

3 Protocolos y funcionalidad de la capa de Aplicación

3.0 Introducción del capítulo

3.0.1 Introducción del capítulo

La mayoría de nosotros experimentamos Internet a través de World Wide Web, servicios de e-mail y programas para compartir archivos. Éstas y muchas otras aplicaciones proporcionan la interfaz humana a la red subyacente, lo que nos permite enviar y recibir información con relativa facilidad. Generalmente, las aplicaciones que utilizamos son intuitivas; es decir, podemos acceder a ellas y usarlas sin saber cómo funcionan. Sin embargo, para los profesionales de redes es importante conocer cómo una aplicación puede formatear, transmitir e interpretar mensajes que se envían y reciben a través de la red.

La visualización de los mecanismos que permiten la comunicación a través de la red se hace más sencilla si utilizamos el marco en capas del modelo [Interconexión de sistema abierto \(OSI\)](#). En este capítulo, enfatizaremos el rol de una capa, la capa de Aplicación, y sus componentes: aplicaciones, servicios y protocolos. Exploraremos cómo esos tres elementos hacen posible la comunicación sólida a través de la red de información.



3.1 Aplicaciones: La interfaz entre redes

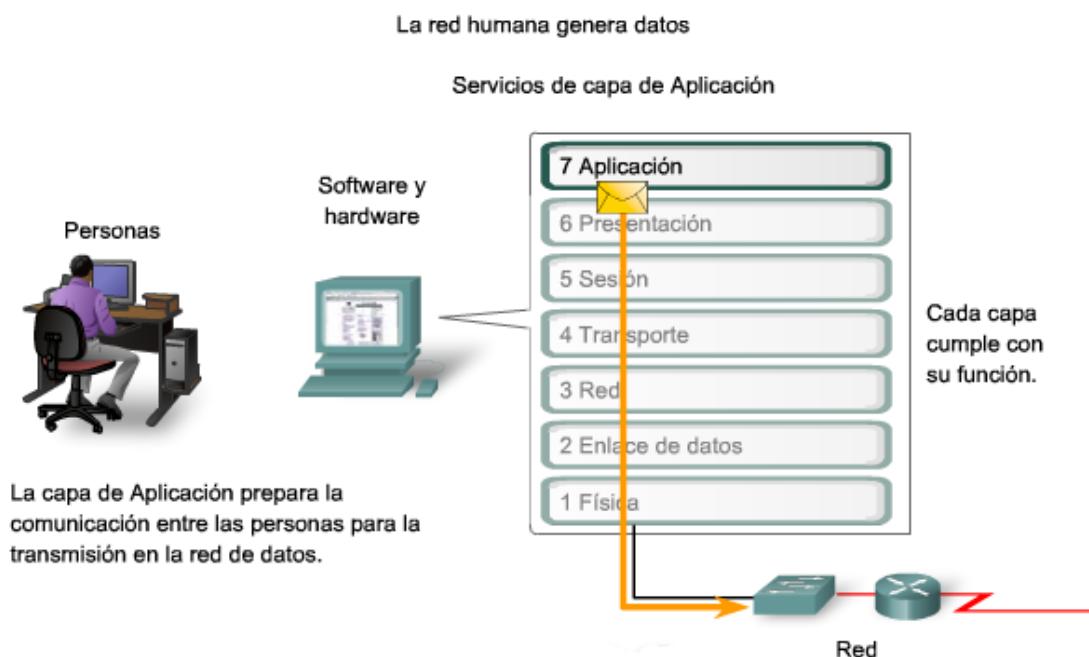
3.1.1 Modelo OSI y Modelo TCP/IP

El modelo de referencia de interconexión de sistemas abiertos es una representación abstracta en capas, creada como guía para el diseño del protocolo de red. El modelo OSI

divide el proceso de networking en diferentes capas lógicas, cada una de las cuales tiene una única funcionalidad y a la cual se le asignan protocolos y servicios específicos.

En este modelo, la información se pasa de una capa a otra, comenzando en la capa de Aplicación en el host de transmisión, siguiendo por la jerarquía hacia la capa Física, pasando por el canal de comunicaciones al host de destino, donde la información vuelve a la jerarquía y termina en la capa de Aplicación. La figura ilustra los pasos en este proceso.

La capa de Aplicación, Capa siete, es la capa superior de los modelos OSI y TCP/IP. Es la capa que proporciona la interfaz entre las aplicaciones que utilizamos para comunicarnos y la red subyacente en la cual se transmiten los mensajes. Los protocolos de capa de aplicación se utilizan para intercambiar los datos entre los programas que se ejecutan en los hosts de origen y destino. Existen muchos protocolos de capa de aplicación y siempre se desarrollan protocolos nuevos.



Aunque el grupo de protocolos TCP/IP se desarrolló antes de la definición del modelo OSI, la funcionalidad de los protocolos de capa de aplicación de TCP/IP se adaptan aproximadamente a la estructura de las tres capas superiores del modelo OSI: Capas de Aplicación, Presentación y Sesión.

La mayoría de los protocolos de capa de aplicación de TCP/IP se desarrollaron antes de la aparición de computadoras personales, interfaces del usuario gráficas y objetos multimedia. Como resultado, estos protocolos implementan muy poco de la funcionalidad que se especifica en las capas de Sesión y Presentación del modelo OSI.

Capa de Presentación

La capa de Presentación tiene tres funciones primarias:

- Codificación y conversión de datos de la capa de aplicación para garantizar que los datos del [dispositivo de origen](#) puedan ser interpretados por la aplicación adecuada en el dispositivo de destino.
- Compresión de los datos de forma que puedan ser descomprimidos por el dispositivo de destino.
- Encriptación de los datos para transmisión y descifre de los datos cuando se reciben en el destino.

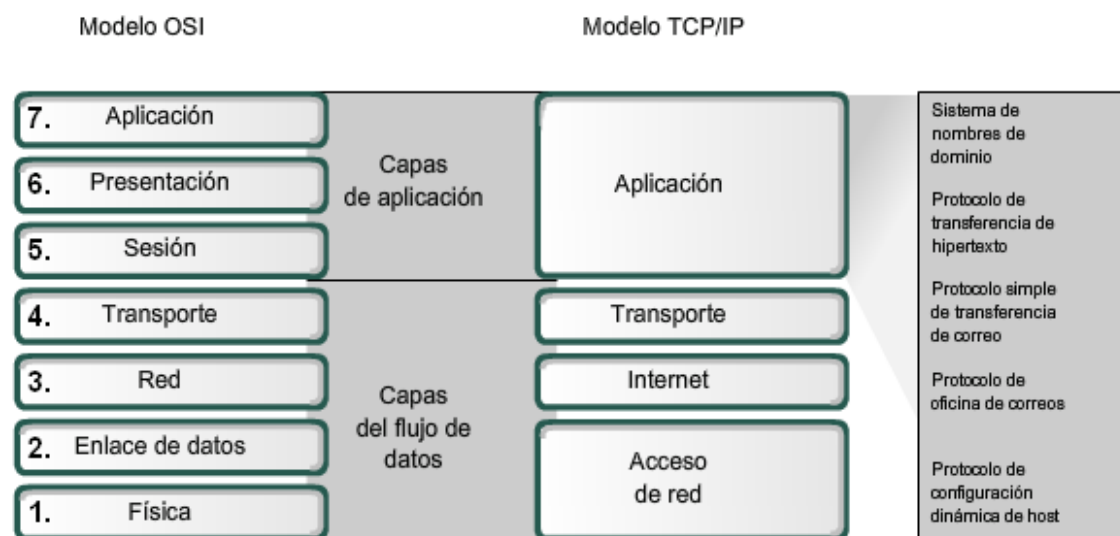
Las implementaciones de la capa de presentación generalmente no se vinculan con una stack de protocolos determinada. Los estándares para vídeos y gráficos son algunos ejemplos. Dentro de los estándares más conocidos para vídeo encontramos QuickTime y el Grupo de expertos en películas (MPEG). QuickTime es una especificación de Apple Computer para audio y vídeo, y MPEG es un estándar para la codificación y compresión de vídeos.

Dentro de los formatos de imagen gráfica más conocidos encontramos Formato de intercambio gráfico (GIF), Grupo de expertos en fotografía (JPEG) y Formato de archivo de imagen etiquetada (TIFF). GIF y JPEG son estándares de compresión y codificación para imágenes gráficas, y TIFF es una formato de codificación estándar para imágenes gráficas.

Capa de Sesión

Como lo indica el nombre de la capa de Sesión, las funciones en esta capa crean y mantienen diálogos entre las aplicaciones de origen y destino. La capa de sesión maneja el intercambio de información para iniciar los diálogos y mantenerlos activos, y para reiniciar sesiones que se interrumpieron o desactivaron durante un periodo de tiempo prolongado.

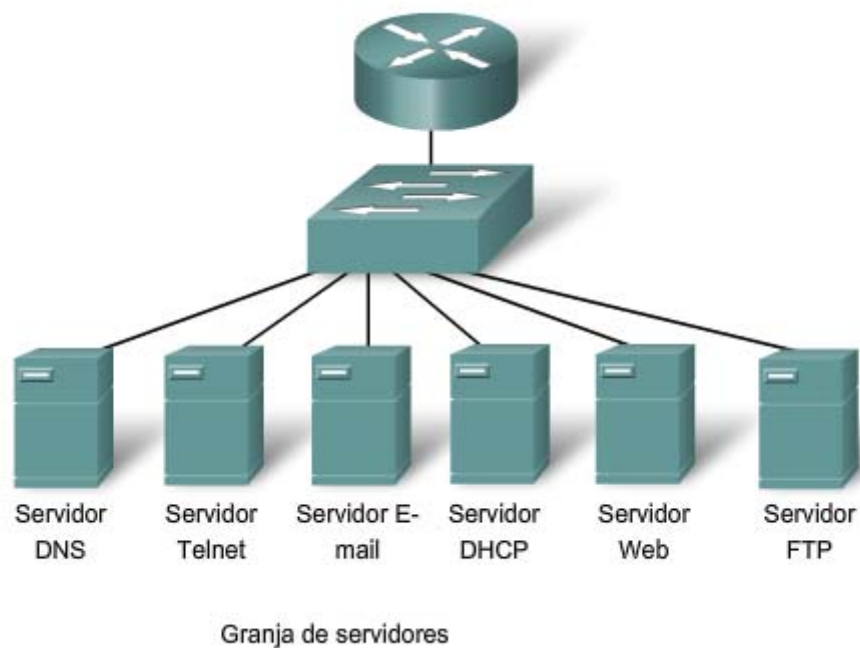
La mayoría de las aplicaciones, como los exploradores Web o los clientes de correo electrónico, incorporan la funcionalidad de las capas 5, 6 y 7 del modelo OSI.



Los protocolos de capa de aplicación de TCP/IP más conocidos son aquellos que proporcionan intercambio de la información del usuario. Estos protocolos especifican la información de control y formato necesaria para muchas de las funciones de comunicación de Internet más comunes. Algunos de los protocolos TCP/IP son:

- El protocolo Servicio de nombres de dominio (DNS, Domain Name Service) se utiliza para resolver nombres de Internet en direcciones IP.
- El protocolo de transferencia de hipertexto (HTTP, Hypertext Transfer Protocol) se utiliza para transferir archivos que forman las páginas Web de la World Wide Web.
- El [Protocolo simple de transferencia de correo \(SMTP\)](#) se utiliza para la transferencia de mensajes de correo y adjuntos.
- Telnet, un protocolo de emulación de terminal, se utiliza para proporcionar acceso remoto a servidores y a dispositivos de red.
- El Protocolo de transferencia de archivos (FTP, File Transfer Protocol) se utiliza para la transferencia interactiva de archivos entre sistemas.

Los protocolos de la suite TCP/IP generalmente son definidos por Solicitudes de comentarios (RFCs). El Grupo de trabajo de ingeniería de Internet mantiene las RFCs como los estándares para el conjunto TCP/IP.



3.1.2 Software de la capa de Aplicación

Las funciones asociadas con los protocolos de capa de Aplicación permiten a la red humana comunicarse con la red de datos subyacente. Cuando abrimos un explorador Web o una ventana de mensajería instantánea, se inicia una aplicación, y el programa se coloca en la memoria del dispositivo donde se ejecuta. Cada programa ejecutable cargado a un dispositivo se denomina proceso.

Dentro de la capa de Aplicación, existen dos formas de procesos o programas de software que proporcionan acceso a la red: aplicaciones y servicios.

Aplicaciones reconocidas por la red

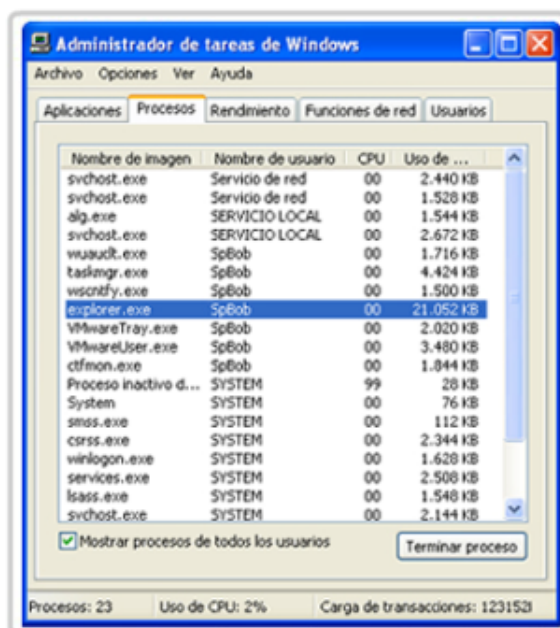
Aplicaciones son los programas de software que utiliza la gente para comunicarse a través de la red. Algunas aplicaciones de usuario final son compatibles con la red, lo cual significa que implementan los protocolos de la capa de aplicación y pueden comunicarse directamente con las capas inferiores del stack de protocolos. Los clientes de correo electrónico y los exploradores Web son ejemplos de este tipo de aplicaciones.

Servicios de la capa de Aplicación

Otros programas pueden necesitar la ayuda de los servicios de la capa de Aplicación para utilizar los recursos de la red, como transferencia de archivos o cola de impresión en red. Aunque son transparentes para el usuario, estos servicios son los programas que se comunican con la red y preparan los datos para la transferencia. Diferentes tipos de datos, ya sea texto, gráfico o vídeo, requieren de diversos servicios de red para asegurarse de que estén bien preparados para procesar las funciones de las capas inferiores del modelo OSI.

Cada servicio de red o aplicación utiliza protocolos que definen los estándares y formatos de datos a utilizarse. Sin protocolos, la red de datos no tendría una manera común de formatear y direccionar los datos. Para comprender la función de los distintos servicios de red, es necesario familiarizarse con los protocolos subyacentes que rigen su operación.

Procesos de software



Ejemplos de procesos en ejecución en el sistema operativo Windows

Los procesos son programas de software individuales que se ejecutan en forma simultánea.

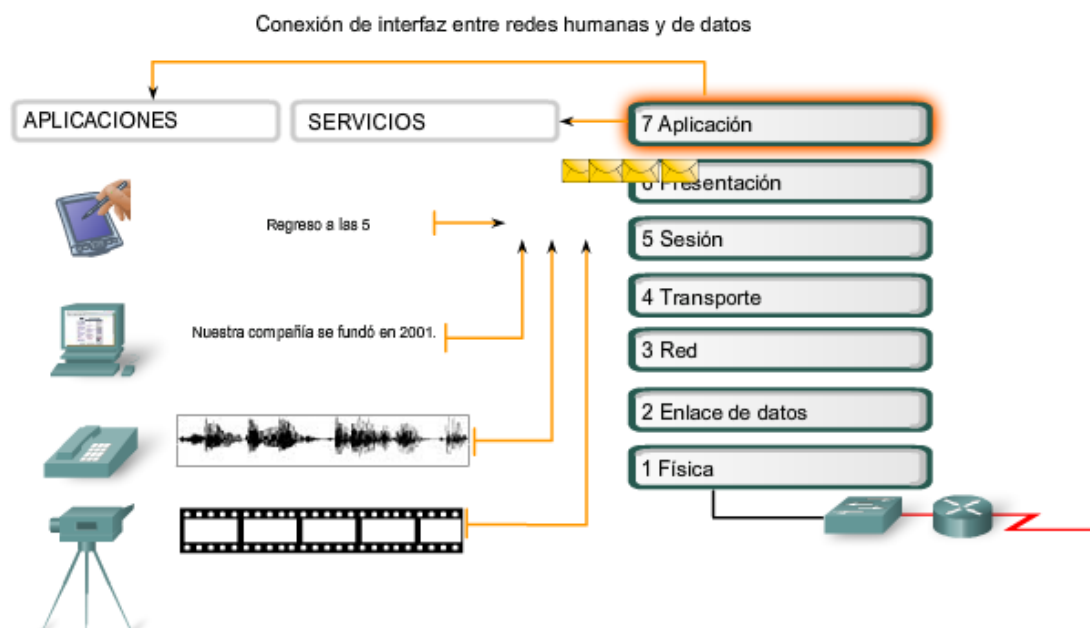
Los procesos pueden ser

- 1 Aplicaciones
- 2 Servicios
- 3 Operaciones del sistema
- 4 Un programa puede estar en ejecución varias veces, cada vez dentro de su propio proceso.

3.1.3 Aplicaciones del usuario, servicios y protocolos de capa de Aplicación

Como se mencionó anteriormente, la capa de Aplicación utiliza los protocolos implementados dentro de las aplicaciones y servicios. Mientras que las aplicaciones proporcionan a las personas una forma de crear mensajes y los servicios de la capa de aplicación establecen una interfaz con la red, los protocolos proporcionan las reglas y los formatos que regulan el tratamiento de los datos. Un único programa ejecutable debe utilizar los tres componentes e inclusive el mismo nombre. Por ejemplo: cuando analizamos "Telnet" nos podemos referir a la aplicación, el servicio o el protocolo.

En el modelo OSI, se considera que las aplicaciones que interactúan directamente con las personas se encuentran en la parte superior del stack, al igual que las personas. Al igual que todas las personas dentro del modelo OSI, la capa de Aplicación se basa en la funciones de las capas inferiores para completar el proceso de comunicación. Dentro de la capa de aplicación, los protocolos especifican qué mensajes se intercambian entre los host de origen y de destino, la [sintaxis](#) de los comandos de control, el tipo y formato de los datos que se transmiten y los métodos adecuados para notificación y recuperación de errores.



3.1.4 Funciones del protocolo de capa de Aplicación

Los protocolos de la capa de aplicación son utilizados tanto por los dispositivos de origen como de destino durante una [sesión](#) de comunicación. Para que las comunicaciones sean exitosas, deben coincidir los protocolos de capa de aplicación implementados en el host de origen y destino.

Los protocolos establecen reglas consistentes para intercambiar datos entre las aplicaciones y los servicios cargados en los dispositivos participantes. Los protocolos especifican cómo se estructuran los datos dentro de los mensajes y los tipos de mensajes que se envían entre origen y destino. Estos mensajes pueden ser solicitudes de servicios, acuses de recibo, mensajes de datos, mensajes de estado o mensajes de error. Los

protocolos también definen los diálogos de mensajes, asegurando que un mensaje enviado encuentre la respuesta esperada y se invoquen los servicios correspondientes cuando se realiza la transferencia de datos.

Muchos y diversos tipos de aplicaciones se comunican a través de las redes de datos. Por lo tanto, los servicios de la capa de Aplicación deben implementar protocolos múltiples para proporcionar la variedad deseada de experiencias de comunicación. Cada protocolo tiene un fin específico y contiene las características requeridas para cumplir con dicho propósito. Deben seguirse los detalles del protocolo correspondiente a cada capa, así las funciones en una capa se comunican correctamente con los servicios en la capa inferior.

Las aplicaciones y los servicios también pueden utilizar protocolos múltiples durante el curso de una comunicación simple. Un protocolo puede especificar cómo se establece la conexión de redes y otro describir el proceso para la transferencia de datos cuando el mensaje se pasa a la siguiente capa inferior.



3.2 Toma de medidas para las aplicaciones y servicios

3.2.1 El modelo cliente-servidor

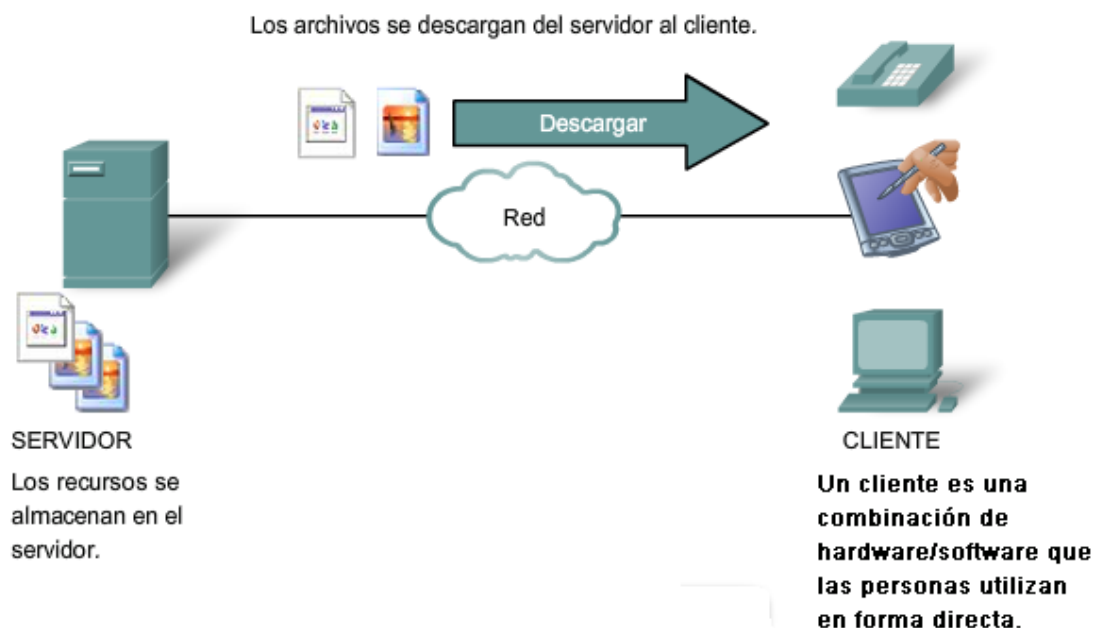
Cuando la gente intenta acceder a información en sus dispositivos, ya sean éstos una computadora personal o portátil, un PDA, teléfono celular o cualquier otro dispositivo conectado a la red, los datos pueden no estar físicamente almacenados en sus dispositivos. Si así fuere, se debe solicitar al dispositivo que contiene los datos, permiso para acceder a esa información.

Modelo cliente-servidor

En el [modelo cliente-servidor](#), el dispositivo que solicita información se denomina cliente y el dispositivo que responde a la solicitud se denomina servidor. Los procesos de cliente y servidor se consideran una parte de la capa de Aplicación. El cliente comienza el intercambio solicitando los datos al servidor, que responde enviando uno o más streams de datos al cliente. Los protocolos de capa de Aplicación describen el formato de las solicitudes y respuestas entre clientes y servidores. Además de la transferencia real de datos, este intercambio puede requerir de información adicional, como la autenticación del usuario y la identificación de un archivo de datos a transferir.

Un ejemplo de una red cliente/servidor es un entorno corporativo donde los empleados utilizan un servidor de e-mail de la empresa para enviar, recibir y almacenar e-mails. El cliente de correo electrónico en la computadora de un empleado emite una solicitud al servidor de e-mail para un mensaje no leído. El servidor responde enviando el e-mail solicitado al cliente.

Aunque los datos generalmente se describen como un flujo del servidor al cliente, algunos datos siempre fluyen del cliente al servidor. El flujo de datos puede ser el mismo en ambas direcciones o inclusive ser mayor en la dirección que va del cliente al servidor. Por ejemplo, un cliente puede transferir un archivo al servidor con fines de almacenamiento. **La transferencia de datos de un cliente a un servidor se conoce como [subida](#) y la de los datos de un servidor a un cliente, descarga.**



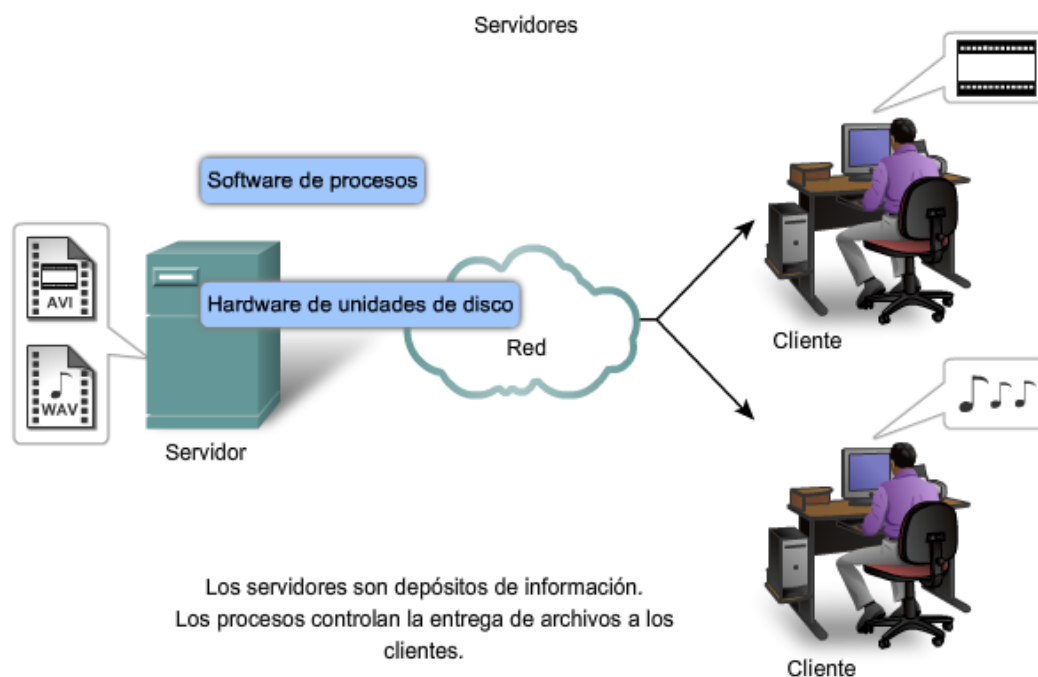
3.2.2 Servidores

En un contexto general de redes, cualquier dispositivo que responde a una solicitud de aplicaciones de cliente funciona como un servidor. Un servidor generalmente es una computadora que contiene información para ser compartida con muchos sistemas de cliente. Por ejemplo, páginas Web, documentos, bases de datos, imágenes, archivos de

audio y vídeo pueden almacenarse en un servidor y enviarse a los clientes que lo solicitan. En otros casos, como una impresora de red, el servidor de impresión envía las solicitudes de impresión del cliente a la impresora específica.

Diferentes tipos de aplicaciones del servidor tienen diferentes requerimientos para el acceso de clientes. Algunos servidores pueden requerir de autenticación de la información de cuenta del usuario para verificar si el usuario tiene permiso para acceder a los datos solicitados o para utilizar una operación en particular. Dichos servidores deben contar con una lista central de cuentas de usuarios y autorizaciones, o permisos (para operaciones y acceso a datos) otorgados a cada usuario. Cuando se utiliza un cliente FTP, por ejemplo, si usted solicita subir datos al servidor FTP, se le puede dar permiso para escribir su carpeta personal pero no para leer otros archivos del sitio.

En una red cliente-servidor, el servidor ejecuta un servicio o proceso, a veces denominado [daemon](#) de servidor. Al igual que la mayoría de los servicios, los daemons generalmente se ejecutan en segundo plano y no se encuentran bajo control directo del usuario. Los daemons se describen como servidores que "escuchan" una solicitud del cliente, porque están programados para responder cada vez que el servidor recibe una solicitud para el servicio proporcionado por el daemon. Cuando un daemon "escucha" una solicitud de un cliente, intercambia los mensajes adecuados con el cliente, según lo requerido por su protocolo, y procede a enviar los datos solicitados al cliente en el formato correspondiente.

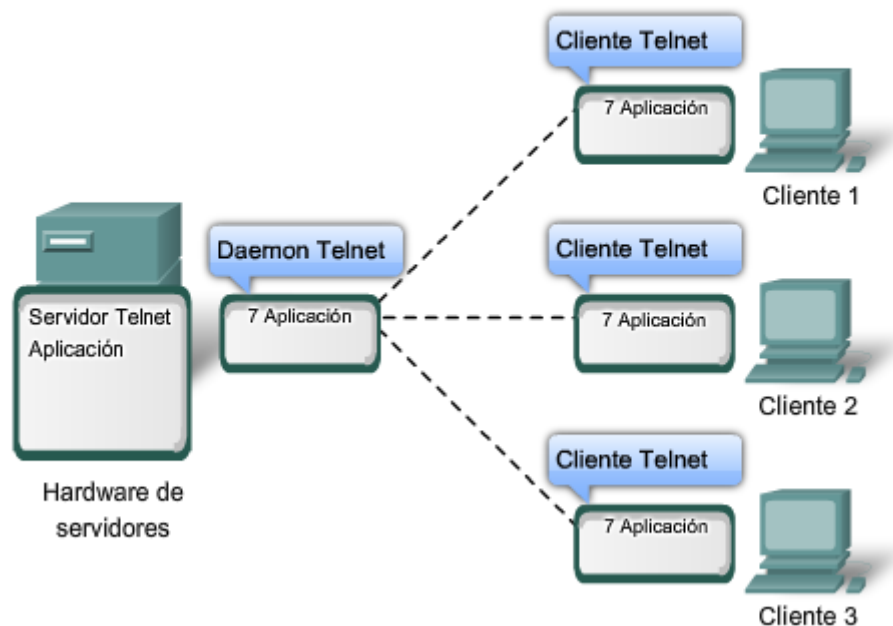


3.2.3 Protocolos y servicios de la capa de Aplicación

Una única aplicación puede emplear diferentes servicios de la capa de Aplicación, así lo que aparece para el usuario como una solicitud para una página Web puede, de hecho, ascender a docenas de solicitudes individuales. Y, para cada solicitud, pueden ejecutarse múltiples procesos. Por ejemplo, un cliente puede necesitar de diversos procesos individuales para formular sólo una solicitud al servidor.

Además, los servidores generalmente tienen múltiples clientes que solicitan información al mismo tiempo. Por ejemplo, un servidor **Telnet** puede tener varios clientes que requieren conectarse a él. Estas solicitudes individuales del cliente pueden manejarse en forma simultánea y separada para que la red sea exitosa. Los servicios y procesos de capa de Aplicación dependen del soporte de las funciones de la capa inferior para administrar en forma exitosa las múltiples conversaciones.

Los procesos de servidores pueden admitir múltiples clientes.



3.2.4 Redes y aplicaciones entre pares (P2P, Peer-to-Peer)

Modelo Punto a Punto

Además del modelo cliente/servidor para redes, existe también un modelo punto a punto. Las redes punto a punto tienen dos formas distintivas: diseño de redes punto a punto y aplicaciones punto a punto (P2P). Ambas formas tienen características similares pero en la práctica funcionan en forma muy distinta.

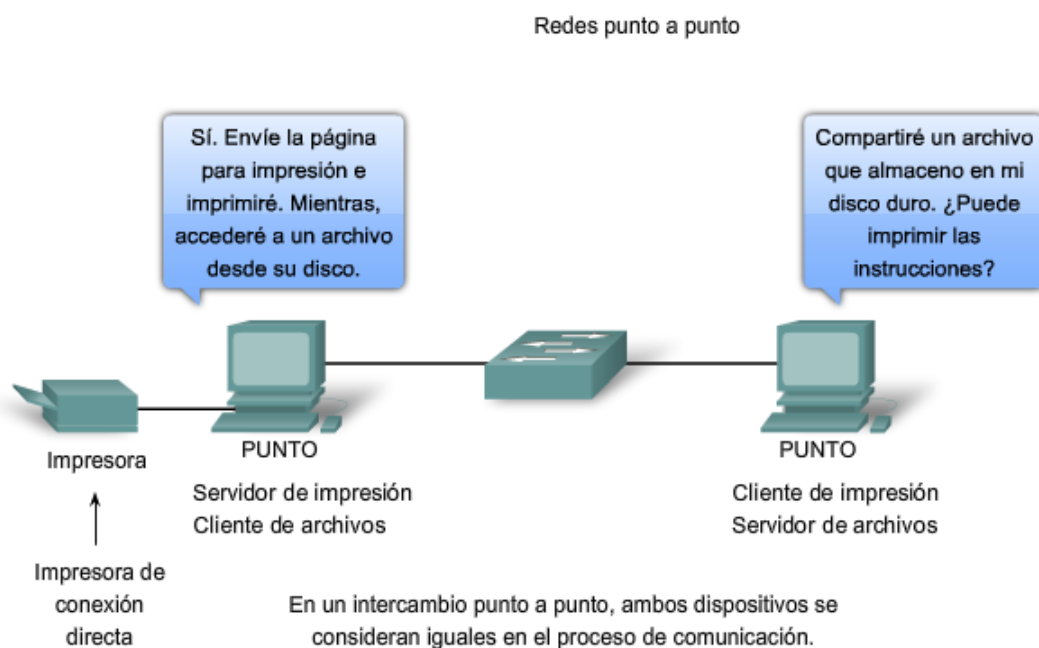
Redes entre pares

En una red entre pares, dos o más computadoras están conectadas a través de una red y pueden compartir recursos (por ejemplo, impresora y archivos) sin tener un servidor dedicado. Cada dispositivo final conectado (conocido como punto) puede funcionar como un servidor o como un cliente. Una computadora puede asumir el rol de servidor para una transacción mientras funciona en forma simultánea como cliente para otra transacción. Los roles del cliente y el servidor se configuran según las solicitudes.

Un ejemplo de una red entre pares es una simple red doméstica con dos computadoras conectadas que comparten una impresora. Cada persona puede configurar su computadora para compartir archivos, habilitar juegos en red o compartir una conexión de Internet. Otro ejemplo sobre la funcionalidad de la red punto a punto son dos

computadoras conectadas a una gran red que utilizan aplicaciones de software para compartir recursos entre ellas a través de la red.

A diferencia del modelo cliente/servidor, que utiliza servidores dedicados, las redes punto a punto descentralizan los recursos en una red. En lugar de ubicar información para compartir en los servidores dedicados, la información puede colocarse en cualquier parte de un dispositivo conectado. La mayoría de los sistemas operativos actuales admiten compartir archivos e impresoras sin requerir software del servidor adicional. Debido a que las redes punto a punto generalmente no utilizan cuentas de usuarios centralizadas, permisos ni monitores, es difícil implementar las políticas de acceso y seguridad en las redes que contienen mayor cantidad de computadoras. Se deben establecer cuentas de usuario y derechos de acceso en forma individual para cada dispositivo.



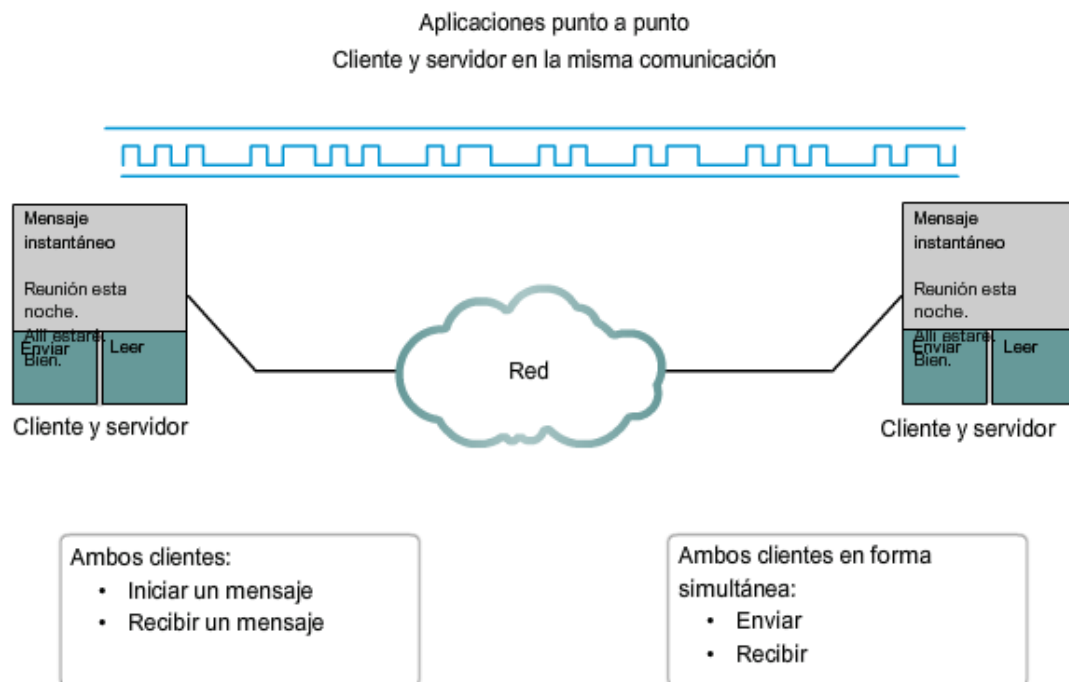
Aplicaciones punto a punto

Una aplicación punto a punto (P2P), a diferencia de una red punto a punto, permite a un dispositivo actuar como cliente o como servidor dentro de la misma comunicación. En este modelo, cada cliente es un servidor y cada servidor es un cliente. Ambos pueden iniciar una comunicación y se consideran iguales en el proceso de comunicación. Sin embargo, las aplicaciones punto a punto requieren que cada dispositivo final proporcione una interfaz de usuario y ejecute un servicio en segundo plano. Cuando inicia una aplicación punto a punto específica, ésta invoca la interfaz de usuario requerida y los servicios en segundo plano. Luego, los dispositivos pueden comunicarse directamente.

Algunas aplicaciones P2P utilizan un sistema híbrido donde se descentraliza el acceso a los recursos pero los índices que apuntan a las ubicaciones de los recursos están almacenados en un directorio centralizado. En un sistema híbrido, cada punto accede a

un servidor de índice para alcanzar la ubicación de un recurso almacenado en otro punto. El servidor de índice también puede ayudar a conectar dos puntos, pero una vez conectados, la comunicación se lleva a cabo entre los dos puntos, sin comunicación adicional al servidor de índice.

Las aplicaciones punto a punto pueden utilizarse en las redes punto a punto, en redes cliente/servidor y en Internet.



3.3 Ejemplos de servicios y protocolos de la capa de Aplicación

3.3.1 Protocolo y servicios DNS

Ahora que comprendemos mejor cómo las aplicaciones proporcionan una interfaz para el usuario y acceso a la red, veremos algunos protocolos específicos que se utilizan comúnmente.

Como veremos más adelante, la capa de transporte utiliza un [esquema](#) de direccionamiento que se llama número de puerto. Los números de puerto identifican las aplicaciones y los servicios de la capa de Aplicación que son los datos de origen y destino. Los programas del servidor generalmente utilizan números de puerto predefinidos comúnmente conocidos por los clientes. Mientras examinamos los diferentes servicios y protocolos de la capa de Aplicación de TCP/IP, nos referiremos a los números de puerto TCP y UDP normalmente asociados con estos servicios. Algunos de estos servicios son:

- [Sistema de nombres de dominio \(DNS\)](#): puerto TCP/UDP 53.
- Protocolo de transferencia de hipertexto (HTTP, Hypertext Transfer Protocol): puerto TCP 80.

- Protocolo simple de transferencia de correo (SMTP, Simple Mail Transfer Protocol): puerto TCP 25.
- [Protocolo de oficina de correos \(POP\)](#): puerto UDP 110.
- Telnet: puerto TCP 23.
- Protocolo de configuración dinámica de host: puerto UDP 67.
- Protocolo de transferencia de archivos (FTP, File Transfer Protocol): puertos TCP 20 y 21.

DNS

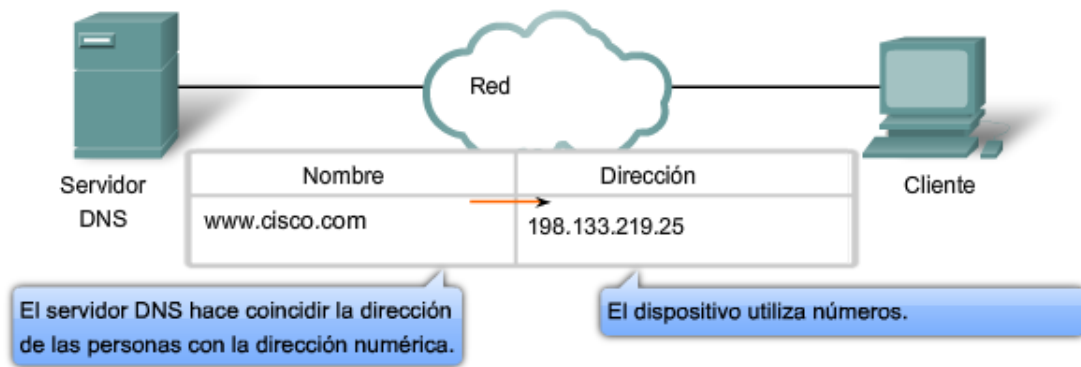
En redes de datos, los dispositivos son rotulados con direcciones IP numéricas para que puedan participar en el envío y recepción de mensajes a través de la red. Sin embargo, la mayoría de las personas pasan mucho tiempo tratando de recordar estas direcciones numéricas. Por lo tanto, los nombres de dominio fueron creados para convertir las direcciones numéricas en nombres simples y reconocibles.

En Internet, esos nombres de dominio, como www.cisco.com, son mucho más sencillos de recordar que 198.133.219.25, que es la dirección numérica real para este servidor. Además, si Cisco decide cambiar la dirección numérica, para el usuario es transparente ya que el nombre de dominio seguirá siendo www.cisco.com. La nueva dirección simplemente estará enlazada con el nombre de dominio existente y la conectividad se mantendrá. Cuando las redes eran pequeñas, resultaba fácil mantener la asignación entre los nombres de dominios y las direcciones que representaban. Sin embargo, a medida que las redes y el número de dispositivos comenzó a crecer, el sistema manual dejó de ser práctico.

El Sistema de nombres de dominio (DNS) se creó para que el nombre del dominio busque soluciones para estas redes. DNS utiliza un conjunto distribuido de servidores para resolver los nombres asociados con estas direcciones numéricas.

El **protocolo DNS** define un servicio automatizado que coincide con nombres de recursos que tienen la [dirección de red](#) numérica solicitada. Incluye las consultas sobre formato, las respuestas y los formatos de datos. Las comunicaciones del protocolo DNS utilizan un formato simple llamado mensaje. Este formato de mensaje se utiliza para todos los tipos de solicitudes de clientes y respuestas del servidor, mensajes de error y para la transferencia de información de [registro de recursos](#) entre servidores.

Resolución de direcciones DNS



DNS es un servicio cliente/servidor; sin embargo, difiere de los otros servicios cliente/servidor que estamos examinando. Mientras otros servicios utilizan un cliente que es una aplicación (como un explorador Web o un cliente de correo electrónico), el cliente DNS ejecuta un servicio por sí mismo. El cliente DNS, a veces denominado [resolución DNS](#), admite resolución de nombre para otras aplicaciones de red y servicios que lo necesiten.

Al configurar un dispositivo de red, generalmente proporcionamos una o más direcciones del **servidor DNS** que el cliente DNS puede utilizar para la resolución de nombres. En general, el proveedor de servicios de Internet provee las direcciones para utilizar con los servidores DNS. Cuando una aplicación de usuario solicita conectarse con un dispositivo remoto por nombre, el cliente DNS solicitante envía una petición a uno de esos servidores de nombre para resolver el nombre en una dirección numérica.

Los sistemas operativos informáticos también tienen una utilidad denominada [nslookup](#) que permite al usuario [consultar](#) manualmente los servidores de nombre para resolver un determinado nombre de host. Esta utilidad también puede utilizarse para resolver los problemas de resolución de nombres y verificar el estado actual de los servidores de nombres.

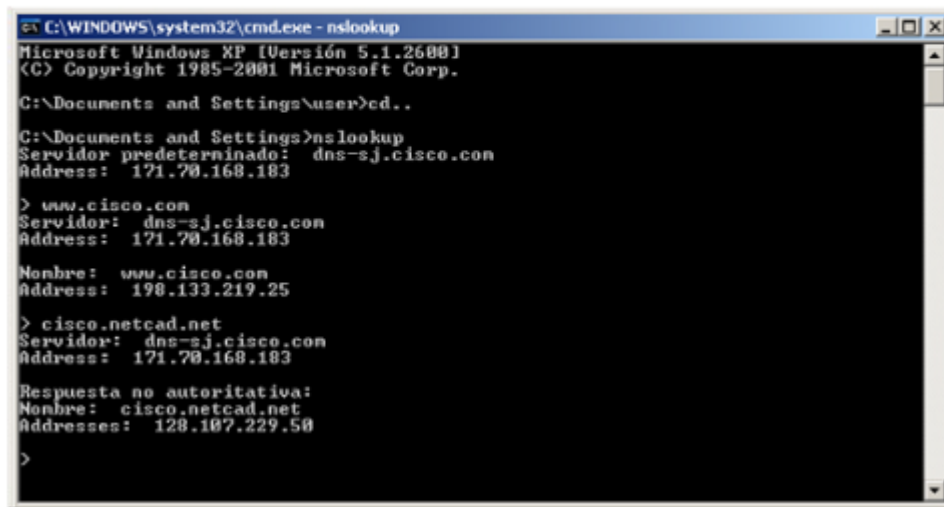
En la figura, cuando se ejecuta **nslookup**, se muestra el servidor DNS por defecto configurado para su host. En este ejemplo, el servidor DNS es dns-sjk.cisco.com que tiene una dirección de 171.68.226.120.

Luego podemos escribir el nombre de un host o dominio para el cual deseamos obtener la dirección. En la primera consulta de la figura, se hace una consulta para www.cisco.com. El servidor de nombre que responde proporciona la dirección 198.133.219.25.

Las consultas mostradas en la figura son sólo pruebas simples. La utilidad **nslookup** tiene muchas opciones disponibles para lograr una extensa verificación y prueba del

proceso

Uso de nslookup



```
C:\WINDOWS\system32\cmd.exe - nslookup
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\user>cd..
C:\Documents and Settings>nslookup
Servidor predeterminado: dns-sj.cisco.com
Address: 171.70.168.183

> www.cisco.com
Servidor: dns-sj.cisco.com
Address: 171.70.168.183

Nombre: www.cisco.com
Address: 198.133.219.25

> cisco.netcad.net
Servidor: dns-sj.cisco.com
Address: 171.70.168.183

Respuesta no autoritativa:
Nombre: cisco.netcad.net
Addresses: 128.107.229.50

>
```

DNS.

Un servidor DNS proporciona la resolución de nombres utilizando el daemon de nombre que generalmente se llama named (se pronuncia name-dee).

El servidor DNS almacena diferentes tipos de registros de recursos utilizados para resolver nombres. Estos registros contienen el nombre, la dirección y el tipo de registro.

Algunos de estos tipos de registro son:

- A: una dirección de un dispositivo final.
- NS: un servidor de nombre [autoritativo](#).
- CNAME: el nombre ideal (o Nombre de dominio completamente calificado) para un alias, que se utiliza cuando varios servicios tienen una única dirección de red pero cada servicio tiene su propia entrada en DNS.
- MX: registro de intercambio de correos, asigna un [nombre de dominio](#) a una lista de servidores de intercambio de correos para ese dominio.

Cuando un cliente realiza una consulta, el proceso "named" del servidor primero observa en sus propios registros para ver si puede resolver el nombre. Si no puede resolver el nombre utilizando los registros almacenados, contacta a otros servidores para hacerlo.

La solicitud puede pasar por un número de servidores, lo cual lleva tiempo adicional y consume ancho de banda. Una vez que se encuentra una coincidencia y se devuelve al servidor solicitante original, el servidor almacena temporalmente en la [caché](#) la dirección numerada que coincide con el nombre.

Si vuelve a solicitarse ese mismo nombre, el primer servidor puede regresar la dirección utilizando el valor almacenado en el caché de nombres. El almacenamiento en caché reduce el tráfico de la red de datos de consultas DNS y las cargas de trabajo de los servidores más altos de la jerarquía. El servicio del cliente DNS en las PC de Windows

optimiza el rendimiento de la resolución de nombres DNS almacenando previamente los nombres resueltos en la memoria. El comando **ipconfig /displaydns** muestra todas las entradas DNS en caché en un sistema informático con Windows XP o 2000.

Formato del mensaje DNS

DNS utiliza el mismo formato de mensaje para:

- todos los tipos de consultas de clientes y respuestas de servidor
- mensajes de error
- la transferencia de información de registros de recursos entre servidores

Encabezado	
Pregunta	La pregunta para el servidor de nombres
Respuesta	Registros de recursos que responden la pregunta
Autoridad	Registros de recursos que apuntan a una autoridad
Adicional	Registros de recursos que poseen información adicional

El sistema de nombres de dominio utiliza un sistema jerárquico para crear una base de datos para proporcionar una resolución de nombres. La jerarquía es similar a un árbol invertido con la raíz en la parte superior y las ramas por debajo.

En la parte superior de la jerarquía, los servidores raíz mantienen registros sobre cómo alcanzar los servidores de dominio de nivel superior, los cuales a su vez tienen registros que apuntan a los servidores de dominio de nivel secundario y así sucesivamente.

Los diferentes dominios de primer nivel representan el tipo de organización o el país de origen. Algunos ejemplos de dominios de primer nivel son:

- **.au: Australia**
- **.co: Colombia**
- **.com: una empresa o industria**
- **.jp: Japón**
- **.org: una organización sin fines de lucro**

Después de los dominios de primer nivel se encuentran los dominios de segundo nivel y, debajo de estos, hay otros dominios de nivel inferior.

Cada nombre de dominio es una ruta a través de este árbol invertido que comienza desde la raíz.

Por ejemplo: como se muestra en la figura, el servidor DNS raíz puede no saber exactamente dónde se encuentra el servidor de correo electrónico mail.cisco, pero lleva

un registro de los dominios "com" dentro de los dominios de primer nivel. Asimismo, los servidores dentro del dominio "com" pueden no tener un registro de mail.cisco.com, pero sí tienen un registro para el dominio "cisco.com". Los servidores dentro del dominio cisco.com tienen un registro (un registro MX para ser exactos) para mail.cisco.com.

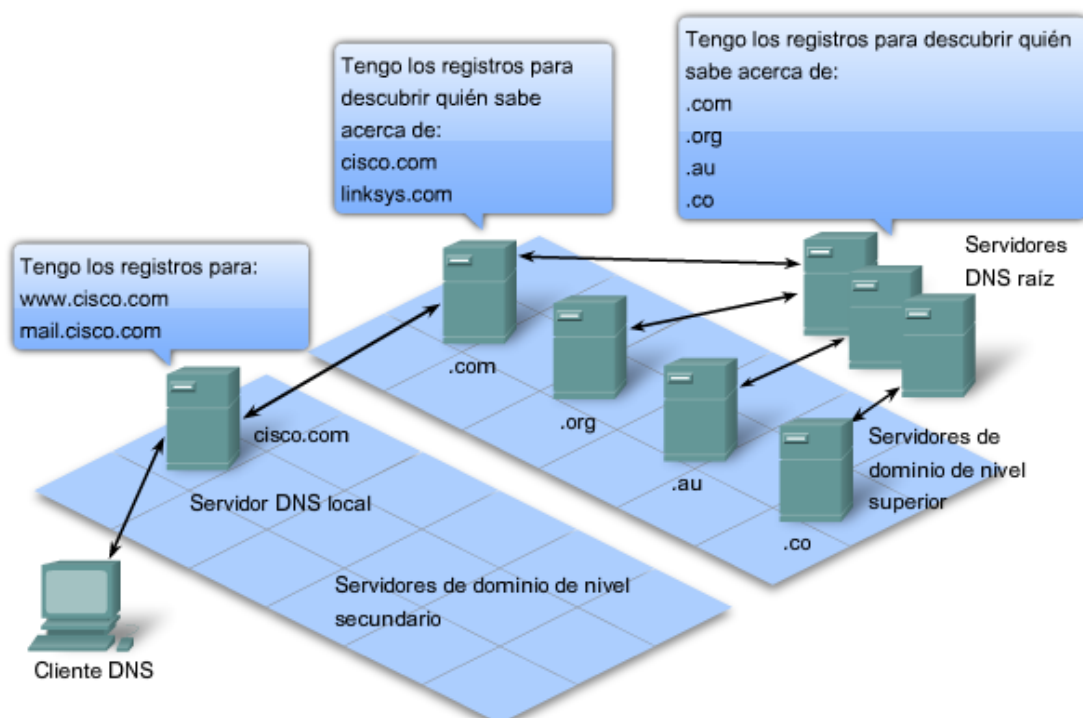
El sistema de nombres de dominio depende de esta jerarquía de servidores descentralizados y mantiene estos registros de recursos. Los registros de recursos enumeran nombres de dominios que el servidor puede resolver y servidores alternativos que también pueden procesar solicitudes. Si un determinado servidor tiene registros de recursos que corresponden a su nivel en la jerarquía de dominios, se dice que es **autoritativo** para esos registros.

Por ejemplo: un servidor de nombres en el dominio cisco.netacad.net no sería autoritativo para el registro mail.cisco.com porque ese registro se mantiene en un servidor de nivel de dominio superior, específicamente el servidor de nombres en el dominio cisco.com .

Enlaces

<http://www.ietf.org/rfc/rfc1034.txt>

<http://www.ietf.org/rfc/rfc1035.txt>



Una jerarquía de servidores DNS contiene los registros de recursos que coordinan los nombres con las direcciones.

3.3.2 Servicio WWW y HTTP

Cuando se escribe una dirección Web (o URL) en un explorador de Internet, el explorador establece una conexión con el servicio Web del servidor que utiliza el protocolo HTTP. URL (o Localizador uniforme de recursos) y URI (Identificador uniforme de recursos) son los nombres que la mayoría de las personas asocian con las direcciones Web.

El URL <http://www.cisco.com/index.html> es un ejemplo de un URL que se refiere a un recurso específico: una página Web denominada **index.html** en un servidor identificado como **cisco**.

Los exploradores Web son las aplicaciones de cliente que utilizan nuestras computadoras para conectarse con la World Wide Web y para acceder a los recursos almacenados en un servidor Web. Al igual que con la mayoría de los procesos de servidores, el servidor Web funciona como un servicio básico y genera diferentes tipos de archivos disponibles.

Para acceder al contenido, los clientes Web realizan conexiones al servidor y solicitan los recursos deseados. El servidor responde con los recursos y, una vez recibidos, el explorador interpreta los datos y los presenta al usuario.

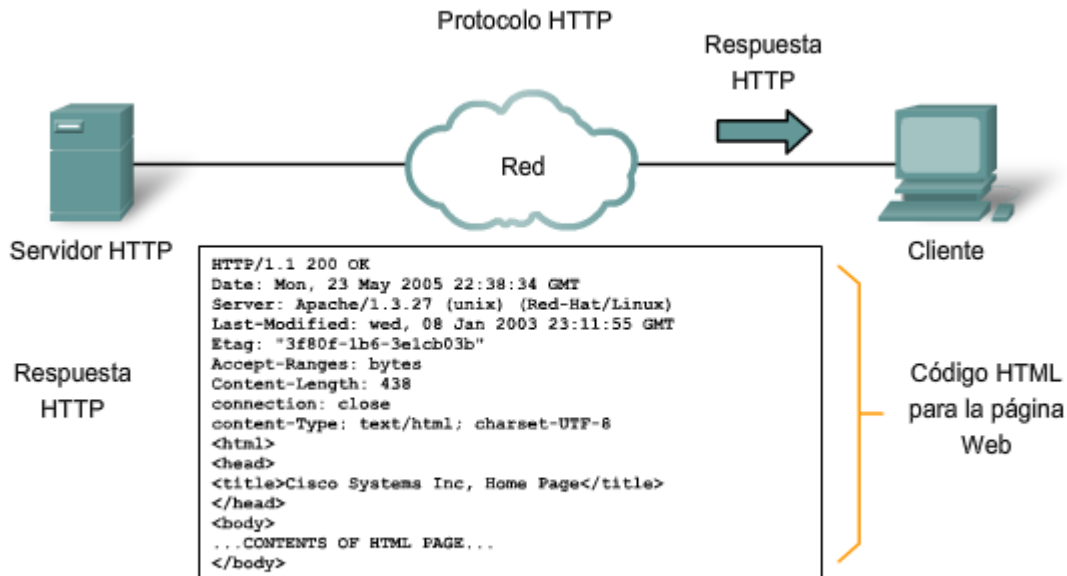
Los exploradores pueden interpretar y presentar muchos tipos de datos, como texto sin formato o Lenguaje de marcado de hipertexto (HTML, el lenguaje que se utiliza para construir una página Web). Otros tipos de datos, sin embargo, requieren de otro servicio o programa. Generalmente se los conoce como plug-ins o complementos. Para ayudar al explorador a determinar qué tipo de archivo está recibiendo, el servidor especifica qué clase de datos contiene el archivo.

Para comprender mejor cómo interactúan el explorador Web con el cliente Web, podemos analizar cómo se abre una página Web en un explorador. Para este ejemplo, utilizaremos la dirección URL: <http://www.cisco.com/web-server.htm>.

Primero, el explorador interpreta las tres partes de la URL:

1. **http** (el protocolo o esquema),
2. www.cisco.com (el nombre del servidor), y
3. **web-server.htm** (el nombre específico del archivo solicitado).

El explorador luego verifica con un servidor de nombres para convertir a www.cisco.com en una dirección numérica que utilizará para conectarse con el servidor. Al utilizar los requerimientos del protocolo HTTP, el explorador envía una solicitud GET al servidor y pide el archivo **web-server.htm**. El servidor, a su vez, envía al explorador el [código](#) HTML de esta página Web. Finalmente, el explorador descifra el código HTML y da formato a la página para la ventana del explorador.



En respuesta a la solicitud, el servidor HTTP envía el código para una página Web.

El protocolo de transferencia de hipertexto (HTTP), uno de los protocolos del grupo TCP/IP, se desarrolló en sus comienzos para publicar y recuperar las páginas HTML, y en la actualidad se utiliza para sistemas de información distribuidos y de colaboración. HTTP se utiliza a través de la World Wide Web para transferencia de datos y es uno de los protocolos de aplicación más utilizados.

HTTP especifica un protocolo de solicitud/respuesta. Cuando un cliente, generalmente un explorador Web, envía un mensaje de solicitud a un servidor, el protocolo HTTP define los tipos de mensajes que el cliente utiliza para solicitar la página Web y envía los tipos de mensajes que el servidor utiliza para responder. Los tres tipos de mensajes más comunes son GET, POST y PUT.

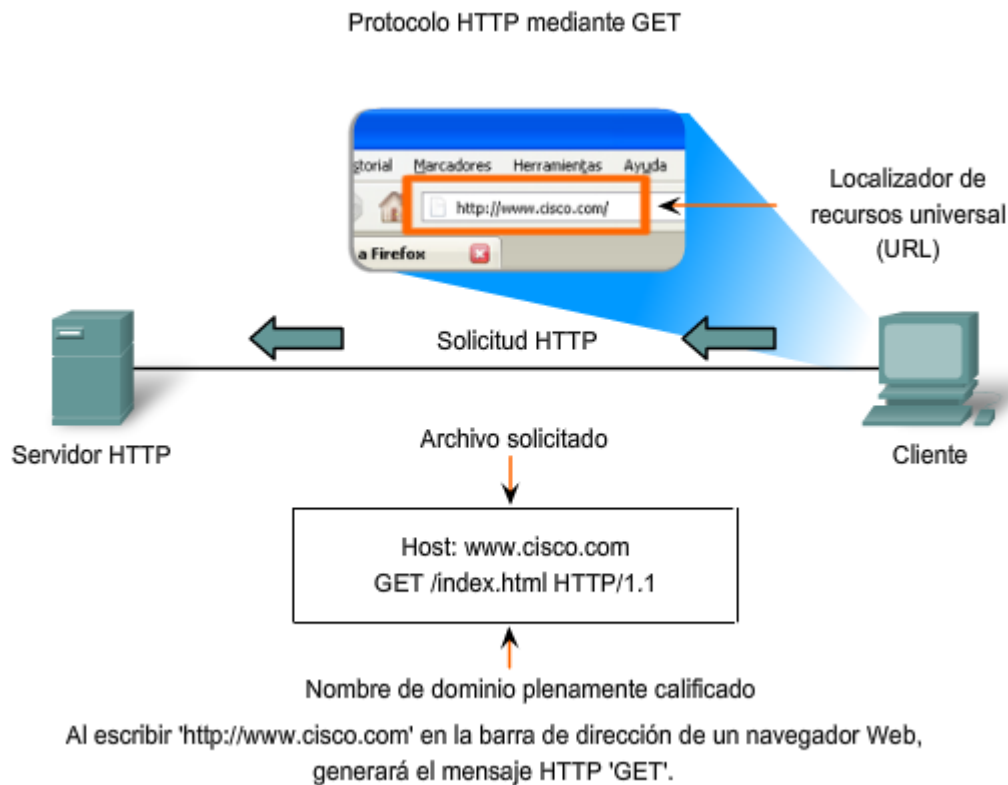
GET es una solicitud de datos del cliente. Un explorador Web envía el mensaje **GET** para solicitar las páginas desde un servidor Web. Como se muestra en la figura, una vez que el servidor recibe la solicitud **GET**, responde con una línea de estado, como HTTP/1.1 200 OK, y un mensaje solo, cuyo cuerpo puede ser el archivo solicitado, un mensaje de error o alguna otra información.

POST y **PUT** se utilizan para enviar mensajes que cargan los datos al servidor Web. Por ejemplo, cuando el usuario ingresa datos en un formulario incorporado en una página Web, **POST** incluye los datos en el mensaje enviado al servidor.

PUT carga los recursos o el contenido al servidor Web.

Aunque es muy flexible, HTTP no es un protocolo seguro. Los mensajes **POST** cargan información al servidor en un texto sin formato que puede ser interceptado y leído. De forma similar, las respuestas del servidor, generalmente páginas HTML, también son descifradas.

Para una comunicación segura a través de Internet, se utiliza el protocolo HTTP seguro (HTTPS) para acceder o subir información al servidor Web. HTTPS puede utilizar autenticación y [encriptación](#) para asegurar los datos cuando viajan entre el cliente y el servidor. HTTPS especifica reglas adicionales para pasar los datos entre la capa de Aplicación y la capa de Transporte.

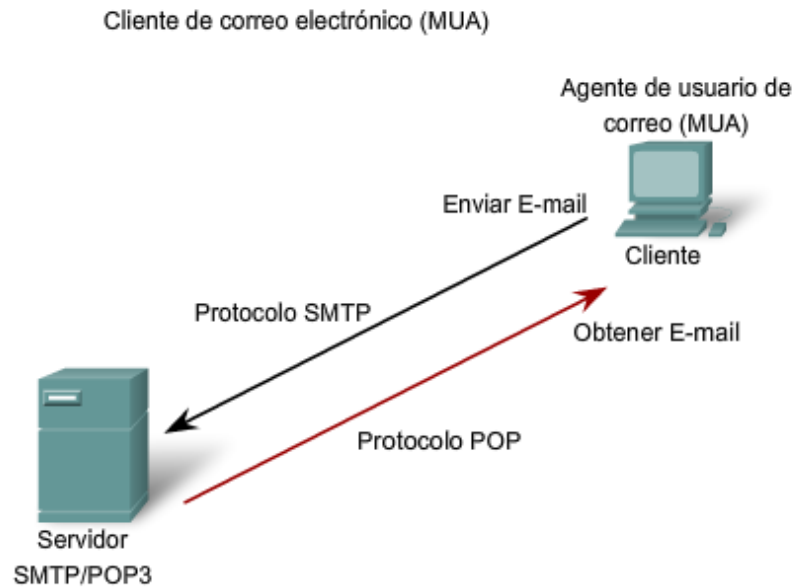


3.3.3 Servicios de e-mail y protocolos SMTP/POP

E-mail, el servidor de red más conocido, ha revolucionado la manera en que nos comunicamos, por su simpleza y velocidad. Inclusive para ejecutarse en una computadora o en otro dispositivo, los e-mails requieren de diversos servicios y aplicaciones. Dos ejemplos de protocolos de capa de aplicación son Protocolo de oficina de correos (POP) y **Protocolo simple de transferencia de correo (SMTP)**, que aparecen en la figura. Como con HTTP, estos protocolos definen procesos cliente-servidor.

Cuando una persona escribe mensajes de correo electrónico, generalmente utiliza una aplicación denominada [Agente de usuario de correo \(MUA\)](#) o cliente de correo electrónico. MUA permite enviar los mensajes y colocar los mensajes recibidos en el buzón del cliente; ambos procesos son diferentes.

Para recibir e-mails desde un servidor de e-mail, el cliente de correo electrónico puede utilizar un POP. Al enviar un e-mail desde un cliente o un servidor, se utilizan formatos de mensajes y cadenas de comando definidas por el protocolo SMTP. En general, un cliente de correo electrónico proporciona la funcionalidad de ambos protocolos dentro de una aplicación.



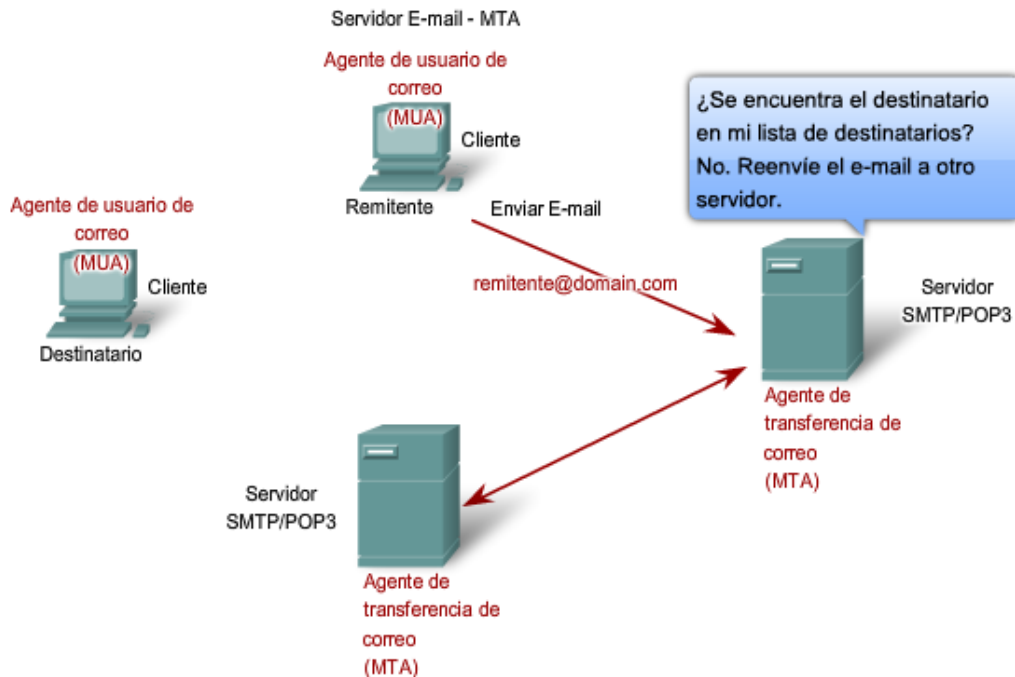
Los clientes envían e-mails a un servidor mediante SMTP y reciben e-mails mediante POP3.

Procesos del servidor de e-mail: MTA y MDA

El servidor de e-mail ejecuta dos procesos individuales:

- Agente de transferencia de correo (MTA, Mail Transfer Agent).
- Agente de entrega de correo (MDA, Mail Delivery Agent).

El proceso Agente de transferencia de correo (MTA) se utiliza para enviar correos electrónicos. Como se muestra en la figura, el MTA recibe mensajes desde el MUA u otro MTA en otro servidor de e-mail. Según el encabezado del mensaje, determina cómo debe reenviarse un mensaje para llegar a destino. Si el correo está dirigido a un usuario cuyo buzón está en el servidor local, el correo se pasa al MDA. Si el correo es para un usuario que no está en el servidor local, el MTA enruta el e-mail al MTA en el servidor correspondiente.



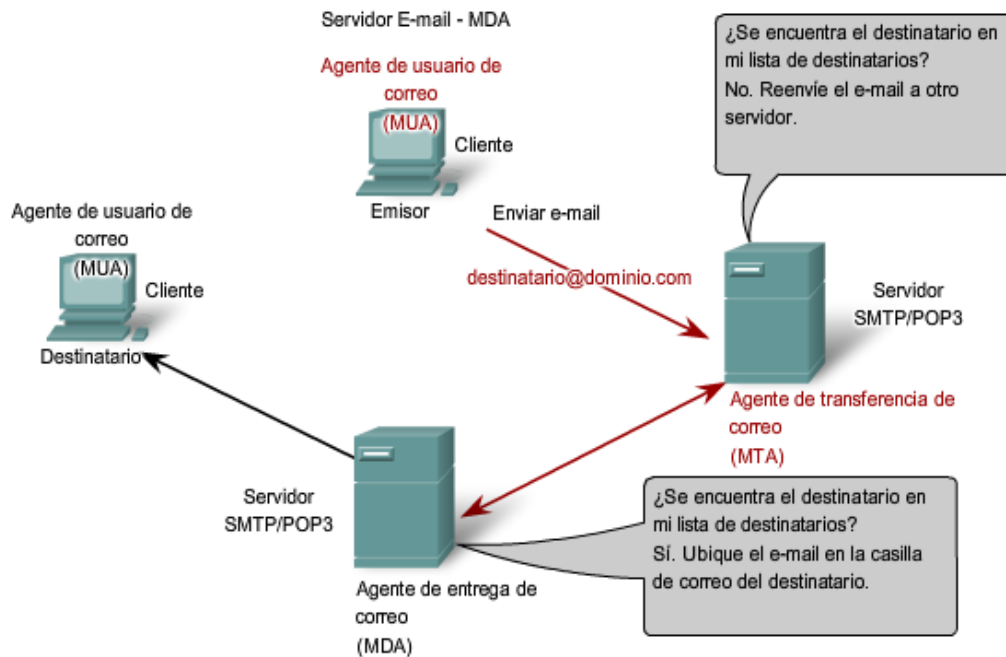
El proceso de agente de transferencia de correo rige el manejo de e-mails entre servidores.

En la figura, vemos que el Agente de envío de correo (MDA) acepta una parte del e-mail desde un Agente de transferencia de correo (MTA) y realiza el envío real. El MDA recibe todo el correo entrante desde el MTA y lo coloca en los buzones de los usuarios correspondientes. El MDA también puede resolver temas de entrega final, como análisis de virus, [correo no deseado filtrado](#) y manejo de acuses de recibo. La mayoría de las comunicaciones de e-mail utilizan las aplicaciones MUA, MTA y MDA. Sin embargo, existen otras alternativas para enviar e-mails.

El cliente puede estar conectado a un sistema de e-mails corporativo, como Lotus Notes de IBM, Groupwise de Novell o Microsoft Exchange. Estos sistemas a veces tienen su propio formato interno de correo electrónico y sus clientes generalmente se comunican con el servidor de correo electrónico a través de un protocolo propietario.

El servidor envía o recibe correos electrónicos por Internet a través de la [gateway](#) de correo de internet del producto, que realiza el reformato que sea necesario. Si, por ejemplo, dos personas que trabajan para la misma empresa intercambian e-mails entre ellos utilizando un protocolo propietario, los mensajes pueden permanecer completamente dentro del sistema de e-mails corporativo de la empresa.

Como segunda alternativa, las computadoras que no tienen un MUA pueden conectarse a un servicio de correo en un explorador Web para así recuperar y enviar mensajes. Algunas computadoras pueden ejecutar su propio MTA y administrar e-mails de dominio interno.



El proceso de agente de entrega de correo rige la entrega de e-mails entre servidores y clientes.

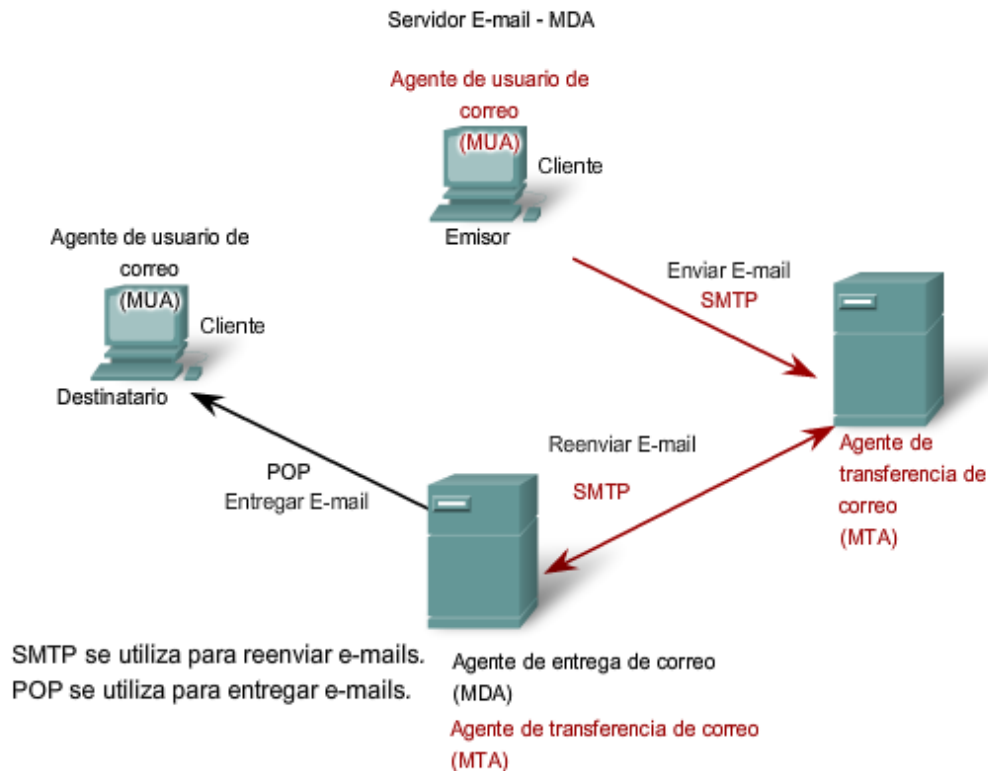
Como se mencionó anteriormente, los e-mails pueden utilizar los protocolos POP y SMTP (vea la figura para saber cómo funcionan). POP y POP3 (Protocolo de oficina de correos v.3) son protocolos de **envío** de correo entrante y protocolos cliente/servidor típicos. Envían e-mails desde el servidor de e-mail al cliente (MUA). El MDA escucha cuando un cliente se conecta a un servidor. Una vez establecida la conexión, el servidor puede enviar el e-mail al cliente.

El protocolo simple de transferencia de correo (SMTP), por el contrario, rige la transferencia de e-mails **salientes** desde el cliente emisor al servidor de e-mail (MDA), como así también el transporte de e-mails entre servidores de e-mail (MTA). SMTP permite transportar e-mails por las redes de datos entre diferentes tipos de software de cliente y servidor, y hace posible el intercambio de e-mails en Internet.

El formato de mensajes del protocolo SMTP utiliza un conjunto rígido de comandos y respuestas. Estos comandos admiten los procedimientos utilizados en el SMTP, como inicio de sesión, transacción de correo, reenvío de correo, verificación de nombres de buzones, expansión de listas de correo y apertura y cierre de intercambios.

Algunos de los comandos especificados en el protocolo SMTP son:

- HELO: identifica el proceso de cliente SMTP para el proceso de servidor SMTP.
- EHLO: es la versión más nueva de HELO, que incluye extensiones de servicios, y
- MAIL FROM: identifica al emisor.
- RCPT TO: identifica al receptor, y
- DATA: identifica el cuerpo del mensaje.



3.3.4 FTP

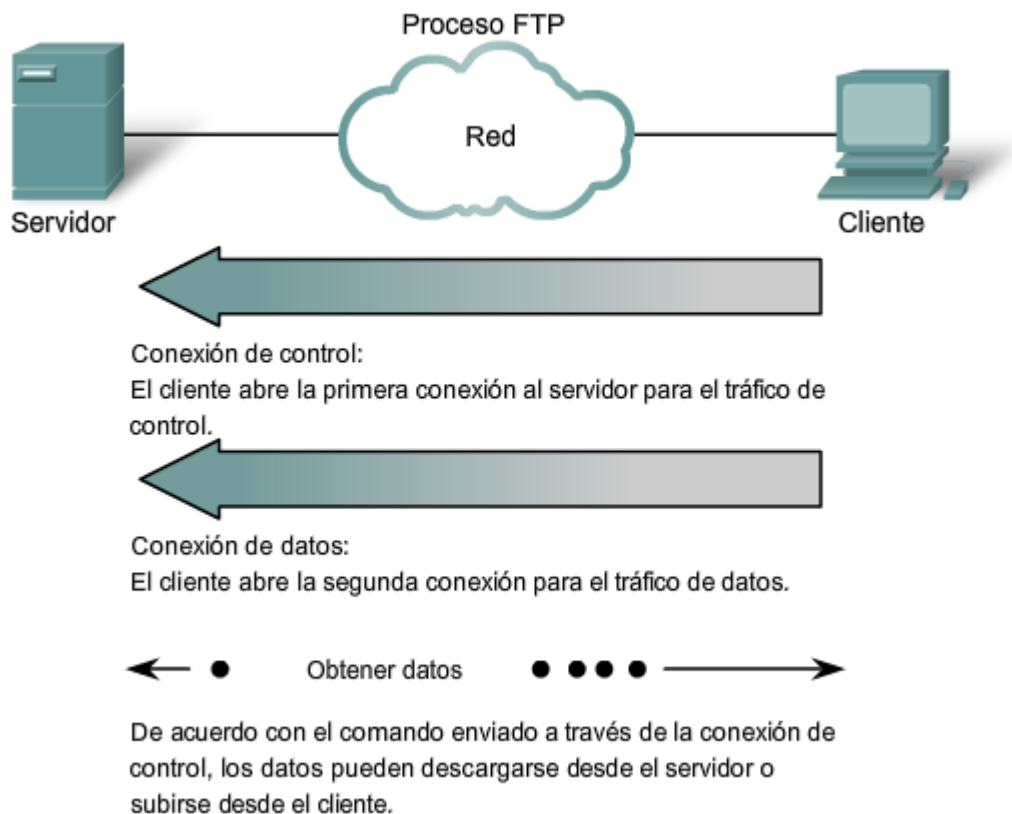
El protocolo de transferencia de archivos (FTP) es otro protocolo de la capa de aplicación comúnmente utilizado. El FTP se desarrolló para permitir las transferencias de archivos entre un cliente y un servidor. Un cliente FTP es una aplicación que se ejecuta en una computadora y se utiliza para cargar y descargar archivos desde un servidor que ejecuta el daemon FTP (FTPD).

Para transferir los archivos en forma exitosa, el FTP requiere de dos conexiones entre cliente y servidor: una para comandos y respuestas, otra para la transferencia real de archivos.

El cliente establece la primera conexión con el servidor en TCP puerto 21. Esta conexión se utiliza para controlar el tráfico, que consiste en comandos del cliente y respuestas del servidor.

El cliente establece la segunda conexión con el servidor en TCP puerto 20. Esta conexión es para la transferencia real de archivos y se crea cada vez que se transfiere un archivo.

La transferencia de archivos puede producirse en ambas direcciones. El cliente puede descargar (bajar) un archivo desde el servidor o el cliente puede cargar (subir) un archivo en el servidor.



3.3.5 DHCP

El servicio [Protocolo de configuración dinámica de host \(DHCP\)](#) permite a los dispositivos de una red obtener direcciones IP y demás información de un servidor DHCP. Este servicio automatiza la asignación de direcciones IP, máscaras de subred, gateways y otros parámetros de redes IP.

DHCP permite a un host obtener una dirección IP en forma dinámica cuando se conecta a la red. Se realiza el contacto con el servidor de DHCP y se solicita una dirección. El servidor DHCP elige una dirección de un rango configurado de direcciones denominado "pool" y se la asigna ("alquila") al host por un período establecido.

En redes locales más grandes o donde cambia frecuentemente la población usuaria, es preferible el DHCP. Los nuevos usuarios llegan con computadoras portátiles y necesitan una conexión. Otros tienen nuevas estaciones de trabajo que necesitan conexión. En lugar de tener direcciones IP asignadas por el administrador de red en cada estación de trabajo, resulta más eficiente tener direcciones IP asignadas en forma automática utilizando un DHCP.

Las direcciones de DHCP distribuidas no se asignan a los hosts en forma permanente, sólo se alquilan durante un período de tiempo. Si el host se apaga o se desconecta de la red, la dirección regresa al pool para volver a utilizarse. Esto es muy útil para los usuarios móviles que entran y salen de la red. Los usuarios pueden moverse libremente desde una ubicación a otra y volver a establecer las conexiones de red. El host puede obtener una dirección IP una vez que se realice la conexión del hardware, ya sea mediante una LAN inalámbrica o conectada por cable.

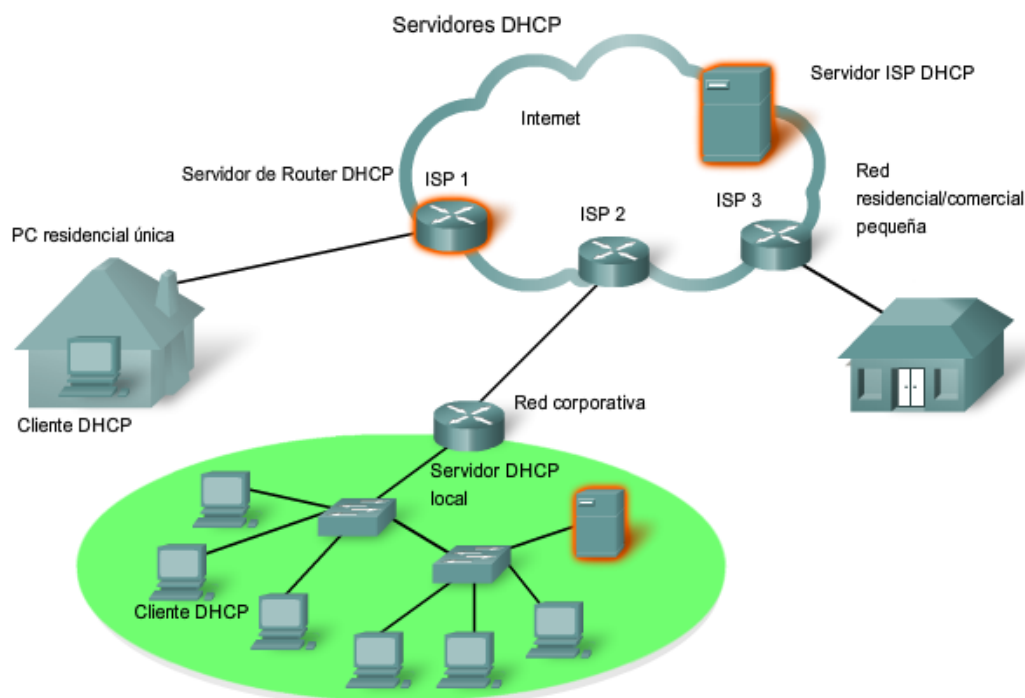
DHCP hace posible el acceso a Internet utilizando zonas activas inalámbricas en aeropuertos o cafés. Una vez que ingresa al área, el cliente de DHCP de la computadora portátil contacta al servidor de DHCP mediante una conexión inalámbrica. El servidor de DHCP asigna una dirección IP a la computadora portátil.

Como muestra la figura, diferentes tipos de dispositivos pueden ser servidores de DHCP al ejecutar el software de servicio de DHCP. El servidor de DHCP en la mayoría de las redes medianas y grandes está generalmente ubicado en un servidor dedicado local basado en PC.

Con las redes domésticas, el servidor de DHCP se ubica en el ISP y un host de la red doméstica recibe la configuración IP directamente desde el ISP.

DHCP puede representar un riesgo a la seguridad porque cualquier dispositivo conectado a la red puede recibir una dirección. Este riesgo hace de la seguridad física un factor importante a la hora de determinar si se utiliza direccionamiento manual o dinámico.

Los direccionamientos dinámico y estático tienen su lugar en los diseños de red. Muchas redes utilizan tanto el direccionamiento estático como el DHCP. DHCP se utiliza para hosts de propósitos generales, como los dispositivos de usuario final, y las direcciones fijas se utilizan para dispositivos de red como gateways, switches, servidores e impresoras.



Sin DHCP los usuarios tienen que ingresar manualmente la dirección IP, la [máscara de subred](#) y otras configuraciones para poder unirse a la red. El servidor de DHCP mantiene un pool de las direcciones IP y alquila una dirección a cualquier cliente habilitado por DHCP cuando el cliente está activado. Debido a que las direcciones IP son dinámicas (alquiladas) en lugar de estáticas (asignadas en forma permanente), las

direcciones en desuso regresan automáticamente al pool para volver a asignarse. Cuando un dispositivo configurado por DHCP se inicia o conecta a la red, el cliente envía un paquete **DESCUBRIMIENTO** de DHCP para identificar cualquier servidor de DHCP disponible en la red. Un servidor DHCP contesta con una oferta de DHCP, que es un mensaje de oferta de alquiler con información asignada de dirección IP, máscara de subred, servidor DNS y [gateway por defecto](#), como también la duración del alquiler.

El cliente puede recibir varios paquetes de oferta de DHCP si hay más de un servidor DHCP en la red local, por lo tanto debe escoger entre ellos y enviar un broadcast de paquete con una solicitud de DHCP que identifique el servidor y la oferta de alquiler específicos que el cliente está aceptando. Un cliente puede elegir solicitar una dirección previamente asignada por el servidor.

Teniendo en cuenta que la dirección IP solicitada por el cliente u ofrecida por el servidor, aún es válida, el servidor devolverá un mensaje [ACK](#) DHCP que le informa al cliente que finalizó el alquiler. Si la oferta ya no es válida, quizás debido al tiempo o o que a otro cliente se le asign el alquiler, el servidor seleccionado responderá con un mensaje NAK DHCP (acuse de recibo negativo). Si se envía un mensaje NAK DHCP, el proceso de selección debe comenzar nuevamente con la transmisión de un nuevo mensaje DHCP DISCOVER.

Una vez que el cliente tenga el alquiler, debe renovarse antes de la expiración del alquiler por medio de otro mensaje DHCP REQUEST.

El servidor de DHCP asegura que todas las direcciones son únicas (una dirección IP no puede asignarse a dos dispositivos de red diferentes en forma simultánea). Usar DHCP permite a los administradores de red volver a configurar fácilmente las direcciones IP del cliente sin tener que realizar cambios a los clientes en forma manual. La mayoría de los proveedores de Internet utilizan DHCP para asignar las direcciones a sus clientes que no solicitan direcciones estáticas.



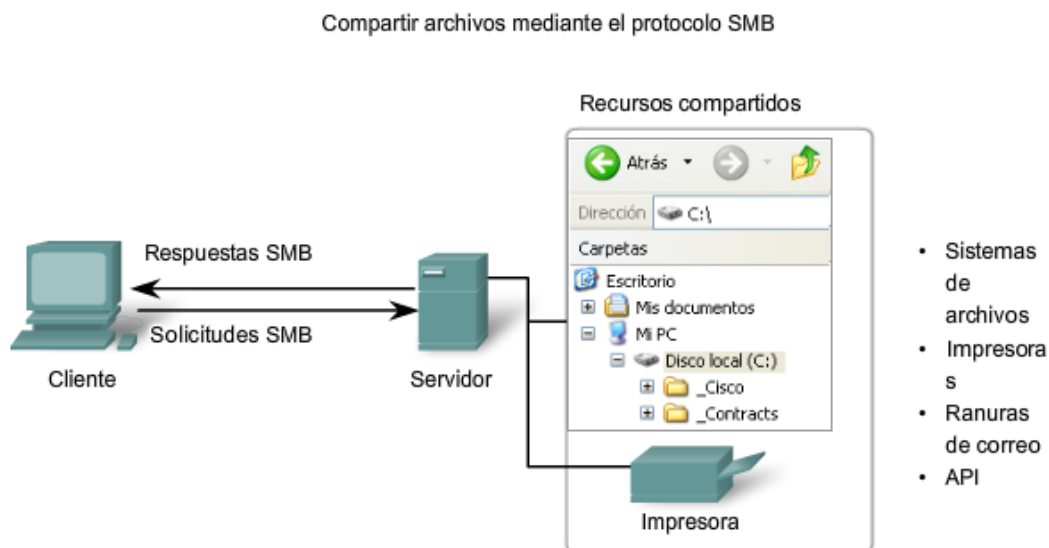
3.3.6 Protocolo SMB y servicios para compartir archivos

El [Bloque de mensajes del servidor \(SMB\)](#) es un protocolo cliente-servidor para compartir archivos. IBM desarrolló el Bloque de mensajes del servidor (SMB) a fines de la década del '80 para describir la estructura de recursos de red compartidos, como directorios, archivos, impresoras y puertos seriales. Es un protocolo de solicitud-

respuesta. A diferencia del protocolo para compartir archivos respaldado por FTP, los clientes establecen una conexión a largo plazo con los servidores. Una vez establecida la conexión, el usuario del cliente puede acceder a los recursos en el servidor como si el recurso fuera local para el host del cliente.

Los servicios de impresión y el SMB para compartir archivos se han transformado en el pilar de las redes de Microsoft. Con la presentación de la serie Windows 2000 del software, Microsoft cambió la estructura subyacente para el uso del SMB. En versiones anteriores de los productos de Microsoft, los servicios de SMB utilizaron un protocolo que no es TCP/IP para implementar la resolución de nombres. Comenzando con Windows 2000, todos los productos subsiguientes de Microsoft utilizan denominación DNS. Esto permite a los protocolos TCP/IP admitir directamente el compartir recursos SMB, como se muestra en la figura.

Los sistemas operativos LINUX y [UNIX](#) también proporcionan un método para compartir recursos con las redes Microsoft a través de una versión de SMB denominada SAMBA. Los sistemas operativos Macintosh de Apple también admiten recursos compartidos utilizando el protocolo SMB.



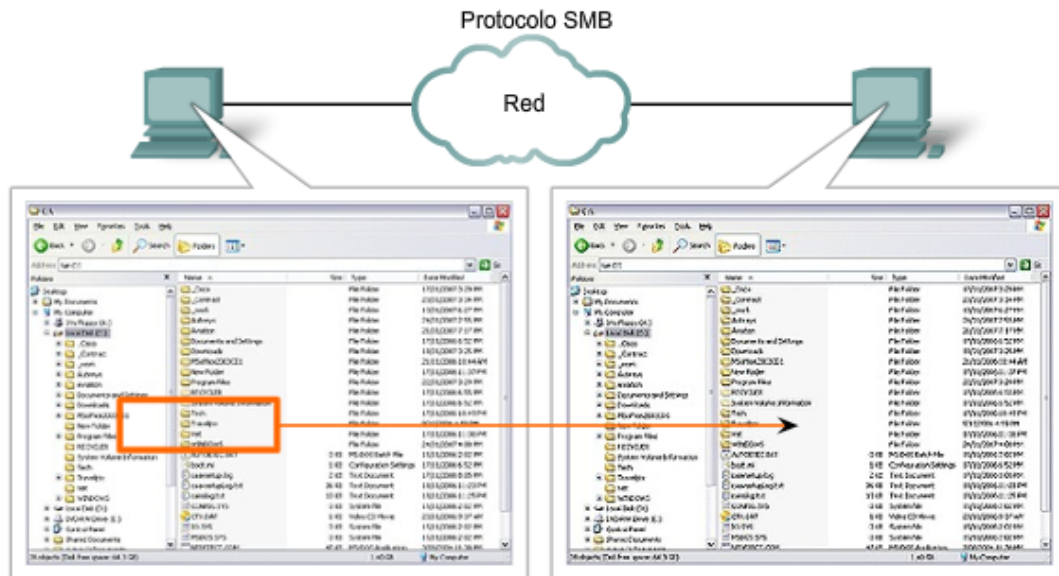
SMB es un protocolo de solicitud-respuesta y cliente-servidor. Los servidores pueden poner sus recursos a disposición de los clientes en la red.

El protocolo SMB describe el acceso al sistema de archivos y la manera en que los clientes hacen solicitudes de archivos. Además describe la comunicación entre procesos del protocolo SMB. Todos los mensajes SMB comparten un mismo formato. Este formato utiliza un encabezado de tamaño fijo seguido por un parámetro de tamaño variable y un componente de datos.

Los mensajes SMB pueden:

- Iniciar, autenticar y terminar sesiones
- Controlar el acceso a archivos e impresoras
- Permitir a una aplicación enviar o recibir mensajes hacia o desde otro dispositivo

El proceso de intercambio de archivos SMB se muestra en la figura.



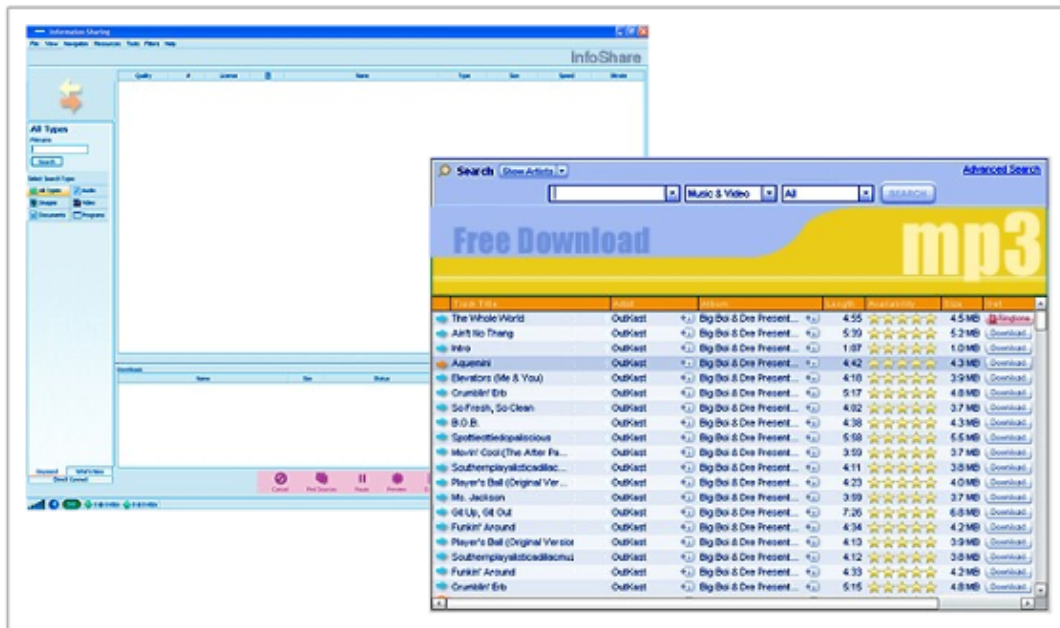
Puede copiarse un archivo desde una PC a otra con Windows Explorer mediante el protocolo SMB.

3.3.7 Protocolo Gnutella y servicios P2P

Aprendimos acerca de FTP y SMB como formas de obtener archivos; aquí presentamos otro protocolo de aplicación. Compartir archivos en Internet se ha transformado en algo muy popular. Con las **aplicaciones P2P** basadas en el protocolo Gnutella, las personas pueden colocar archivos en sus discos rígidos para que otros los descarguen. El software del cliente compatible con Gnutella permite a los usuarios conectarse con los servicios Gnutella en Internet, ubicarlos y acceder a los recursos compartidos por otros pares Gnutella.

Muchas aplicaciones del cliente están disponibles para acceder en la red Gnutella, entre ellas: BearShare, Gnucleus, LimeWire, Morpheus, WinMX y XoloX (consulte una captura de pantalla de LimeWire en la figura). Mientras que el Foro de desarrolladores de Gnutella mantiene el protocolo básico, los proveedores de las aplicaciones generalmente desarrollan extensiones para lograr que el protocolo funcione mejor en las aplicaciones.

Aplicaciones punto a punto

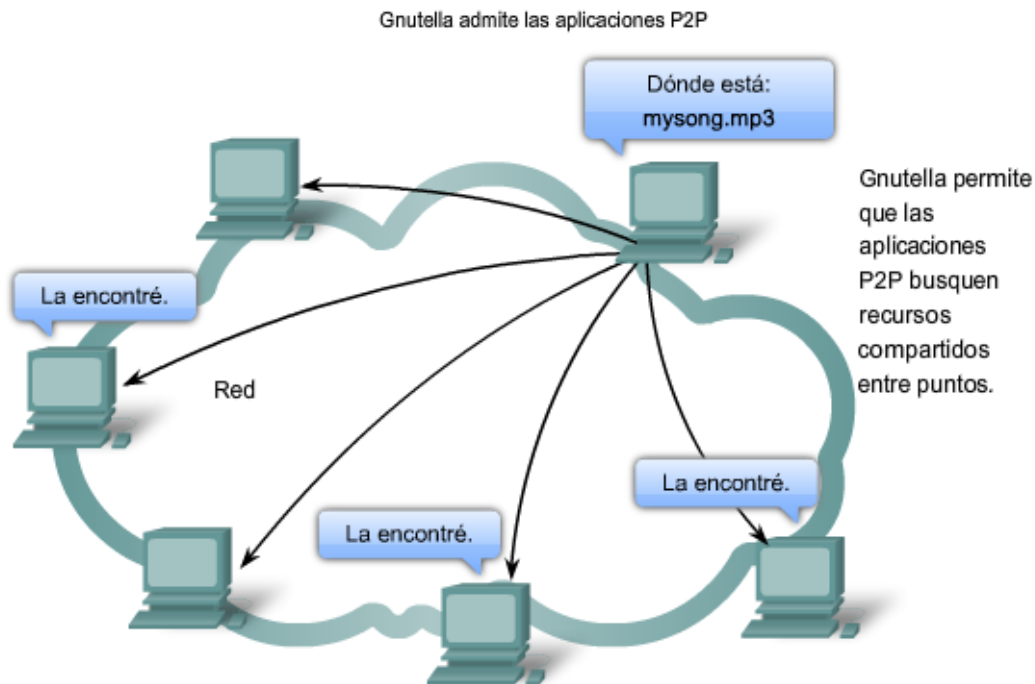


Muchas de las aplicaciones P2P no utilizan una base de datos central para registrar todos los archivos disponibles en los puntos. Por el contrario, los dispositivos en la red se indican entre ellos qué archivos están disponibles cuando hay una consulta, y utilizan el protocolo Gnutella y los servicios para respaldar los recursos ubicados.

Cuando un usuario se conecta a un servicio Gnutella, las aplicaciones del cliente buscarán otros nodos Gnutella para conectarse. Estos nodos manejan las consultas para las ubicaciones de los recursos y responden a dichas solicitudes. Además, gobiernan los mensajes de control que ayudan al servicio a descubrir otros nodos. Las verdaderas transferencias de archivos generalmente dependen de los servicios HTTP.

El protocolo Gnutella define cinco tipos de paquetes diferentes:

- ping: para descubrir un dispositivo,
- pong: como respuesta a un ping,
- consulta: para ubicar un archivo,
- query hit: como respuesta a una consulta, y
- push: como una solicitud de descarga.



3.3.8 Protocolo y servicios Telnet

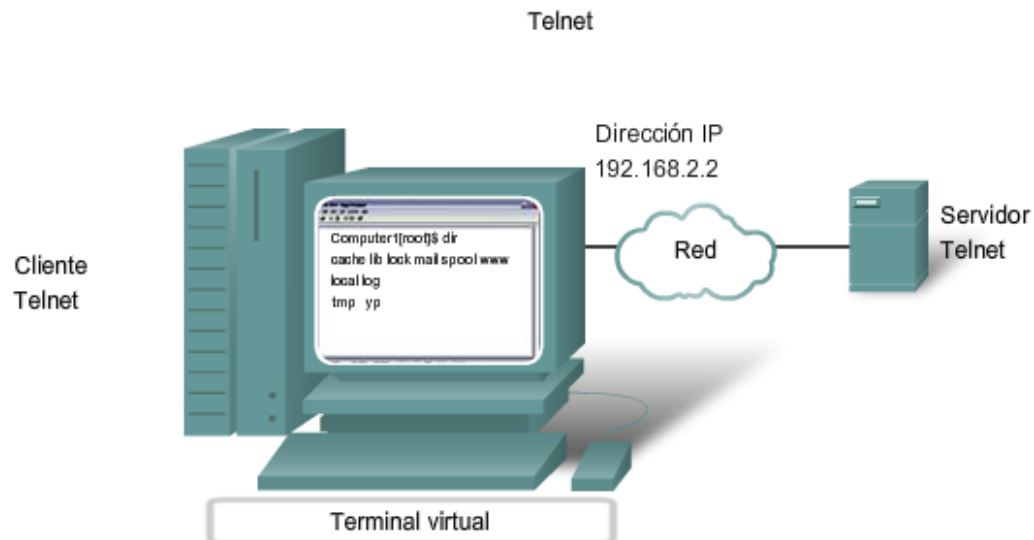
Mucho antes de que existieran las computadoras de escritorio con interfaces gráficas sofisticadas, las personas utilizaban sistemas basados en textos que eran simplemente terminales conectadas físicamente a una computadora central. Una vez que las redes estuvieran disponibles, las personas necesitaban acceder en forma remota a los sistemas informáticos de la misma manera en que lo hacían con las terminales conectadas en forma directa.

Telnet se desarrolló para satisfacer esta necesidad. Telnet se remonta a principios de la década de los setenta y se encuentra entre los servicios y protocolos de capa de aplicación más antiguo dentro del grupo TCP/IP. Telnet proporciona un método estándar de emulación de dispositivos de terminal basados en texto en la red de datos. El protocolo y el software del cliente que implementa el protocolo comúnmente se definen como Telnet.

Y como consecuencia, una conexión que utiliza Telnet se llama Sesión o conexión de terminal virtual (VTY). En lugar de utilizar un dispositivo físico para conectar al servidor, Telnet utiliza software para crear un dispositivo virtual que proporciona las mismas funciones que una sesión terminal con acceso a la Interfaz de línea de comandos (CLI) del servidor.

Para admitir conexiones al cliente Telnet, el servidor ejecuta un servicio llamado daemon de Telnet. Se establece una conexión de terminal virtual desde un dispositivo final utilizando una aplicación del cliente Telnet. La mayoría de los sistemas operativos incluye un cliente de Telnet de la capa de aplicación. En una PC de Microsoft Windows, Telnet puede ejecutarse desde la entrada del comando. Otras aplicaciones de terminal comunes que ejecutan clientes de Telnet son HyperTerminal, Minicom y TeraTerm.

Una vez establecida una conexión Telnet, los usuarios pueden realizar cualquier función autorizada en el servidor, como si utilizaran una sesión de línea de comandos en el servidor mismo. Si están autorizados, pueden iniciar y detener procesos, configurar el dispositivo e inclusive cerrar el sistema.



Telnet proporciona una forma de utilizar una computadora, conectada a través de la red, para acceder a un dispositivo de red como si el teclado y el monitor estuvieran conectados directamente al dispositivo.

Telnet es un protocolo cliente-servidor y especifica cómo se establece y se termina una sesión VTY. Además proporciona la sintaxis y el orden de los comandos utilizados para iniciar la sesión Telnet, como así también los comandos de control que pueden ejecutarse durante una sesión. Cada comando Telnet consiste en por lo menos dos bytes. El primer byte es un carácter especial denominado **Interpretar como comando** (IAC). Como su nombre lo indica, el IAC define el byte siguiente como un comando en lugar de un texto.

Algunos de los comandos del protocolo Telnet de muestra son:

Are You There (AYT): Permite al usuario solicitar que aparezca algo en la pantalla del terminal para indicar que la sesión VTY está activa.

Erase Line (EL): Elimina todo el texto de la línea actual.

Interrupt Process (IP): Suspende, interrumpe, aborta o termina el proceso al cual se conectó la terminal virtual. Por ejemplo, si un usuario inició un programa en el servidor Telnet por medio de VTY, puede enviar un comando IP para detener el programa.

Aunque el protocolo Telnet admite autenticación de usuario, no admite el transporte de datos encriptados. Todos los datos intercambiados durante una sesión Telnet se transportan como texto sin formato por la red. Esto significa que los datos pueden ser interceptados y entendidos fácilmente.

Si la seguridad es un problema, el protocolo Shell seguro (SSH) ofrece un método seguro y alternativo para acceder al servidor. SSH proporciona la estructura para un inicio de sesión remoto seguro y otros servicios de red seguros. Además proporciona mayor autenticación que Telnet y admite el transporte de datos de sesión utilizando cifrado. **Como una mejor práctica, los profesionales de red deberían siempre utilizar SSH en lugar de Telnet, cada vez que sea posible.**

Más adelante en este curso, utilizaremos Telnet y SSH para acceder y configurar los dispositivos de red en la red de laboratorios.

Telnet: Aplicación, servicio y protocolo

