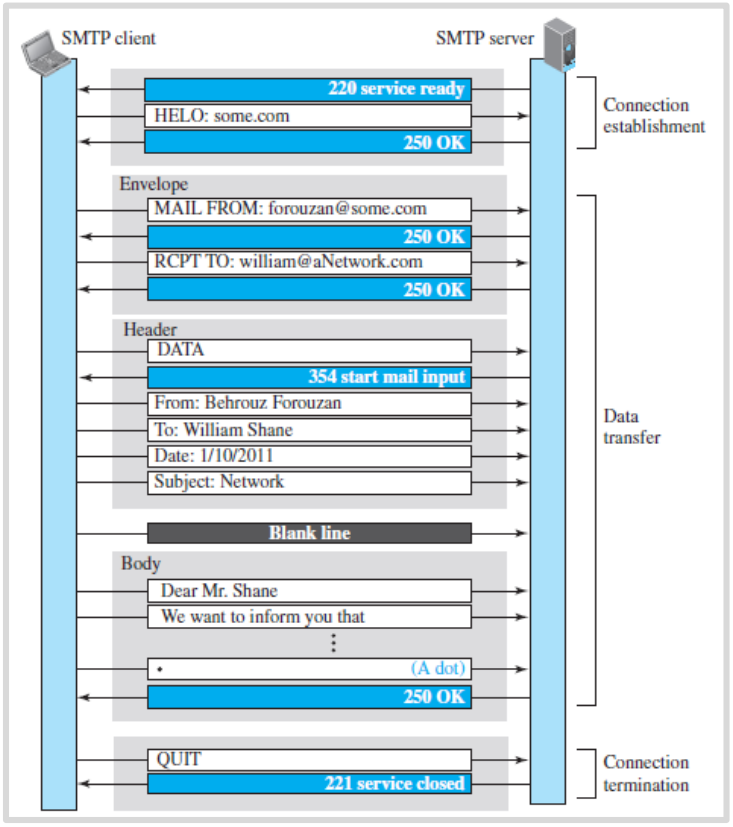
Para terminar la conexión, el cliente envía un comando QUIT y el servidor código 221. En la figura 2 puede verse un ejemplo de las 3 fases involucradas en el envío de un correo electrónico.

Las respuestas más comunes de un servidor SMTP son:

- 220: servicio listo
- 250: petición de comando completada
- 354: comienza el ingreso de correo
- 421: servicio no disponible
- 450: casilla de correo no disponible

- 500: error de sintaxis

Figura 2: Ejemplo de las 3 fases para enviar un correo



Fuente: Forouzan, 2013, p. 898

ESMTP

La extensión SMTP (Extension SMTP) permite la autenticación de los clientes en el servidor de correo SMTP. Esto es súmamente útil ya que en caso de no utilizarlo, los spammers pueden utilizar los serivores para enviar correo basura utilizando cualquier dirección de origen.

Para profundizar tus conocimientos sobre ESMTP consulta los RFC2554 y RFC4409.

Entrega final del mensaje

El protocolo SMTP analizado hasta el momento cumple con la función de transferir mensajes entre servidores, o enviar un mensaje desde el agente de usuario al servidor. SMTP es un protoclo “push”, empuja mensajes. Pero el destinatario necesita extraer sus mensajes desde un servidor, y para ellos se utilizan protocolso “pull” (tirar). Actualmente los protocolos utilizados son POP3 y IMA4.

Protocolo POP3

El protocolo POP3 (Post Office Protocol) es simple aunque sus funciones son muy limitadas. El cliente que necesita descargar sus correos se conecta al servidor POP3 mediante una conexión TCP al puerto 100. Envía su usario y clave al servidor para autenticación para luego desargar los correos desde el servidor a su dispositivo.
POP3 ofrece dos modos: delete (borra los mensajes del servidor una vez que son descargados) y keep (los deja almacenados).

A continuación se muestra un ejemplo de la interacción entre el cliente (c) y el servidor (s) en donde se descargan 2 mensajes.

S: <wait for connection on TCP port 110>
C: <open connection>
S: +OK POP3 server ready <1896.697170952@dbc.mtview.ca.us>
C: APOP mrose c4c9334bac560ecc979e58001b3e22fb
S: +OK mrose's maildrop has 2 messages (320 octets)
C: STAT
S: +OK 2 320
C: LIST
S: +OK 2 messages (320 octets)
S: 1 120
S: 2 200
S: .
C: RETR 1
S: +OK 120 octets
S: <the POP3 server sends message 1>
S: .
C: DELE 1
S: +OK message 1 deleted
C: RETR 2
S: +OK 200 octets
S: <the POP3 server sends message 2>
S: .
C: DELE 2
S: +OK message 2 deleted
C: QUIT
S: +OK dewey POP3 server signing off (maildrop empty)
C: <close connection>
S: <wait for next connection>

Protocolo IMAP4

El protocolo POP3 si bien es simple tiene limitaciones. No es posible por ejemplo organizar los correos en carpetas en el servidor. Tampoco es posible

disponer de visas previas de los mensajes antes de descargarlos. Estas funcionalidades si están presentes en el protocolo IMAP4 (Internet Mail Access Protocol version 4).

Protocolos propietarios

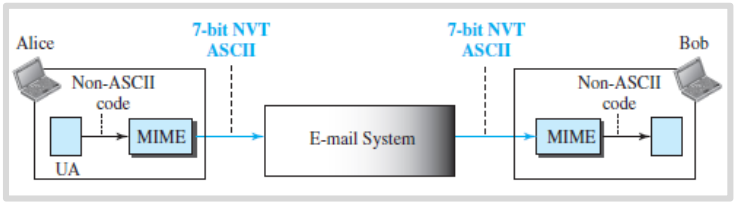
Ejemplos de protocolos propietarios de correo son Microsoft Exchange Server o IBM Lotus Notes.

Formato de los mensajes

La estructura de los mensajes de correo electrónico solo admite formato ASCII de 7 bits. Esta limitación no permite caracteres utilizados por ejemplo en nuestro idioma como acentos o la letra ñ. Para que sea posible utilizar más caracteres y además enviar archivos binarios, se utiliza el protocolo suplementario denominado MIME (Multipurpose Internet Mail Extensions).

Lo que hace MIME es transformar los datos que no son ASCII en ASCII cuando se envía un correo, para luego realizar la conversión inversa en el receptor. El funcionamiento se explica gráficamente en la figura 3.

Figura 3: utilización de MIME



Fuente: Forouzan, 2013, p. 900

MIME define 5 encabezados los cuales pueden ser agregados al encabezado original del correo para definir los parámetros de transformación. Estos 5 encabezados son:

- MIME-Version: indica la versión utilizada siendo 1.1 la actual.
- Content-Type: define el tipo de datos en el cuerpo del mensaje
- Content-Transfer-Encoding: define el método utilizado para codificar
- Content-Id: identifica al mensaje completo en un ambiente multimensaje.
- Content-Description: define si el cuerpo es una imagen, audio o video.

Para conocer en detalle cada encabezado, consultar el capítulo 7 de Tanenbaum.

Referencias

Tanenbaum, A (2012). La capa de aplicación en Redes *de Computadoras.* Madrid: Editorial Pearson Education

Forouzan, B (2013). Application Layer en *Data Communications AND Networking.* Estados Unidos: McGraw-Hill

IETF (1982). RFC821. SIMPLE MAIL TRANSFER PROTOCOL. Estados Unidos. https://tools.ietf.org/html/rfc821

IETF (1999). RFC2554. SMTP Service Extension for Authentication. Estados Unidos. https://tools.ietf.org/html/rfc2554

IETF (1996). RFC1939. Post Office Protocol - Version 3. Estados Unidos. https://www.ietf.org/rfc/rfc1939.txt

World Wide Web

Redes

UNIVERSIDAD SIGLO 21 | MIEMBRO DE LA RED ILUMNO

World Wide Web

La capa de aplicación ofrece muchísimos servicios como accesos remotos, transferencia de archivos o correo electrónico, pero sin dudas el servicio de sitios web es el más importante y el que hizo crecer de forma exponencial a la red Internet.
Muchas personas utilizan el correo electrónico, muchas menos el servicio de transferencia de archivos y solo muy pocas el acceso remoto, pero la gran mayoría, por no decir todos, utilizamos la Web.

Arquitectura

Las páginas web a las que acceden los usuarios están formadas por vínculos a otras páginas en lo que se conoce como hipertexto. Estas páginas están almacenadas en lo que se denomina un servidor web. Del lado del cliente, la aplicación que hace posible visualizarlas se denomina browser o navegador siendo los más comunes Chrome, Firefox, Edge, Opera y Safari.
El navegador solicita una determinada página al servidor y obtiene como respuesta la página solicitada. El protocolo involucrado en las solicitudes y las respuestas se denomina HTTP (Hyper Text Transfer Protocol).

Para identificar las diferentes páginas a las que un cliente puede acceder, cada una de las páginas tiene asignado un URL (Localizador Uniforme de Recursos) el cual está compuesto por tres partes:

- El protocolo (http)
- El dominio de la página (www.ues21.edu.ar)
- Puerto
- Nombre de ruta donde (index.html)

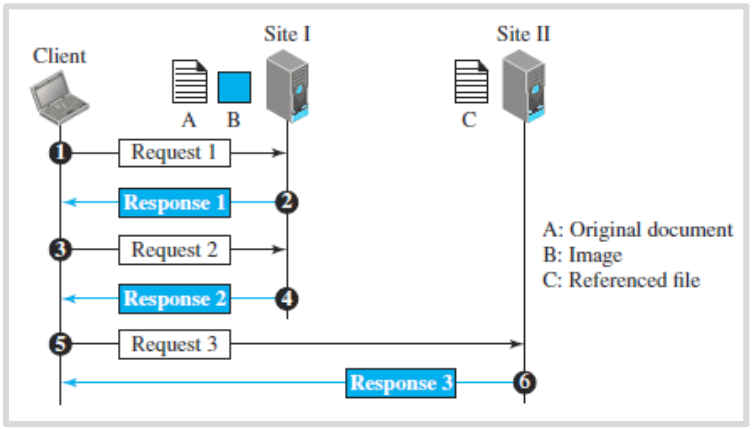
La URL en este caso será http://www.ues21.edu.ar/index.html
Cuando el puerto utilizado es el bien conocido TCP 80, no es necesario indicarlo en la URL. De lo contrario, deberá ser escrito. Por ejemplo si el servidor de Siglo 21 utiliza un puerto diferente, la URL será http://www.ues21.edu.ar:987/index.html

Para poder acceder a una determinada página el navegador deberá primero resolver el dominio mediante una consulta DNS. Una vez que obtiene la IP del servidor en donde se ubica la página, establecerá una conexión TCP al puerto 80 del servidor y acto seguido enviará una solicitud para que le envíen el archivo index.html.

Otros protocolos que pueden usarse en un browser son FTP, File, HTTPS.

La figura 1 grafica el proceso de solicitud de una página web, la cual puede tener información en más de un servidor: un cliente solicita un documento al servidor del sitio 1. Una vez que el servidor envía dicho documento, el cliente solicita una imagen al mismos servidor. Finalmente, y luego de obtener la imagen, el cliente solicita un archivo referenciado que se aloja en un segundo servidor.

Figura 1: Interacción con dos servidores web



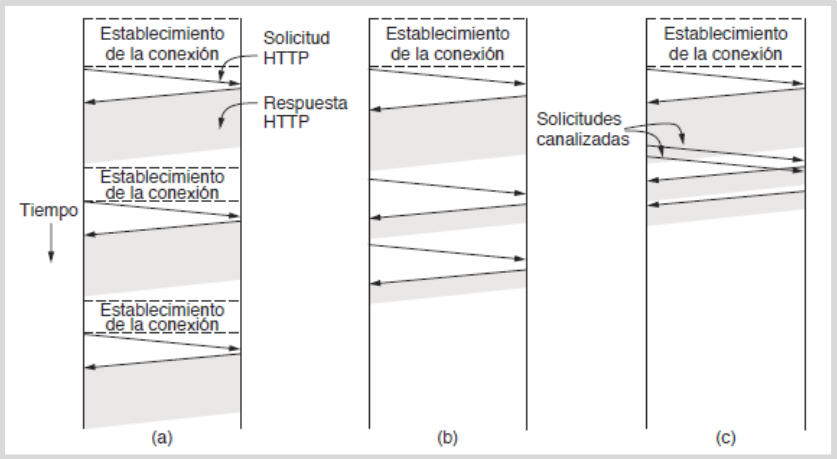
Fuente: Forouza, 2013, p. 873

Protocolo HTTP

El protocolo HTTP es muy simple y consiste en pares solicitud (enviado por el cliente) – respuesta (enviado por el servidor).
En su versión original (HTTP 1.0) la conexión TCP terminaba cada vez que el servidor enviaba una respuest a la solicitud del cliente. Esto funcionó cuando las páginas web eran muy simples, pero actualmente además de la página principal se requiere solicitar y recibir mucha mas información como imágenes, animaciones, videos, etc. Realizar una conexión TCP para cada objeto no es eficiente, por lo que la versión 1.1 utiliza conexiones persistentes que permiten un mayor número de solicitudes-respuestas. La última actualización del protocolo es la 2.0 y presenta nuevas mejoras con respecto a la velocidad de carga de las páginas.

En la figura 2 puede compararse los tiempos empleados para cada situación. En a), se realiza una conexión por cada solicitud. En el caso b), en una misma conexión se envían varias solicitudes y respuestas. Finalmente en c) es posible enviar varias solicitudes antes de recibir una respuesta, siempre utilizando la misma conexión TCP.

Figura 2: Tipos de conexiones en http



Fuente: Tanenbaum, 2012, p. 588

Métodos

HTTP tiene múltiples usos además de solicitar y recibir una página web. Por ejemplo el protocolo SOAP (Simple Object Access Protocol) utiliza mensajes xml para que dos programas puedan intercambiar información utilizando el protocolo HTTP. Cada operación (como por ejemplo solicitar una página web) se denomina método.

En la figura 3 se observan los métodos utilizados en el protocolo. Para solicitar el archivo index.html se utiliza el método GET de la siguiente manera (método, nombre de la página, versión del protocolo).

GET index.html HTTP/1.1

Figura 3: Métodos

Método	Descripción
GET	Leer una página web.
HEAD	Leer el encabezado de una página web.
POST	Adjuntar a una página web.
PUT	Almacenar una página web.
DELETE	Eliminar la página web.
TRACE	Repetir la solicitud entrante
CONNECT	Conectarse a través de un proxy
OPTIONS	Consultar las opciones para una página

Fuente: Tanenbaum, 2012, p. 590

Para conocer sobre el resto de los métodos consulta la bibliografía básica y la RFC2616.

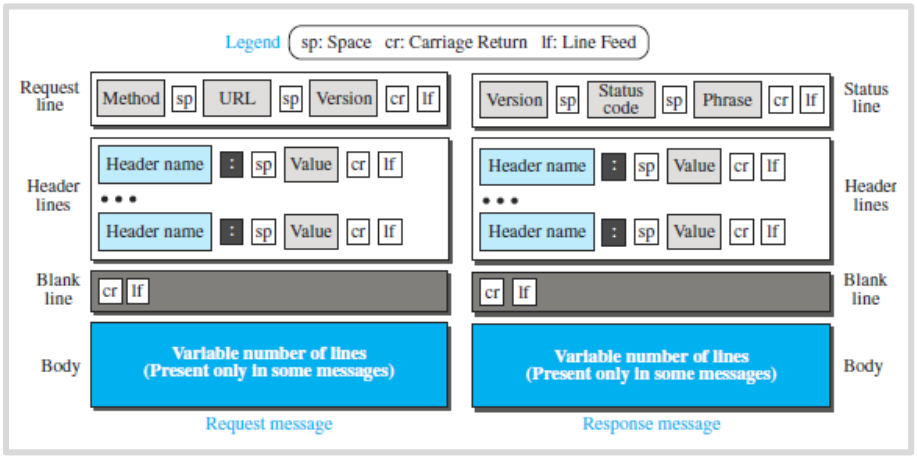
Respuestas

Cada vez que se envía una solicitud a un servidor, este debe responder al cliente utilizando alguno de los códigos preestablecidos. Este código es de 3 digitos y según el digito inicial será un código de información (1XX), éxito (2XX), redirección (3XX), error del cliente (4XX), o error del servidor (5XX).
Algunas de las respuestas más comunes son 100 indicando que el servidor acepta manejar la solicitud del cliente, 200 solicitud exitosa, 301 la página fue movida, 404 página no encontrada y 500 error interno del servidor.

Formatos

El formato de los mensajes HTTP se muestra en la figura 4. La primera línea se compone del método, un espacio, la URL solicitada, otro espacio, la versión del protocolo, un retorno de carro y nueva línea tanto para el mensaje de solicitud; en el caso de la respuesta, se indica la versión, un espacio, el código de status, otro espacio, la frase y un retorno de carro con comienzo de una nueva línea. Seguido de la línea de solicitud o status continúan las líneas de encabezado (aunque puede que no haya ninguna). Estas líneas envían información adicional del cliente al servidor, por ejemplo, solicitando que le envíen el documento en un formato especial.
Los tipos de encabezado pueden consultarse en la bibliografía básica o en la RFC2616.

Figura 4: Formato de mensajes HTTP



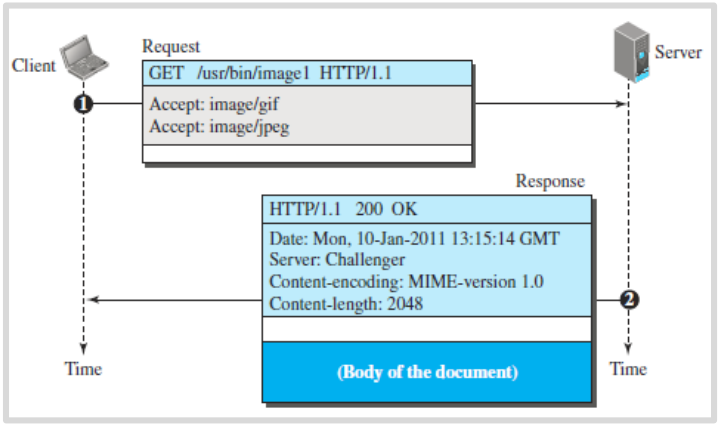
Fuente: Forouzan, 2013, p. 879

Ejemplos de solicitudes y respuestas

En el ejemplo 1 el cliente solicita una imagen mediante el método GET y utilizando cabeceras anuncia que acepta formatos .gif y .jpeg. La solicitud no

tiene ningun cuerpo (body). El servidor responde con código 200, cuatro líneas de header donde indica la fecha, el servidor, la codificación del contenido y la longiutd.

Figura 5: Ejemplo 1 solicitud de imagen

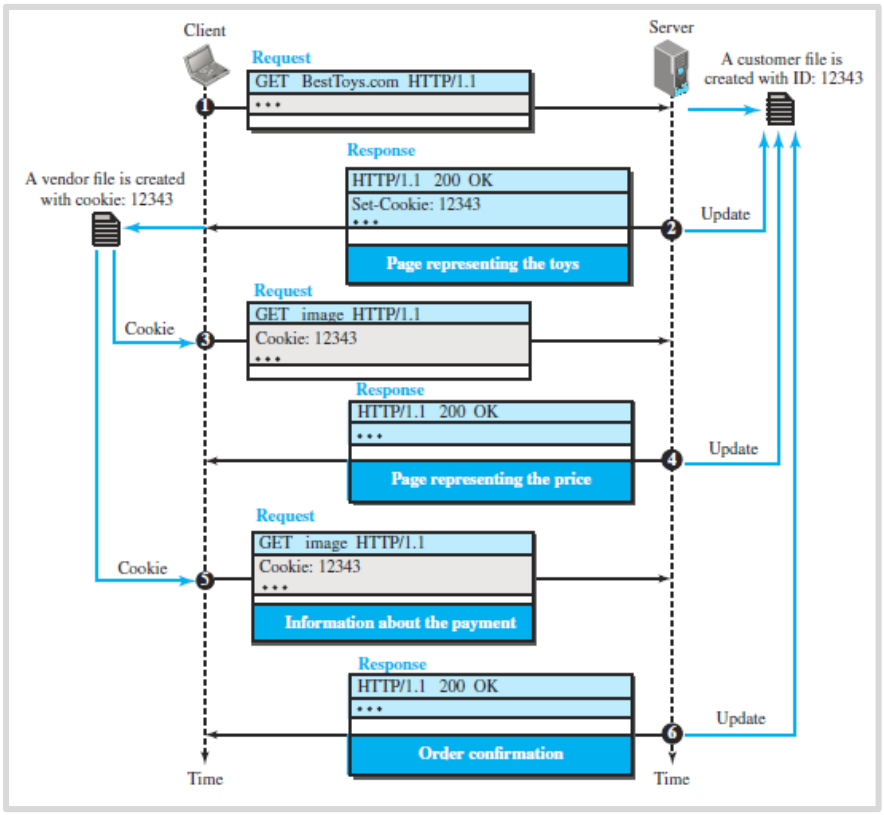


Fuente: Forouzan, 2013, p. 882

En el ejemplo 2, el cliente envía una solicitud a un sitio de compras. El servidor responde enviando imágenes de cada juguete disponible para ser comprando y además un header con “set-cookie”. El cliente visualizará las imágenes y guardará la Cookie. Para conocer como funciona una Cookie, consultar la bibliografía básica y la RFC6265

En el siguiente mensaje, el cliente hace click sobre una de las imágenes correspondiente a un juguete por lo que se envía un GET con la correspondiente Cookie (12345). El servidor puede reconocer al cliente 12345 y agrega el jueguete seleccionado al carrito de compras 12345.
El server ahora responde con información sobre el precio que debe abonarse por el juguete. El comprandor envía información de pago usando la misma cookie para que el servidor pueda reconocerlo. FInalment el sevidor responde confirmando el pago.

Figura 6: Ejemplo 2 uso de cookies



Fuente: Forouzan, 2013, p. 885

Referencias

Tanenbaum, A (2012). La capa de aplicación en Redes *de Computadoras.* Madrid: Editorial Pearson Education

Forouzan, B (2013). Application Layer en *Data Communications AND Networking.* Estados Unidos: McGraw-Hill

IETF (1999). RFC2616. Hypertext Transfer Protocol -- HTTP/1.1. Estados Unidos. https://tools.ietf.org/html/rfc2616

IETF (2015). RFC7514. Hypertext Transfer Protocol Version 2 (HTTP/2). Estados Unidos. https://tools.ietf.org/html/rfc7540