

Un servidor IMAP asociará cada mensaje con una carpeta; cuando un mensaje llega al servidor, se asocia con la carpeta INBOX (Bandeja de entrada) del destinatario, el cual puede entonces pasar el mensaje a una nueva carpeta creada por el usuario, leer el mensaje, borrarlo, etc. El protocolo IMAP proporciona comandos que permiten a los usuarios crear carpetas y mover los mensajes de una carpeta a otra. IMAP también proporciona comandos que permiten a los usuarios realizar búsquedas en carpetas remotas para localizar mensajes que cumplan unos determinados criterios. Observe que, a diferencia de POP3, un servidor IMAP mantiene información acerca del estado a lo largo de las sesiones IMAP, como por ejemplo, los nombres de las carpetas y los mensajes asociados con cada una de ellas.

Otra importante característica de IMAP es que dispone de comandos que permiten a un agente de usuario obtener partes componentes de los mensajes. Por ejemplo, un agente de usuario puede obtener solo la cabecera del mensaje o solo una parte de un mensaje MIME de varias partes. Esta característica resulta útil cuando se emplea una conexión con un ancho de banda pequeño (por ejemplo, un enlace de módem de baja velocidad) entre el agente de usuario y su servidor de correo. Con una conexión de ancho de banda pequeño, puede que el usuario no quiera descargar todos los mensajes guardados en su buzón y que desee saltarse los mensajes largos que pueda haber; por ejemplo un archivo de audio o de vídeo.

Correo electrónico web

Actualmente, cada vez más usuarios envían y acceden a su correo electrónico a través de sus navegadores web. Hotmail introdujo a mediados de la década de 1990 el acceso basado en la Web; actualmente, también ofrece este servicio Yahoo, Google, así como casi todas las principales universidades y empresas. Con este servicio, el agente de usuario es un navegador web corriente y el usuario se comunica con su buzón remoto a través de HTTP. Cuando un destinatario, como Benito, desea acceder a un mensaje de su buzón, este es enviado desde el servidor de correo de Benito al navegador del mismo utilizando el protocolo HTTP en lugar de los protocolos POP3 o IMAP. Cuando un emisor, como Alicia, desea enviar un mensaje de correo electrónico, este es transmitido desde su navegador a su servidor de correo a través de HTTP en lugar de mediante SMTP. Sin embargo, el servidor de correo de Alicia, continúa enviando mensajes a, y recibiendo mensajes de, otros servidores de correo que emplean SMTP.

2.4 DNS: el servicio de directorio de Internet

Las personas podemos ser identificadas de muchas maneras. Por ejemplo, podemos ser identificadas por el nombre que aparece en nuestro certificado de nacimiento, por nuestro número de la seguridad social o por el número del carnet de conducir. Aunque cada uno de estos datos se puede utilizar para identificar a las personas, dentro de un determinado contexto un identificador puede ser más apropiado que otro. Por ejemplo, las computadoras del IRS (la terrible agencia tributaria de Estados Unidos) prefieren utilizar los números de la seguridad social de longitud fija que el nombre que aparece en el certificado de nacimiento. Por otro lado, las personas prefieren emplear el nombre que figura en los certificados de nacimiento, más fáciles de recordar, a los números de la seguridad social (puede imaginar que alguien le dijera “Hola. Mi nombre es 132-67-9875. Este es mi marido, 178-87-1146.”)

No solo las personas podemos ser identificadas de diversas formas, los hosts de Internet también. Un identificador para los hosts es el **nombre de host**. Los nombres de host, como por ejemplo `www.facebook.com`, `www.google.com`, `gaia.cs.umass.edu`, son mnemónicos y son, por tanto, entendidos por las personas. Sin embargo, los nombres de host proporcionan poca o ninguna información acerca de la ubicación del host dentro de Internet. (Un nombre de host como `www.eurecom.fr`, que termina con el código de país, `.fr`, nos informa de que es probable que el host

se encuentre en Francia, pero esto no dice mucho más.) Además, puesto que los nombres de host pueden constar de caracteres alfanuméricos de longitud variable, podrían ser difíciles de procesar por los routers. Por estas razones, los hosts también se identifican mediante **direcciones IP**.

En el Capítulo 4 veremos en detalle las direcciones IP, pero le resultará útil leer ahora una breve exposición sobre las mismas. Una dirección IP consta de cuatro bytes y sigue una rígida estructura jerárquica. El aspecto de una dirección IP es, por ejemplo, 121.7.106.83, donde cada punto separa uno de los bytes expresados en notación decimal con valores comprendidos entre 0 y 255. Una dirección IP es jerárquica porque al explorar la dirección de izquierda a derecha, se obtiene información cada vez más específica acerca del lugar en el que está ubicado el host dentro de Internet (es decir, en qué red de la red de redes). De forma similar, cuando examinamos una dirección postal de abajo a arriba, obtenemos cada vez información más específica acerca del lugar definido por la dirección.

2.4.1 Servicios proporcionados por DNS

Acabamos de ver que hay dos formas de identificar un host, mediante un nombre de host y mediante una dirección IP. Las personas prefieren utilizar como identificador el nombre de host, mientras que los routers prefieren emplear las direcciones IP de longitud fija y que siguen una estructura jerárquica. Para reconciliar estas preferencias necesitamos un servicio de directorio que traduzca los nombres de host en direcciones IP. Esta es la tarea principal que lleva a cabo el **Sistema de nombres de dominio (DNS, Domain Name System)** de Internet. DNS es (1) una base de datos distribuida implementada en una jerarquía de servidores DNS y (2) un protocolo de la capa de aplicación que permite a los hosts consultar la base de datos distribuida. Los servidores DNS suelen ser máquinas UNIX que ejecutan el software BIND (*Berkeley Internet Name Domain*, Dominio de nombres de Internet de Berkeley) [BIND 2016]. El protocolo DNS se ejecuta sobre UDP y utiliza el puerto 53.

Otros protocolos de la capa de aplicación, como HTTP y SMTP, emplean habitualmente DNS para traducir los nombres de host suministrados por el usuario en direcciones IP. Por ejemplo, veamos qué ocurre cuando un navegador (es decir, un cliente HTTP), que se ejecuta en un determinado host de usuario, solicita el URL `www.unaescuela.edu/index.html`. Para que el host del usuario pueda enviar un mensaje de solicitud HTTP al servidor web `www.unaescuela.edu`, el host del usuario debe obtener en primer lugar la dirección IP de `www.unaescuela.edu`. Esto se hace del siguiente modo:

1. La propia máquina cliente ejecuta el lado del cliente de la aplicación DNS.
2. El navegador extrae el nombre de host, `www.unaescuela.edu`, del URL y pasa el nombre de host al lado del cliente de la aplicación DNS.
3. El cliente DNS envía una consulta que contiene el nombre de host a un servidor DNS.
4. El cliente DNS recibe finalmente una respuesta, que incluye la dirección IP correspondiente al nombre del host.
5. Una vez que el navegador recibe la dirección IP del servidor DNS, puede iniciar una conexión TCP con el proceso servidor HTTP localizado en el puerto 80 en esa dirección IP.

Podemos ver a partir de este ejemplo que DNS añade un retardo adicional, en ocasiones, sustancial, a las aplicaciones de Internet que le utilizan. Afortunadamente, como veremos más adelante, la dirección IP deseada a menudo está almacenada en caché en un servidor DNS “próximo”, lo que ayuda a reducir el tráfico de red DNS, así como el retardo medio del servicio DNS.

DNS proporciona algunos otros servicios importantes además de la traducción de los nombres de host en direcciones IP:

- **Alias de host.** Un host con un nombre complicado puede tener uno o más alias. Por ejemplo, un nombre de host como `relay1.west-coast.enterprise.com` podría tener, digamos, dos alias como `enterprise.com` y `www.enterprise.com`. En este caso, el nombre de host

`relay1.west-coast.enterprise.com` se dice que es el **nombre de host canónico**. Los alias de nombres de host, cuando existen, normalmente son más mnemónicos que los nombres canónicos. Una aplicación puede invocar DNS para obtener el nombre de host canónico para un determinado alias, así como la dirección IP del host.

- **Alias del servidor de correo.** Por razones obvias, es enormemente deseable que las direcciones de correo electrónico sean mnemónicas. Por ejemplo, si Benito tiene una cuenta de Yahoo Mail, su dirección de correo puede ser tan simple como `benito@yahoo.mail`. Sin embargo, el nombre de host del servidor de correo de Yahoo es más complicado y mucho menos mnemónico que simplemente `yahoo.com` (por ejemplo, el nombre canónico podría ser algo parecido a `relay1.west-coast.yahoo.com`). Una aplicación de correo puede invocar al servicio DNS para obtener el nombre de host canónico para un determinado alias, así como la dirección IP del host. De hecho, el registro MX (véase más adelante) permite al servidor de correo y al servidor web de una empresa tener nombres de host (con alias) iguales; por ejemplo, tanto el servidor web como el servidor de correo de una empresa se pueden llamar `enterprise.com`.
- **Distribución de carga.** DNS también se emplea para realizar la distribución de carga entre servidores replicados, como los servidores web replicados. Los sitios con una gran carga de trabajo, como `cnn.com`, están replicados en varios servidores, ejecutándose cada servidor en un sistema terminal distinto y teniendo cada uno una dirección IP diferente. Para los servidores web replicados hay asociado un *conjunto* de direcciones IP con un único nombre de host canónico. La base de datos DNS contiene este conjunto de direcciones IP. Cuando los clientes realizan una consulta DNS sobre un nombre asignado a un conjunto de direcciones, el servidor responde con el conjunto completo de direcciones IP, pero rota el orden de las direcciones en cada respuesta. Dado que normalmente un cliente envía su mensaje de solicitud HTTP a la dirección IP que aparece en primer lugar dentro del conjunto, la rotación DNS distribuye el tráfico entre los servidores replicados. La rotación DNS también se emplea para el correo electrónico de modo que múltiples servidores de correo pueden tener el mismo alias. Además, las empresas de distribución de contenido como Akamai han utilizado DNS de formas muy sofisticadas [Dilley 2002] para proporcionar la distribución de contenido web (véase la Sección 2.6.3).

DNS está especificado en los documentos RFC 1034 y RFC 1035, y actualizado en varios RFC adicionales. Es un sistema complejo y aquí solo hemos visto los aspectos básicos de su funcionamiento. El lector interesado puede consultar estos RFC y el libro de Abitz y Liu [Abitz 1993]; también puede consultar el documento retrospectivo [Mockapetris 1988], que proporciona una descripción sobre qué es DNS y el por qué de DNS; asimismo puede ver [Mockapetris 2005].

2.4.2 Cómo funciona DNS

Ahora vamos a presentar a alto nivel cómo funciona DNS. Nos centraremos en el servicio de traducción nombre de host-dirección IP.

Suponga que una determinada aplicación (como por ejemplo un navegador web o un lector de correo), que se ejecuta en el host de un usuario, necesita traducir un nombre de host en una dirección IP. La aplicación invocará al lado del cliente de DNS, especificando el nombre de host del que necesita la correspondiente traducción. (En muchos sistemas basados en UNIX, `gethostbyname()` es la llamada a función que una aplicación utiliza para llevar a cabo la traducción.) Entonces, la aplicación DNS en el host del usuario entra en funcionamiento, enviando un mensaje de consulta a la red. Todos los mensajes de consulta y de respuesta DNS se envían dentro de datagramas UDP al puerto 53. Transcurrido un cierto retardo, del orden de milisegundos a segundos, el servicio DNS del host del usuario recibe un mensaje de respuesta DNS que proporciona la traducción deseada, la cual se pasa entonces a la aplicación que lo ha invocado. Por tanto, desde la perspectiva de dicha aplicación que se ejecuta en el host del usuario, DNS es una caja negra que proporciona un servicio de traducción simple y directo. Pero, de hecho, la caja negra que implementa el servicio es compleja,



EN LA PRÁCTICA

DNS: FUNCIONES CRÍTICAS DE RED MEDIANTE EL PARADIGMA CLIENTE-SERVIDOR

Como HTTP, FTP y SMTP, el protocolo DNS es un protocolo de la capa de aplicación puesto que (1) se ejecuta entre sistemas terminales que están en comunicación utilizando el paradigma cliente-servidor y (2) se basa en un protocolo de transporte subyacente extremo a extremo para transferir los mensajes DNS entre los sistemas terminales en comunicación. Sin embargo, desde otro punto de vista, la función de DNS es bastante diferente a la de las aplicaciones web de transferencia de archivos y de correo electrónico. A diferencia de estas aplicaciones, DNS no es una aplicación con la que el usuario interactúe directamente; en su lugar, DNS lleva a cabo una de las funciones básicas en Internet: traducir los nombres de hosts en sus direcciones IP subyacentes para las aplicaciones de usuario y otras aplicaciones software de Internet. Hemos mencionado en la Sección 1.2 que gran parte de la complejidad de la arquitectura de Internet se encuentra en las “fronteras” de la red. DNS, que implementa el importante proceso de traducción de nombres en direcciones, utilizando clientes y servidores ubicados en la frontera de la red, es otro ejemplo más de dicha filosofía de diseño.

constando de un gran número de servidores DNS distribuidos por todo el mundo, así como de un protocolo de la capa de aplicación que especifica cómo los servidores DNS y los hosts que realizan las consultas se comunican.

Un diseño simple de DNS podría ser un servidor DNS que contuviera todas las correspondencias entre nombres y direcciones. En este diseño centralizado, los clientes simplemente dirigirían todas las consultas a un mismo servidor DNS y este respondería directamente a las consultas de los clientes. Aunque la simplicidad de este diseño es atractiva, es inapropiado para la red Internet de hoy día a causa de la enorme (y creciente) cantidad de hosts. Entre los problemas con un diseño centralizado podemos citar los siguientes:

- **Un único punto de fallo.** Si el servidor DNS falla, entonces ¡también falla toda la red Internet!
- **Volumen de tráfico.** Un único servidor DNS tendría que gestionar todas las consultas DNS (de todas las solicitudes HTTP y mensajes de correo electrónico generados en cientos de millones de hosts).
- **Base de datos centralizada distante.** Un único servidor DNS no puede “estar cerca” de todos los clientes que efectúan consultas. Si colocamos ese único servidor DNS en la ciudad de Nueva York, entonces todas las consultas desde Australia deberían viajar hasta el otro extremo del globo, quizá a través de enlaces lentos y congestionados. Esto podría llevar por tanto a retardos significativos.
- **Mantenimiento.** Ese único servidor DNS tendría que mantener registros de todos los hosts de Internet. No solo sería una enorme base de datos centralizada, sino que tendría que ser actualizada con frecuencia con el fin de incluir todos los hosts nuevos.

En resumen, una base de datos centralizada almacenada en un único servidor DNS simplemente *no podría escalarse*. En consecuencia, el sistema DNS utiliza un diseño distribuido. De hecho, DNS es un estupendo ejemplo de cómo se puede implementar una base de datos distribuida en Internet.

Una base de datos jerárquica y distribuida

Para abordar el problema del escalado, DNS utiliza un gran número de servidores, organizados de forma jerárquica y distribuidos alrededor de todo el mundo. Ningún servidor DNS dispone de todas las correspondencias de todos los hosts de Internet. En su lugar, las correspondencias están repartidas por los servidores DNS. En una primera aproximación, podemos decir que existen tres clases de

servidores DNS: los servidores DNS raíz, los servidores DNS de dominio de nivel superior (TLD, *Top-Level Domain*) y los servidores DNS autoritativos, organizados en una jerarquía como la mostrada en la Figura 2.17. Con el fin de comprender cómo estas tres clases de servidores interactúan, suponga que un cliente DNS desea determinar la dirección IP correspondiente al nombre de host `www.amazon.com`. En una primera aproximación tienen lugar los siguientes sucesos: primero, el cliente contacta con uno de los servidores raíz, el cual devuelve las direcciones IP para los servidores TLD del dominio de nivel superior `com`. A continuación, el cliente contacta con uno de estos servidores TLD, que devuelve la dirección IP de un servidor autoritativo para `amazon.com`. Por último, el cliente contacta con uno de los servidores autoritativos de `amazon.com`, el cual devuelve la dirección IP correspondiente al nombre de host `www.amazon.com`. Enseguida examinaremos este proceso de búsqueda DNS con más detalle. Pero en primer lugar, echemos un vistazo a estas tres clases de servidores DNS:

- **Servidores DNS raíz.** Existen unos 400 servidores de nombres raíz distribuidos por todo el mundo. La Figura 2.18 muestra los países que disponen de estos servidores, con los países que tienen más de diez servidores sombreados en gris oscuro. Trece organizaciones diferentes gestionan estos servidores de nombres raíz. Puede encontrar una lista completa de los servidores de nombres raíz, junto con las organizaciones que los gestionan y sus direcciones IP en [Root Servers 2016]. Los servidores de nombres raíz proporcionan las direcciones IP de los servidores TLD.
- **Servidores de dominio de nivel superior (TLD).** Para todos los dominios de nivel superior, como son `com`, `org`, `net`, `edu` y `gov`, y todos los dominios de nivel superior correspondientes a los distintos países, como por ejemplo, `uk`, `fr`, `ca` y `jp`, existe un servidor TLD (o agrupación de servidores). La empresa Verisign Global Registry Services mantiene los servidores TLD para el dominio de nivel superior `com` y la empresa Educause mantiene los servidores TLD para el dominio de nivel superior `edu`. La infraestructura de red que da soporte a un TLD puede ser muy grande y compleja; consulte [Osterweil 2012] para ver una introducción de la red de Verisign. Consulte [TLD list 2016] para ver una lista de todos los dominios de nivel superior. Los servidores TLD proporcionan las direcciones IP para los servidores DNS autoritativos.
- **Servidores DNS autoritativos.** Todas las organizaciones que tienen hosts accesibles públicamente (como son los servidores web y los servidores de correo) a través de Internet deben proporcionar registros DNS accesibles públicamente que establezcan la correspondencia entre los nombres de dichos hosts y sus direcciones IP. Un servidor DNS autoritativo de una organización alberga estos registros DNS. Una organización puede elegir implementar su propio servidor DNS autoritativo para almacenar estos registros; alternatively, la organización puede pagar por tener esos registros almacenados en un servidor DNS autoritativo de algún proveedor de servicios. La mayoría de las universidades y de las empresas de gran tamaño implementan y mantienen sus propios servidores DNS autoritativos principal y secundario (*backup*).

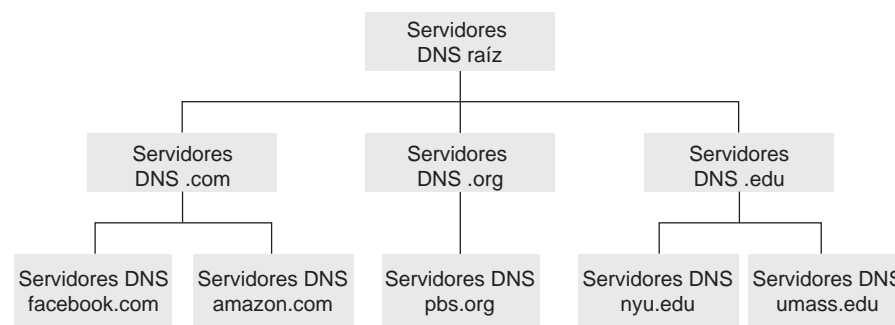


Figura 2.17 ♦ Parte de la jerarquía de los servidores DNS.



Figura 2.18 ♦ Servidores DNS raíz en 2016.

Todos los servidores DNS raíz, TLD y autoritativos pertenecen a la jerarquía de servidores DNS, como se muestra en la Figura 2.17. Existe otro tipo importante de servidor DNS conocido como **servidor DNS local**. Un servidor DNS local no pertenece estrictamente a la jerarquía de servidores, pero no obstante es importante dentro de la arquitectura DNS. Cada ISP (como por ejemplo un ISP residencial o un ISP institucional) tiene un servidor DNS local (también denominado servidor de nombres predeterminado). Cuando un host se conecta a un ISP, este proporciona al host las direcciones IP de uno o más de sus servidores DNS locales (normalmente a través de DHCP, lo que veremos en el Capítulo 4). Usted puede determinar fácilmente la dirección IP de su servidor DNS local accediendo a las ventanas de estado de la red en Windows o UNIX. Generalmente, un servidor DNS local de un host se encuentra “cerca” de ese host. En el caso de un ISP institucional, el servidor DNS local puede encontrarse en la misma LAN que el host; en el caso de un ISP residencial, estará separado del host no más de unos pocos routers. Cuando un host realiza una consulta DNS, esta se envía al servidor DNS local, que actúa como proxy, reenviando la consulta a la jerarquía de servidores DNS, como veremos más detalladamente a continuación.

Veamos un ejemplo sencillo. Suponga que el host `cse.nyu.edu` desea conocer la dirección de `gaia.cs.umass.edu`. Suponga también que el servidor DNS local de la Universidad de Nueva York para `cse.nyu.edu` se denomina `dns.nyu.edu` y que un servidor DNS autoritativo para `gaia.cs.umass.edu` se llama `dns.umass.edu`. Como se muestra en la Figura 2.19, el host `cse.nyu.edu` envía en primer lugar un mensaje de consulta DNS a su servidor DNS local, `dns.nyu.edu`. El mensaje de consulta contiene el nombre de host que debe ser traducido, `gaia.cs.umass.edu`. El servidor DNS local reenvía la consulta a un servidor DNS raíz, el cual toma el sufijo `edu` y devuelve al servidor DNS local una lista de las direcciones IP de los servidores TLD responsables del dominio `edu`. El servidor DNS local reenvía a continuación el mensaje de consulta a uno de estos servidores TLD. El servidor TLD toma nota del sufijo `umass.edu` y responde con la dirección IP del servidor DNS autoritativo correspondiente a la Universidad de Massachusetts, es decir, `dns.umass.edu`. Por último, el servidor DNS local reenvía la consulta directamente a `dns.umass.edu`, que responde con la dirección IP de `gaia.cs.umass.edu`. Observe que en este ejemplo, para obtener la dirección correspondiente a un nombre de host, se han enviado ocho mensajes DNS: ¡cuatro mensajes de consulta y cuatro mensajes de respuesta! Pronto veremos cómo el almacenamiento en caché DNS reduce este tráfico de consultas.

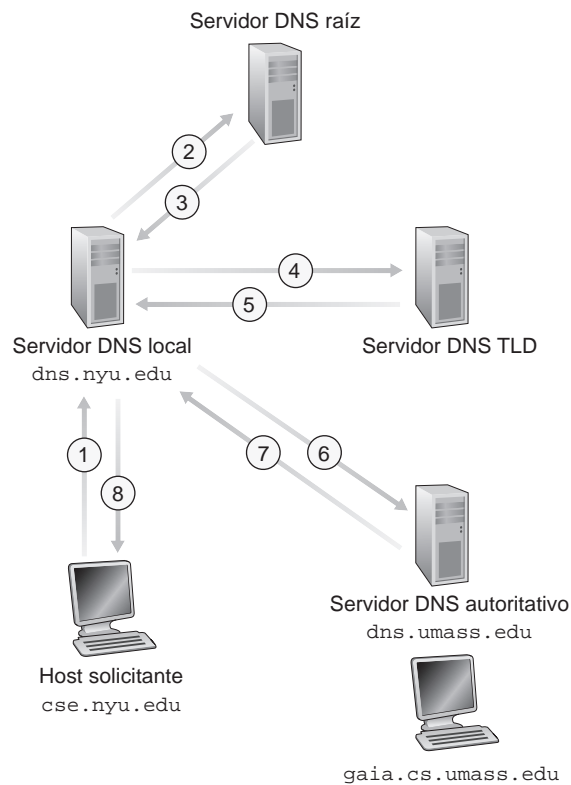


Figura 2.19 ♦ Interacción de los distintos servidores DNS.

En el ejemplo anterior suponíamos que el servidor TLD conoce el servidor DNS autoritativo correspondiente al nombre de host. En general esto no es así. En lugar de ello, el servidor TLD puede saber únicamente de un servidor DNS intermedio, el cual a su vez sabe cuál es el servidor DNS autoritativo para el nombre de host. Por ejemplo, suponga de nuevo que la Universidad de Massachusetts tiene un servidor DNS para la universidad, denominado `dns.umass.edu`. Suponga también que cada uno de los departamentos de la Universidad de Massachusetts tiene su propio servidor DNS y que cada servidor DNS departamental es un servidor autoritativo para todos los hosts del departamento. En este caso, cuando el servidor DNS intermedio `dns.umass.edu` recibe una consulta para un host cuyo nombre termina en `cs.umass.edu`, devuelve a `dns.nyu.edu` la dirección IP de `dns.cs.umass.edu`, que es autoritativo para todos los nombres de host que terminan con `cs.umass.edu`. El servidor DNS local `dns.nyu.edu` envía entonces la consulta al servidor DNS autoritativo, que devuelve la correspondencia deseada al servidor DNS local, el cual a su vez devuelve la correspondencia al host que ha hecho la solicitud. En este caso, ¡se han enviado un total de 10 mensajes!

El ejemplo mostrado en la Figura 2.19 utiliza tanto **consultas recursivas** como **consultas iterativas**. La consulta enviada desde `cse.nyu.edu` a `dns.nyu.edu` es una consulta recursiva, ya que la consulta solicita a `dns.nyu.edu` que obtenga por sí mismo la correspondencia. Pero las tres consultas siguientes son iterativas puesto que todas las respuestas son devueltas directamente a `dns.nyu.edu`. En teoría, cualquier consulta DNS puede ser iterativa o recursiva. Por ejemplo, la Figura 2.20 muestra una cadena de consultas DNS para las que todas las consultas son recursivas. En la práctica, las consultas normalmente siguen el patrón mostrado en la Figura 2.19: la consulta procedente del host que hace la solicitud al servidor DNS local es recursiva y las restantes consultas son iterativas.

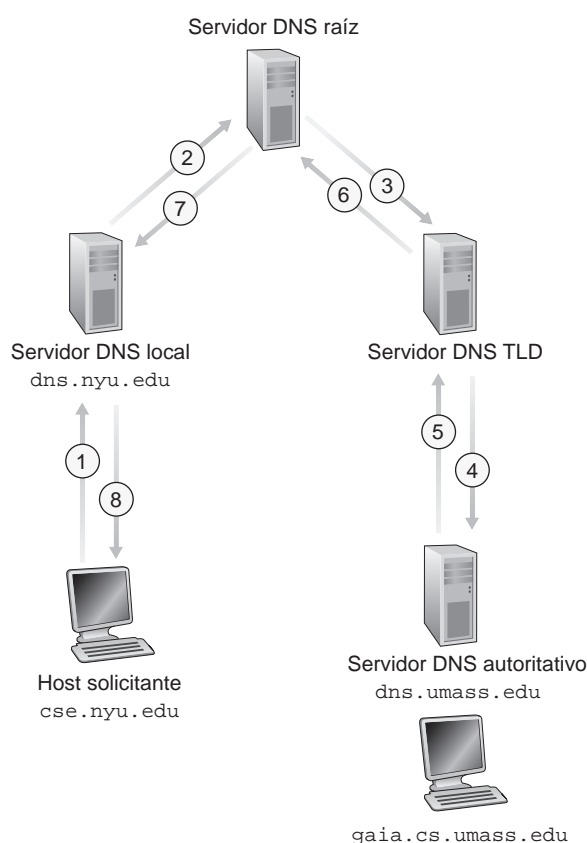


Figura 2.20 ♦ Consultas recursivas en DNS.

Almacenamiento en caché DNS

Hasta el momento hemos ignorado el **almacenamiento en caché DNS**, una funcionalidad extremadamente importante del sistema DNS. En realidad, DNS explota exhaustivamente el almacenamiento en caché para mejorar el comportamiento de los retardos y reducir el número de mensajes DNS que van de un lado a otro por Internet. La idea que se esconde detrás del almacenamiento en caché DNS es muy simple. En una cadena de consultas, cuando un servidor DNS recibe una respuesta DNS (que contiene, por ejemplo, una correspondencia entre un nombre de host y una dirección IP), puede almacenar esta información en su memoria local. Por ejemplo, en la Figura 2.19, cada vez que el servidor DNS local `dns.nyu.edu` recibe una respuesta de algún servidor DNS, puede almacenar en caché cualquier información contenida en esa respuesta. Si una relación nombre de host/dirección IP se almacena en la caché de un servidor DNS y llegan otras consultas a ese mismo servidor DNS preguntando por el mismo nombre de host, el servidor DNS puede proporcionar la dirección IP deseada, incluso aunque no sea autoritativo para el nombre de host. Dado que los hosts y las correspondencias entre nombres de host y direcciones IP no son permanentes, los servidores DNS descartan la información almacenada en caché pasado un cierto periodo de tiempo (normalmente, unos dos días).

Por ejemplo, suponga que un host `apricot.nyu.edu` consulta a `dns.nyu.edu` solicitándole la dirección IP correspondiente al nombre de host `cnn.com`. Suponga también que unas pocas horas más tarde, otro host de la Universidad de Nueva York, digamos `kiwi.nyu.edu`, también hace una consulta a `dns.nyu.edu` especificando el mismo nombre de host. Gracias al almacenamiento en caché, el servidor DNS local podrá devolver de forma inmediata la dirección IP de `cnn.com` al segundo host que la ha solicitado sin tener que consultar a ningún

otro servidor DNS. Un servidor DNS local también puede almacenar en caché las direcciones IP de los servidores TLD, permitiendo así a los servidores DNS locales saltarse a los servidores DNS raíz en una cadena de consultas. De hecho, gracias al almacenamiento en caché, todas las consultas DNS, excepto una pequeña fracción de las mismas, se saltan a los servidores raíz.

2.4.3 Registros y mensajes DNS

Los servidores DNS que implementan conjuntamente la base de datos distribuida DNS almacenan los **registros de recursos (RR)**, incluyendo los que proporcionan las correspondencias entre nombre de host y dirección IP. Cada mensaje de respuesta DNS transporta uno o más registros de recursos. En esta y en la subsección siguiente, proporcionamos una breve introducción a los recursos y mensajes DNS; puede encontrar información detallada en [Abitz 1993] o en los RFC sobre DNS [RFC 1034; RFC 1035].

Un registro de recurso está formado por los siguientes cuatro campos:

(Nombre, Valor, Tipo, TTL)

TTL es el tiempo de vida del registro de recurso; determina cuándo un recurso debería ser eliminado de una caché. En los registros de ejemplo dados a continuación, hemos ignorado el campo TTL. El significado de Nombre y Valor depende del campo Tipo:

- Si Tipo=A, entonces Nombre es un nombre de host y Valor es la dirección IP correspondiente a dicho nombre. Por tanto, un registro de tipo A proporciona la correspondencia estándar nombre de host-dirección IP. Por ejemplo, (relay1.bar.foo.com, 145.37.93.126, A) es un registro de tipo A.
- Si Tipo = NS, entonces Nombre es un dominio (como foo.com) y Valor es el nombre de host de un servidor DNS autoritativo que sabe cómo obtener las direcciones IP de los hosts del dominio. Este registro se utiliza para enrutar las consultas DNS a lo largo de la cadena de consultas. Por ejemplo, (foo.com, dns.foo.com, NS) es un registro de tipo NS.
- Si Tipo = CNAME, entonces Valor es un nombre de host canónico correspondiente al alias especificado por Nombre. Este registro puede proporcionar a los hosts que hacen consultas el nombre canónico correspondiente a un nombre de host. Por ejemplo, (foo.com, relay1.bar.foo.com) es un registro CNAME.
- Si Tipo = MX, entonces Valor es el nombre canónico de un servidor de correo que tiene un alias dado por Nombre. Por ejemplo, (foo.com, mail.bar.foo.com, MX) es un registro MX. Los registros MX permiten a los nombres de host de los servidores de correo tener alias simples. Observe que utilizando el registro MX, una empresa puede tener el mismo alias para su servidor de correo y para uno de sus otros servidores (como por ejemplo, el servidor web). Para obtener el nombre canónico del servidor de correo, un cliente DNS consultaría un registro MX y para conocer el nombre canónico del otro servidor, consultaría el registro CNAME.

Si un servidor DNS es autoritativo para un determinado nombre de host, entonces el servidor DNS contendrá un registro de tipo A para el nombre de host. (Incluso aunque el servidor DNS no sea autoritativo, puede contener un registro de tipo A en su caché.) Si un servidor no es autoritativo para un nombre de host, entonces el servidor contendrá un registro de tipo NS para el dominio que incluye el nombre de host; también contendrá un registro de tipo A que proporcione la dirección IP del servidor DNS en el campo Valor del registro NS. Por ejemplo, suponga un servidor TLD edu que no es autoritativo para el host gaia.cs.umass.edu. Por tanto, este servidor contendrá un registro para un dominio que incluye el host gaia.cs.umass.edu, por ejemplo, (umass.edu, dns.umass.edu, NS). El servidor TLD edu también contendría un registro de tipo A, que establece la correspondencia entre el servidor DNS dns.umass.edu y una dirección IP, como en (dns.umass.edu, 128.119.40.111, A).

Mensajes DNS

Anteriormente en esta sección hemos hecho referencia a los mensajes de respuesta y consultas DNS. Únicamente existen estas dos clases de mensajes DNS. Además, tanto los mensajes de respuesta como de consulta utilizan el mismo formato, que se muestra en la Figura 2.21. La semántica en los distintos campos de un mensaje DNS es la siguiente:

- Los primeros 12 bytes constituyen la *sección de cabecera*, la cual contiene una serie de campos. El primero de estos campos es un número de 16 bits que identifica la consulta. Este identificador se copia en el mensaje de respuesta a la consulta, lo que permite al cliente establecer las correspondencias correctas entre las respuestas recibidas y las consultas enviadas. En el campo Indicadores se incluyen una serie de indicadores. Un indicador consulta/respuesta de 1 bit informa de si el mensaje es una consulta (0) o una respuesta (1). Un indicador autoritativo de 1 bit se activa en un mensaje de respuesta cuando un servidor DNS es un servidor autoritativo para un nombre solicitado. El indicador recursión-deseada, también de 1 bit, se activa cuando un cliente (host o servidor DNS) desea que el servidor DNS realice una recursión cuando no disponga del registro. En un mensaje de respuesta, el campo de recursión-disponible de 1 bit se activa si el servidor DNS soporta la recursión. En la cabecera también se incluyen cuatro campos “número de”, que indican el número de apariciones de los cuatro tipos de secciones de datos que siguen a la cabecera.
- La *sección cuestiones* contiene información acerca de la consulta que se va a realizar. Esta sección incluye (1) un campo de nombre que contiene el nombre que se va a consultar y (2) un campo de tipo que especifica el tipo de cuestión que se plantea acerca del nombre; por ejemplo, la dirección del host asociada con un nombre (tipo A) o el servidor de correo para un nombre (tipo MX).
- En una respuesta de un servidor DNS, la *sección respuestas* contiene los registros del recurso para el nombre que fue consultado originalmente. Recuerde que en cada registro de recurso existe un parámetro Tipo (por ejemplo, A, NS, CNAME y MX), un parámetro Valor y el parámetro TTL. Una respuesta puede devolver varios registros de recursos, ya que un nombre de host puede tener asociadas varias direcciones IP (por ejemplo, para los servidores web replicados, como hemos visto anteriormente en esta sección).
- La *sección autoridad* contiene registros de otros servidores autoritativos.
- La *sección información adicional* contiene otros registros útiles. Por ejemplo, el campo de respuesta en un mensaje de respuesta a una consulta MX contiene un registro de recurso que

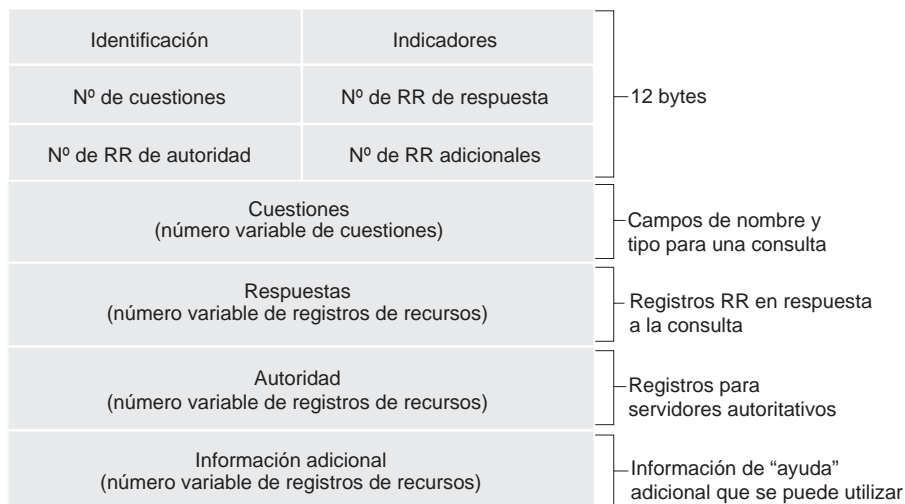


Figura 2.21 ♦ Formato de los mensajes DNS.

proporciona el nombre de host canónico de un servidor de correo. Esta sección de información adicional contiene un registro de tipo A que proporciona la dirección IP para el nombre de host canónico del servidor de correo.

¿Le gustaría enviar un mensaje de consulta DNS directamente desde el host en el que está trabajando a algún servidor DNS? Esto podría hacerse fácilmente con el **programa nslookup**, que está disponible en la mayoría de las plataformas Windows y UNIX. Por ejemplo, en un host Windows basta con abrir la aplicación Símbolo del sistema (*prompt*) e invocar al programa nslookup escribiendo simplemente “nslookup”. A continuación, podrá enviar una consulta DNS a cualquier servidor DNS (raíz, TLD o autoritativo). Después de recibir el mensaje de respuesta del servidor DNS, nslookup mostrará los registros incluidos en la respuesta (en un formato legible para las personas). Como alternativa a la ejecución de nslookup desde su propio host, puede visitar uno de los muchos sitios web que permiten utilizar nslookup de forma remota (basta con escribir “nslookup” en un motor de búsqueda para acceder a uno de estos sitios). La práctica de laboratorio DNS Wireshark disponible al final de este capítulo le permitirá explorar DNS en detalle.

Inserción de registros en la base de datos DNS

La exposición anterior se ha centrado en cómo se recuperan los registros de la base de datos DNS. Es posible que ahora se esté preguntando cómo se introducen los registros en dicha base de datos. Veamos cómo se hace esto en el contexto de un ejemplo concreto. Suponga que hemos creado una nueva empresa llamada Network Utopia. Lo primero que seguramente deseará hacer es registrar el nombre de dominio `networkutopia.com` en un registro. Un **registrador** es una entidad comercial que verifica la unicidad del nombre de dominio, lo añade a la base de datos DNS (como veremos más adelante) y percibe unas pequeñas tasas por sus servicios. Antes de 1999, un único registrador, Network Solutions, tenía el monopolio sobre el registro de nombres para los dominios `com`, `net` y `org`. Pero actualmente, existen muchas entidades registradoras que compiten por los clientes. La ICANN (*Internet Corporation for Assigned Names and Numbers*, Corporación de Internet para nombres y números asignados) acredita a las distintas entidades registradoras. En la dirección `http://www.internic.net` puede encontrar una lista completa de estas entidades.

Al registrar el nombre de dominio `networkutopia.com` en alguna entidad registradora, también tendrá que proporcionarle los nombres y direcciones IP de sus servidores DNS autoritativos principal y secundario. Suponga que en nuestro ejemplo estos datos son: `dns1.networkutopia.com`, `dns2.networkutopia.com`, `212.212.212.1` y `212.212.212.2`. Para cada uno de estos dos servidores DNS autoritativos, el registrador se asegura de que se introduzca un registro de tipo NS y un registro de tipo A en los servidores TLD `com`. Específicamente, para el servidor autoritativo principal de `networkutopia.com`, la entidad registradora deberá insertar los siguientes dos registros de recursos en el sistema DNS:

```
(networkutopia.com, dns1.networkutopia.com, NS)
(dns1.networkutopia.com, 212.212.212.1, A)
```

También tendrá que asegurarse de que el registro de recurso de tipo A para su servidor web `www.networkutopia.com` y el registro de recurso de tipo MX para su servidor de correo `mail.networkutopia.com` se han introducido en sus servidores DNS autoritativos. (Hasta hace poco, los contenidos de los servidores DNS se configuraban estáticamente; por ejemplo, a partir de un archivo de configuración creado por un administrador del sistema. Sin embargo, recientemente se ha añadido una opción de actualización (UPDATE) al protocolo DNS que permite añadir o borrar dinámicamente los datos de la base de datos mediante mensajes DNS. En los documentos [RFC 2136] y [RFC 3007] se especifican las actualizaciones dinámicas de DNS.)

Una vez que haya completado estos pasos, los usuarios podrán visitar su sitio web y enviar correos electrónicos a los empleados de su empresa. Vamos a terminar esta exposición sobre DNS verificando que esta afirmación es cierta. Esta verificación también le ayudará a consolidar lo que



SEGURIDAD

VULNERABILIDADES DNS

Hemos visto que DNS es un componente fundamental de la infraestructura de Internet y que muchos servicios importantes, entre los que incluyen las aplicaciones web y de correo electrónico, no podrían funcionar sin él. Por tanto, lo natural es preguntarse: ¿Cómo puede ser atacado DNS? ¿Es DNS un blanco fácil, que espera a ser puesto fuera de servicio, arrastrando con él a la mayoría de las aplicaciones de Internet?

El primer tipo de ataque que nos viene a la mente es un ataque DDoS por de ancho de banda (véase la Sección 1.6) contra los servidores DNS. Por ejemplo, un atacante podría intentar enviar a cada uno de los servidores DNS raíz una gran cantidad de paquetes, tantos que la mayor parte de las consultas DNS legítimas nunca fueran contestadas. Un ataque DDoS a gran escala contra servidores DNS raíz tuvo lugar realmente el 21 de octubre de 2002. En ese ataque, los atacantes utilizaron una botnet para enviar enormes cargas de mensajes ping ICMP a las direcciones IP de 13 servidores DNS raíz. (Los mensajes ICMP se estudian en la Sección 5.6. Por el momento, nos basta con saber que los paquetes ICMP son tipos especiales de datagramas IP.) Afortunadamente, este ataque a gran escala causó unos daños mínimos, sin tener apenas impacto sobre la experiencia en Internet de los usuarios. Los atacantes dirigieron con éxito gran cantidad de paquetes a los servidores raíz, pero muchos de estos servidores estaban protegidos mediante mecanismos de filtrado de paquetes configurados para bloquear siempre todos los mensajes ping ICMP dirigidos a los mismos. Estos servidores protegidos estaban resguardados y funcionaron normalmente. Además, la mayoría de los servidores DNS locales tenían almacenadas en caché las direcciones IP de los servidores de dominio de nivel superior, permitiendo el procesamiento de consultas ignorando normalmente a los servidores DNS raíz.

Un ataque DDoS potencialmente más efectivo contra servidores DNS consistiría en enviar una gran cantidad de consultas DNS a los servidores de dominio de alto nivel; por ejemplo, a todos aquellos servidores que administren el dominio .com. Sería bastante complicado filtrar las consultas DNS dirigidas a servidores DNS; y los servidores de dominio de nivel superior no pueden ser puenteados tan fácilmente como los servidores raíz. Pero la severidad de un ataque así podría ser parcialmente mitigada por el almacenamiento en caché de los servidores DNS locales.

El sistema DNS también podría ser atacado, en teoría, de otras maneras. En un ataque por intermediación (*man-in-the-middle*), el atacante intercepta las consultas procedentes de los hosts y devuelve respuestas falsas. En el ataque por envenenamiento DNS, el atacante envía respuestas falsas a un servidor DNS, engañándole y haciendo que almacene en su caché registros falsos. Cualquiera de estos ataques podría utilizarse, por ejemplo, para redirigir a un usuario web inadvertido al sitio web del atacante. Sin embargo, estos ataques son difíciles de implementar, ya que requieren interceptar los paquetes o engañar a los servidores [Skoudis 2006].

En resumen, DNS ha demostrado ser sorprendentemente robusto frente a los ataques. Hasta el momento, no ha habido ningún ataque que haya conseguido interrumpir con éxito el servicio DNS.

ha aprendido sobre DNS. Suponga que Alicia se encuentra en Australia y desea ver la página web www.networkutopia.com. Como hemos explicado anteriormente, su host enviará en primer lugar una consulta DNS a su servidor DNS local, el cual a su vez se pondrá en contacto con un servidor TLD com. (El servidor DNS local también tendrá que comunicarse con un servidor DNS raíz si no tiene en caché la dirección de un servidor TLD com.) El servidor TLD contendrá los registros de recursos de tipo NS y de tipo A enumerados anteriormente, ya que la entidad registradora los habrá almacenado en todos los servidores TLD com. El servidor TLD com envía entonces una respuesta al servidor DNS local de Alicia, conteniendo dicha respuesta los dos registros de recursos. A continuación, el servidor DNS local transmite una consulta DNS a 212.212.212.1, solicitando

el registro de tipo A correspondiente a `www.networkutopia.com`. Este registro proporciona la dirección IP del servidor web deseado, por ejemplo, `212.212.71.4`, que el servidor DNS local pasa al host de Alicia. En este momento, el navegador de Alicia puede iniciar una conexión TCP con el host `212.212.71.4` y enviar una solicitud HTTP a través de la misma. Como ha podido ver, son muchas las cosas que suceden entre bastidores cuando navegamos por la Web.

2.5 Distribución de archivos P2P

Las aplicaciones descritas en este capítulo hasta el momento, incluyendo las aplicaciones web, de correo electrónico y DNS, emplean todas ellas arquitecturas cliente-servidor que dependen en gran medida de que exista una infraestructura de servidores siempre activos. Recuerde de la Sección 2.1.1 que con una arquitectura P2P no se depende más que mínimamente (o no se depende en absoluto) de que exista esa infraestructura de servidores siempre activos. En su lugar, una serie de parejas de hosts conectados de forma intermitente, denominados pares o *peers*, se comunican directamente entre sí. Los pares no son propiedad de un proveedor de servicios, sino que son computadoras de escritorio o portátiles controlados por los usuarios.

En esta sección vamos a ver una aplicación P2P muy corriente, la distribución de un archivo de gran tamaño desde un único servidor a muchos otros hosts (denominados pares). El archivo podría ser una nueva versión del sistema operativo Linux, un parche software para una aplicación o un sistema operativo existentes, un archivo de audio MP3 o un archivo de vídeo MPEG. En la distribución de archivos cliente-servidor, el servidor debe enviar una copia del archivo a cada uno de los pares, provocando una enorme sobrecarga en el servidor y consumiendo una gran cantidad de su ancho de banda. En la distribución de archivos P2P, cada par puede redistribuir cualquier parte del archivo que ha recibido a cualesquiera otros pares, ayudando de este modo al servidor a llevar a cabo el proceso de distribución. En 2016, el protocolo de distribución de archivos P2P más popular es BitTorrent. Originalmente, fue desarrollado por Bram Cohen, pero ahora existen muchos clientes BitTorrent independientes distintos que cumplen con el protocolo BitTorrent, al igual que existen diversos navegadores web que cumplen el protocolo HTTP. En esta subsección examinaremos en primer lugar la característica de auto-escalabilidad de las arquitecturas P2P en el contexto de la distribución de archivos. Después describiremos en detalle BitTorrent, destacando sus funcionalidades y características más importantes.

Escalabilidad de las arquitecturas P2P

Con el fin de comparar las arquitecturas cliente-servidor con las arquitecturas P2P y para ilustrar la auto-escalabilidad inherente de P2P, ahora vamos a considerar un modelo cuantitativo simple para la distribución de un archivo a un conjunto fijo de pares en ambos tipos de arquitectura. Como se muestra en la Figura 2.22, el servidor y los pares están conectados a Internet mediante enlaces de acceso. Sea u_s la velocidad de carga del enlace de acceso del servidor, u_i la velocidad de carga del enlace de acceso del par i y d_i la velocidad de descarga del enlace de acceso del par i . Sea F el tamaño en bits del archivo que se va a distribuir y N el número de pares que desean obtener una copia del archivo. El **tiempo de distribución** es el tiempo que tardan los N pares en obtener una copia del archivo. En el análisis del tiempo de distribución que proporcionamos a continuación, para ambas arquitecturas, cliente-servidor y P2P, hemos hecho una simplificación (pero generalmente precisa [Akella 2003]): suponer que el núcleo de Internet tiene el ancho de banda suficiente, lo que implica que todos los cuellos de botella se encuentran en el acceso a red. También hemos supuesto que el servidor y los clientes no están participando en ninguna otra aplicación de red, de modo que los anchos de banda para carga y descarga están dedicados completamente a distribuir el archivo.

En primer lugar, vamos a determinar el tiempo de distribución para la arquitectura cliente-servidor, el cual denotaremos como D_{cs} . En esta arquitectura, ninguno de los pares ayudan a distribuir el archivo. Tenemos que hacer las dos observaciones siguientes: