Vivimos la experiencia de Internet a través de la World Wide Web cuando transmitimos videos, jugamos juegos en línea, chateamos con amigos y les enviamos correos electrónicos, y buscamos ofertas en sitios Web. Las aplicaciones, como las que se utilizan para proporcionar los servicios mencionados, brindan la interfaz humana a la red subyacente. Estas aplicaciones nos permiten enviar y recibir datos de forma relativamente fácil. En general, podemos acceder a estas aplicaciones y utilizarlas sin saber cómo funcionan. Sin embargo, para los profesionales de la red, es importante saber cómo una aplicación puede formatear, transmitir e interpretar mensajes que se envían y se reciben a través de la red.

La visualización de los mecanismos que permiten la comunicación a través de la red se hace más fácil si utilizamos el esquema en capas del modelo OSI.

En este capítulo, analizaremos la función de la capa de aplicación y la manera en que las aplicaciones, los servicios y los protocolos que están dentro de la capa de aplicación hacen posible una comunicación sólida a través de las redes de datos.

Como se muestra en la ilustración, los profesionales de redes utilizan los modelos OSI y TCP/IP para comunicarse tanto verbalmente como mediante documentación técnica escrita. Como tales, los profesionales de redes pueden utilizar estos modelos para describir el comportamiento de protocolos y aplicaciones.

En el modelo OSI, la información pasa de una capa a otra: de la capa de aplicación en el host de transmisión pasa por la jerarquía hacia la capa física y luego por el canal de comunicaciones hacia el host de destino, donde la información vuelve a la jerarquía y termina en la capa de aplicación.

La capa de aplicación es la capa superior de los modelos OSI y TCP/IP. La capa de aplicación de TCP/IP incluye un número de protocolos que proporciona funcionalidad específica a una variedad de aplicaciones de usuario final. La funcionalidad de los protocolos de capa de aplicación de TCP/IP se adapta aproximadamente al esquema de las tres capas superiores del modelo OSI: la de aplicación, la de presentación y la de sesión. Las capas 5, 6 y 7 del modelo OSI se utilizan como referencias para proveedores y desarrolladores de software de aplicación para fabricar productos, como exploradores Web, que necesitan acceder a las redes.

**Capa de aplicación**

La capa de aplicación es la más cercana al usuario final. Como se muestra en la ilustración, 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 están en constante desarrollo. Algunos de los protocolos de capa de aplicación más conocidos incluyen el protocolo de transferencia de hipertexto (HTTP), el protocolo de transferencia de archivos (FTP), el protocolo trivial de transferencia de archivos (TFTP), el protocolo de acceso a mensajes de Internet (IMAP) y el protocolo del Sistema de nombres de dominios (DNS).

**Capa de presentación**

La capa de presentación tiene tres funciones principales:

* Dar formato a los datos del dispositivo de origen, o presentarlos, en una forma compatible para que lo reciba el dispositivo de destino.
* Comprimir los datos de forma tal que los pueda descomprimir el dispositivo de destino.
* Encriptar los datos para su transmisión y posterior descifrado al llegar al dispositivo de destino.

Como se muestra en la ilustración, la capa de presentación da formato a los datos para la capa de aplicación y establece estándares para los formatos de archivo. Dentro de los estándares más conocidos para video encontramos QuickTime y el Grupo de expertos en películas (MPEG). QuickTime es una especificación de PC de Apple para audio y video, y MPEG es un estándar para la codificación y compresión de audio y video.

Entre los formatos gráficos de imagen conocidos que se utilizan en redes, se incluyen los siguientes: formato de intercambio de gráficos (GIF), formato del Joint Photographic Experts Group (JPEG) y formato de gráficos de red portátiles (PNG). Los formatos GIF y JPEG son estándares de compresión y codificación de imágenes gráficas. El formato PNG se diseñó para abordar algunas de las limitaciones del formato GIF y para reemplazar este último.

**Capa de sesión**

Como su nombre lo indica, las funciones de la capa de sesión 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 que estuvieron inactivas durante un período prolongado.

Si bien el modelo OSI separa las funciones individuales de las capas de aplicación, presentación y sesión, las aplicaciones de TCP/IP más conocidas e implementadas incorporan la funcionalidad de las tres capas.

Los protocolos de aplicación de TCP/IP especifican el formato y la información de control necesarios para muchas funciones de comunicación comunes de Internet. Algunos de los protocolos TCP/IP son:

* **Sistema de nombres de dominios (DNS):** este protocolo resuelve nombres de Internet en direcciones IP.
* **Telnet:** se utiliza para proporcionar acceso remoto a servidores y dispositivos de red.
* **Protocolo simple de transferencia de correo (SMTP):** este protocolo transfiere mensajes y archivos adjuntos de correo electrónico.
* **Protocolo de configuración dinámica de host (DHCP):** se utiliza para asignar una dirección IP y direcciones de máscara de subred, de gateway predeterminado y de servidor DNS a un host.
* **Protocolo de transferencia de hipertexto (HTTP):** este protocolo transfiere archivos que conforman las páginas Web de la World Wide Web.
* **Protocolo de transferencia de archivos (FTP):** se utiliza para la transferencia de archivos interactiva entre sistemas.
* **Protocolo trivial de transferencia de archivos (TFTP):** se utiliza para la transferencia de archivos activa sin conexión.
* **Protocolo bootstrap (BOOTP):** este protocolo es un precursor del protocolo DHCP. BOOTP es un protocolo de red que se utiliza para obtener información de la dirección IP durante el arranque.
* **Protocolo de oficina de correos (POP):** es un protocolo que utilizan los clientes de correo electrónico para recuperar el correo electrónico de un servidor remoto.
* **Protocolo de acceso a mensajes de Internet (IMAP):** este es otro protocolo que se utiliza para recuperar correo electrónico.

Los protocolos de capa de aplicación son utilizados tanto por los dispositivos de origen como de destino durante una sesión de comunicación. Para que las comunicaciones se lleven a cabo correctamente, los protocolos de capa de aplicación que se implementaron en los hosts de origen y de destino deben ser compatibles.

Cuando se accede a la información en un dispositivo de red, ya sea una PC, una computadora portátil, una tablet PC, un smartphone o algún otro dispositivo conectado a una red, los datos no se pueden almacenar físicamente en el dispositivo. En este caso, se debe solicitar permiso al dispositivo que contiene los datos para acceder a esa información. En el modelo de red punto a punto (P2P), se accede a los datos de un dispositivo punto sin utilizar un servidor dedicado.

El modelo de red P2P consta de dos partes: las redes P2P y las aplicaciones P2P. Ambas partes tienen características similares, pero en la práctica son muy diferentes.

**P2P Networks**

En una red P2P, hay dos o más PC que están conectadas por medio de una red y pueden compartir recursos (como impresoras y archivos) sin tener un servidor dedicado. Todo dispositivo final conectado (conocido como “punto”) puede funcionar como servidor y como cliente. Una computadora puede asumir la función de servidor para una transacción mientras funciona en forma simultánea como cliente para otra transacción. Las funciones de cliente y servidor se establecen por solicitud.

Un ejemplo de esto es una red doméstica simple con dos PC, como se muestra en la ilustración. En este ejemplo, el Punto2 tiene una impresora conectada a él directamente por USB y está configurado para compartir la impresora en la red de modo que el Punto1 pueda imprimir con esta. El Punto1 está configurado para compartir una unidad o una carpeta en la red. Esto permite que el Punto2 acceda a los archivos de la carpeta compartida y los guarde. Además de compartir archivos, una red como esta permitiría que los usuarios habiliten juegos en red o compartan una conexión a Internet.

Las redes P2P descentralizan los recursos en una red. En lugar de ubicar datos para compartir en los servidores dedicados, los datos se pueden colocar en cualquier parte y en cualquier dispositivo conectado. La mayoría de los sistemas operativos actuales admiten compartir archivos e impresoras sin requerir software del servidor adicional. Sin embargo, las redes P2P no utilizan cuentas de usuario centralizadas ni acceden a servidores para mantener permisos. Por lo tanto, es difícil aplicar políticas de seguridad y de acceso en redes que contienen varias PC. Se deben establecer cuentas de usuario y derechos de acceso en forma individual para cada dispositivo.

Una aplicación punto a punto (P2P) permite que un dispositivo funcione como cliente y como servidor dentro de la misma comunicación, como se muestra en la ilustración. 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 P2P requieren que cada dispositivo final proporcione una interfaz de usuario y ejecute un servicio en segundo plano. Cuando inicia una aplicación P2P específica, se cargan los servicios en segundo plano y la interfaz de usuario requeridos; a continuación, los dispositivos se pueden comunicar directamente.

Algunas aplicaciones P2P utilizan un sistema híbrido donde se descentraliza el intercambio de 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 con el servidor de índice.

Las aplicaciones P2P se pueden utilizar en redes P2P, en redes cliente/servidor y a través de Internet.

Con las aplicaciones P2P, cada PC de la red que ejecuta la aplicación puede funcionar como cliente o como servidor para las otras PC en la red que ejecutan la aplicación. Las aplicaciones P2P comunes incluyen las siguientes:

* eDonkey
* eMule
* Shareaza
* BitTorrent
* Bitcoin
* LionShare

Algunas aplicaciones P2P se basan en el protocolo Gnutella. Estas aplicaciones permiten compartir archivos en discos duros con otras personas. Como se muestra en la ilustración, el software de cliente compatible con Gnutella permite a los usuarios conectarse a los servicios Gnutella a través de Internet, además de ubicar los recursos compartidos por otros puntos Gnutella y acceder a dichos recursos. Hay muchas aplicaciones cliente disponibles para acceder a la red Gnutella tales como BearShare, Gnucleus, LimeWire, Morpheus, WinMX y XoloX.

Mientras que el foro de desarrolladores de Gnutella mantiene el protocolo básico, los proveedores de aplicaciones suelen desarrollar extensiones para lograr que el protocolo funcione mejor con dichas aplicaciones.

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 mutuamente qué archivos están disponibles cuando hay una consulta, y utilizan el protocolo y los servicios de intercambio de archivos para dar soporte a la búsqueda de recursos.

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 parte de la capa de aplicación. El cliente comienza el intercambio solicitando los datos al servidor, quien responde enviando uno o más streams de datos al cliente. Los protocolos de la capa de aplicación describen el formato de las solicitudes y respuestas entre clientes y servidores. Además de la transferencia real de datos, este intercambio también puede requerir la autenticación del usuario y la identificación de un archivo de datos que se vaya a transferir.

Un ejemplo de una red cliente-servidor es el uso del servicio de correo electrónico de un ISP para enviar, recibir y almacenar correo electrónico. El cliente de correo electrónico en una PC doméstica emite una solicitud al servidor de correo electrónico del ISP para que se le envíe todo correo no leído. El servidor responde enviando al cliente el correo electrónico solicitado.

Aunque los datos se describen generalmente como el flujo del servidor al cliente, algunos datos fluyen siempre del cliente al servidor. El flujo de datos puede ser el mismo en ambas direcciones, o inclusive puede 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. Como se muestra en la ilustración, la transferencia de datos de un cliente a un servidor se conoce como “subida” y la transferencia de datos de un servidor a un cliente se conoce como “descarga”.

Existen muchos protocolos de capa de aplicación, pero en un día típico probablemente utiliza solo cinco o seis. Los siguientes son tres protocolos de capa de aplicación que forman parte del trabajo o los juegos cotidianos:

* Protocolo de transferencia de hipertexto (HTTP)
* Protocolo simple de transferencia de correo (SMTP)
* Protocolo de oficina de correos (POP)

Estos protocolos de capa de aplicación permiten explorar la Web y enviar y recibir correo electrónico. HTTP se utiliza para que los usuarios puedan conectarse a sitios Web a través de Internet. SMTP permite que los usuarios puedan enviar correo electrónico. POP permite que los usuarios puedan recibir correo electrónico.

En las próximas páginas, se hará hincapié en estos tres protocolos de capa de aplicación.

Cuando se escribe una dirección Web o un localizador uniforme de recursos (URL) en un explorador Web, el explorador establece una conexión con el servicio Web que se ejecuta en el servidor mediante el protocolo HTTP. Los nombres que la mayoría de las personas asocia con las direcciones Web son URL e identificador uniforme de recursos (URI).

El URL <http://www.cisco.com/index.html>   es un ejemplo de un URL que se refiere a un recurso específico: una página Web llamada **index.html** en un servidor identificado como **cisco.com**. Haga clic en cada ilustración para ver los pasos que utiliza HTTP.

Los exploradores Web son el tipo de aplicación cliente que utiliza una PC para conectarse a la World Wide Web y acceder a 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 establecen conexiones al servidor y solicitan los recursos deseados. El servidor responde con el recurso y, al recibirlo, el explorador interpreta los datos y los presenta al usuario.

Los exploradores pueden interpretar y presentar muchos tipos de datos (como texto no cifrado o lenguaje de marcado de hipertexto, que es el lenguaje que se utiliza para construir páginas Web). Otros tipos de datos, sin embargo, requieren de otro servicio o programa. Generalmente se les 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, utilice el URL <http://www.cisco.com/index.html>  .

Primero, el explorador interpreta las tres partes del URL, como se muestra en la figura 1:

1. **http** (el protocolo o esquema)

2. **www.cisco.com** (el nombre del servidor)

3. **index.html** (el nombre de archivo específico solicitado)

A continuación, el explorador verifica con un servidor de nombre para convertir a www.cisco.com en una dirección numérica que utiliza para conectarse al servidor, como se muestra en la figura 2. Mediante los requisitos de HTTP, el explorador envía una solicitud GET al servidor y solicita el archivo **index.html**. El servidor envía el código HTML para esta página Web al explorador, como se muestra en la figura 3. Finalmente, el explorador descifra el código HTML y da formato a la página para que se pueda visualizar en la ventana del explorador, como se muestra en la figura 4.

HTTP se utiliza a través de la World Wide Web para transferencia de datos y es uno de los protocolos de aplicación más utilizados hoy en día. Originalmente, este protocolo se desarrolló solo para publicar y recuperar páginas HTML. Sin embargo, la flexibilidad de HTTP lo convirtió en una aplicación fundamental de los sistemas de información distribuidos y cooperativos.

HTTP es un protocolo de solicitud/respuesta. Cuando un cliente, por lo general un explorador Web, envía una solicitud a un servidor Web, HTTP especifica los tipos de mensaje que se utilizan para esa comunicación. Los tres tipos de mensajes comunes son GET, POST y PUT (consulte la ilustración).

GET es una solicitud de datos por parte del cliente. Un cliente (explorador Web) envía el mensaje GET al servidor Web para solicitar las páginas HTML. Cuando el servidor recibe la solicitud GET, este responde con una línea de estado, como HTTP/1.1 200 OK, y un mensaje propio. El mensaje del servidor puede incluir el archivo HTML solicitado, si está disponible, o puede contener un mensaje de error o de información, como “Se modificó la ubicación del archivo solicitado”.

Los mensajes POST y PUT se utilizan para subir datos al servidor Web. Por ejemplo, cuando el usuario introduce datos en un formulario que está integrado en una página Web (p. ej., cuando se completa una solicitud de pedido), el mensaje POST se envía al servidor Web. En el mensaje POST, se incluyen los datos que el usuario introdujo en el formulario.

PUT carga los recursos o el contenido en el servidor Web. Por ejemplo, si un usuario intenta subir un archivo o una imagen a un sitio Web, el cliente envía un mensaje PUT al servidor con la imagen o el archivo adjunto.

Aunque HTTP es sumamente flexible, no es un protocolo seguro. Los mensajes de solicitud envían 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 se descifran.

Para una comunicación segura a través de Internet, se utiliza el protocolo HTTP seguro (HTTPS) para acceder o subir información al servidor Web. El HTTPS puede utilizar autenticación y encriptación para asegurar los datos mientras viajan entre el cliente y el servidor. HTTPS especifica reglas adicionales para pasar datos entre la capa de aplicación y la capa de transporte. El protocolo HTTPS utiliza el mismo proceso de solicitud del cliente-respuesta del servidor que HTTP, pero el stream de datos se encripta con capa de sockets seguros (SSL) antes de transportarse a través de la red. El HTTPS crea una carga y un tiempo de procesamiento adicionales en el servidor debido a la encriptación y el descifrado de tráfico.

Uno de los principales servicios que un ISP ofrece es hosting de correo electrónico. El correo electrónico revolucionó la forma en que las personas se comunican gracias a su sencillez y velocidad. No obstante, para ejecutar el correo electrónico en una PC o en otro dispositivo final, este requiere varios servicios y aplicaciones.

El correo electrónico es un método para almacenar y enviar que se utiliza para enviar, almacenar y recuperar mensajes electrónicos a través de una red. Los mensajes de correo electrónico se guardan en bases de datos en servidores de correo. A menudo, los ISP mantienen servidores de correo que admiten varias cuentas de clientes diferentes.

Los clientes de correo electrónico se comunican con servidores de correo para enviar y recibir mensajes de correo electrónico. Los servidores de correo se comunican con otros servidores de correo para transportar mensajes desde un dominio a otro. Un cliente de correo electrónico no se comunica directamente con otro cliente de correo electrónico cuando envía un mensaje. Más bien, ambos clientes dependen del servidor de correo para el transporte de los mensajes. Esto sucede incluso cuando ambos usuarios se encuentran en el mismo dominio.

Los clientes de correo electrónico envían mensajes al servidor de correo electrónico determinado en las configuraciones de aplicaciones. Cuando el servidor recibe el mensaje, verifica si el dominio receptor se encuentra en su base de datos local. De no ser así, envía una solicitud de DNS para determinar la dirección IP del servidor de correo electrónico para el dominio de destino. A continuación, el correo electrónico se reenvía al servidor correspondiente.

El correo electrónico admite tres protocolos diferentes para su funcionamiento: el protocolo simple de transferencia de correo (SMTP), el protocolo de oficina de correos (POP) y el protocolo de acceso a mensajes de Internet (IMAP). El proceso de capa de aplicación que envía correo utiliza SMTP. Esto sucede cuando se envía correo de un cliente a un servidor y cuando se envía correo de un servidor a otro.

Sin embargo, un cliente recupera el correo electrónico mediante uno de dos protocolos de capa de aplicación: POP o IMAP.

El protocolo simple de transferencia de correo (SMTP) transfiere correo electrónico con confianza y eficacia. Para que las aplicaciones del SMTP funcionen bien, se debe formatear correctamente el mensaje de correo electrónico y los procesos SMTP deben estar en ejecución en el cliente y en el servidor.

Los formatos de mensajes SMTP necesitan un encabezado y un cuerpo de mensaje. Mientras que el cuerpo del mensaje puede contener la cantidad de texto que se desee, el encabezado debe contar con una dirección de correo electrónico de destinatario correctamente formateada y una dirección de emisor. Toda otra información de encabezado es opcional.

Cuando un cliente envía correo electrónico, el proceso SMTP del cliente se conecta a un proceso SMTP del servidor en el puerto bien conocido 25. Después de que se establece la conexión, el cliente intenta enviar el correo electrónico al servidor a través de esta. Una vez que el servidor recibe el mensaje, lo ubica en una cuenta local (si el destinatario es local) o lo reenvía mediante el mismo proceso de conexión SMTP a otro servidor de correo para su entrega.

El servidor de correo electrónico de destino puede no estar en línea, o muy ocupado, cuando se envían los mensajes. Por lo tanto, el SMTP pone los mensajes en cola para enviarlos posteriormente. El servidor verifica periódicamente la cola en busca de mensajes e intenta enviarlos nuevamente. Si el mensaje aún no se ha entregado después de un tiempo predeterminado de expiración, se devolverá al emisor como imposible de entregar.

El protocolo de oficina de correos (POP) permite que una estación de trabajo pueda recuperar correos de un servidor de correo. Con POP, el correo se descarga desde el servidor al cliente y después se elimina en el servidor.

El servidor comienza el servicio POP escuchando de manera pasiva en el puerto TCP 110 las solicitudes de conexión del cliente. Cuando un cliente desea utilizar el servicio, envía una solicitud para establecer una conexión TCP con el servidor. Una vez establecida la conexión, el servidor POP envía un saludo. A continuación, el cliente y el servidor POP intercambian comandos y respuestas hasta que la conexión se cierra o cancela.

Dado que estos mensajes de correo electrónico se descargan para el cliente y se eliminan del servidor, esto significa que no existe una ubicación centralizada donde se conserven los mensajes de correo electrónico. Como el POP no almacena mensajes, no es una opción adecuada para una pequeña empresa que necesita una solución de respaldo centralizada.

El POP3 es deseable para los ISP, ya que aligera su responsabilidad de manejar grandes cantidades de almacenamiento para sus servidores de correo electrónico.

El Protocolo de acceso a mensajes de Internet (IMAP, Internet Message Access Protocol) es otro protocolo que describe un método para recuperar mensajes de correo electrónico. Sin embargo, a diferencia del POP, cuando el usuario se conecta a un servidor para IMAP, se descargan copias de los mensajes a la aplicación del cliente. Los mensajes originales se mantienen en el servidor hasta que se eliminen manualmente. Los usuarios ven copias de los mensajes en su software de cliente de correo electrónico.

Los usuarios pueden crear una jerarquía de archivos en el servidor para organizar y guardar el correo. Dicha estructura de archivos se duplica también en el cliente de correo electrónico. Cuando un usuario decide eliminar un mensaje, el servidor sincroniza esa acción y elimina el mensaje del servidor.

Para pequeñas o medianas empresas, son muchas las ventajas al utilizar el protocolo IMAP. El IMAP puede realizar un almacenamiento a largo plazo de mensajes de correo electrónico en servidores de correo y permitir el respaldo centralizado. También les permite a los empleados acceder a mensajes de correo electrónico desde distintas ubicaciones, utilizando dispositivos o software de cliente diferentes. La estructura de carpetas del buzón que un usuario espera ver se encuentra disponible para visualizarla, independientemente del modo en que el usuario obtenga acceso al buzón.

Para un ISP, el IMAP puede no ser el protocolo elegido. El espacio de disco para admitir la gran cantidad de mensajes de correo electrónico almacenados puede ser costoso de comprar y mantener. Además, si los clientes esperan que se realicen copias de respaldo a sus buzones periódicamente, esto puede aumentar aún más los costos para el ISP.

En las redes de datos, los dispositivos se etiquetan con direcciones IP numéricas para enviar y recibir datos a través de las redes. La mayoría de las personas no puede recordar estas direcciones numéricas. Los nombres de dominio se crearon para convertir las direcciones numéricas en un nombre sencillo y reconocible.

En Internet, estos nombres de dominio, como [http://www.cisco.com](http://www.cisco.com/)  , son mucho más fáciles de recordar que algo como 198.133.219.25, que es la dirección numérica real de ese servidor. Si Cisco decide cambiar la dirección numérica [www.cisco.com](http://www.cisco.com/)  , es claro para el usuario, porque el nombre de dominio se mantiene. Simplemente se une la nueva dirección al nombre de dominio existente y se mantiene la conectividad. Cuando las redes eran pequeñas, resultaba fácil mantener la asignación entre los nombres de dominios y las direcciones que representaban. A medida que el tamaño de las redes y la cantidad de dispositivos aumentaron, este sistema manual se volvió inviable.

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. Haga clic en los botones de la ilustración para conocer los pasos para resolver direcciones de DNS.

El protocolo DNS define un servicio automatizado que coincide con nombres de recursos que tienen la dirección de red numérica solicitada. Incluye el formato de consultas, respuestas y datos. Las comunicaciones del protocolo DNS utilizan un único formato 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.

En las figuras 1 a 5, se muestran los pasos relacionados con la resolución DNS.

Un servidor DNS proporciona la resolución de nombres mediante *Berkeley Internet Domain Name* (BIND), o el demonio de nombres, que a menudo se denomina “named” (pronunciado “neimdi”). BIND fue desarrollado originalmente por cuatro estudiantes de la Universidad de California en Berkeley a principios de la década de los ochenta. Como se muestra en la ilustración, el formato del mensaje DNS que utiliza BIND es el formato DNS más utilizado en Internet.

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 registros son:

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

Cuando un cliente realiza una consulta, el proceso BIND del servidor observa primero sus propios registros para resolver el nombre. Si no puede resolverlo con los registros almacenados, contacta a otros servidores para hacerlo.

La solicitud puede pasar a lo largo de cierta cantidad de servidores, lo cual puede tomar más tiempo y consumir banda ancha. Una vez que se encuentra una coincidencia y se la devuelve al servidor solicitante original, este almacena temporalmente en la memoria 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 los equipos Windows optimiza el rendimiento de la resolución de nombres DNS al almacenar también los nombres resueltos previamente en la memoria. El comando **ipconfig /displaydns** muestra todas las entradas DNS en caché en un sistema de computación Windows.

El protocolo DNS utiliza un sistema jerárquico para crear una base de datos que proporcione la resolución de nombres. La jerarquía es similar a un árbol invertido con la raíz en la parte superior y las ramas por debajo (consulte la ilustración). DNS utiliza nombres de domino para formar la jerarquía.

La estructura de denominación se divide en zonas pequeñas y manejables. Cada servidor DNS mantiene un archivo de base de datos específico y sólo es responsable de administrar las asignaciones de nombre a IP para esa pequeña porción de toda la estructura DNS. Cuando un servidor DNS recibe una solicitud para una traducción de nombre que no se encuentra dentro de esa zona DNS, el servidor DNS reenvía la solicitud a otro servidor DNS dentro de la zona adecuada para su traducción.

**Nota:** DNS es escalable, porque la resolución de los nombres de hosts se distribuye entre varios servidores.

Los diferentes dominios de primer nivel representan el tipo de organización o el país de origen. Entre los ejemplos de dominios del nivel superior se encuentran:

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

Después de los dominios del nivel superior, se encuentran los nombres de los dominios de segundo nivel y debajo de estos hay otros dominios de nivel inferior. Cada nombre de dominio es una ruta hacia este árbol invertido que comienza de la raíz. Por ejemplo, como se muestra en la ilustración, es posible que el servidor DNS raíz no sepa exactamente dónde se encuentra el registro del servidor de correo electrónico, mail.cisco.com, pero conserva un registro del dominio .com dentro del dominio de nivel superior. Asimismo, es posible que los servidores dentro del dominio .com no tengan un registro de mail.cisco.com, pero sí tienen un registro del dominio. Los servidores dentro del dominio cisco.com tienen un registro (un registro MX para ser precisos) para mail.cisco.com.

El DNS depende de esta jerarquía de servidores descentralizados para almacenar y mantener 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 servidor dado tiene registros de recursos que corresponden a su nivel en la jerarquía de dominios, se dice que es autoritativo para dichos registros. Por ejemplo, un servidor de nombre en el dominio cisco.netacad.net no sería autoritativo para el registro de mail.cisco.com, porque dicho registro se mantiene en un servidor de nivel de dominio superior, específicamente el servidor de nombre en el dominio cisco.com.

DNS es un servicio cliente/servidor. Sin embargo, difiere de los otros servicios cliente/servidor. 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 la resolución de nombres para otras aplicaciones de red y otros 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 (ISP) suministra las direcciones para utilizar con los servidores DNS. Cuando la aplicación del usuario pide conectarse a un dispositivo remoto por nombre, el cliente DNS solicitante consulta a uno de estos servidores de nombres para resolver el nombre para una dirección numérica.

Los sistemas operativos de las PC también cuentan con una utilidad llamada “nslookup” que permite que el usuario consulte los servidores de nombres de forma manual para resolver un nombre de host determinado. Esta utilidad también puede utilizarse para solucionar los problemas de resolución de nombres y verificar el estado actual de los servidores de nombres.

En la ilustración, cuando se ejecuta el comando **nslookup**, se muestra el servidor DNS predeterminado configurado para su host. En este ejemplo, el servidor DNS es dns-sj.cisco.com, que tiene la dirección 171.70.168.183.

El nombre de un host o de un dominio se puede introducir en la petición de entrada de **nslookup**. En la primera consulta de la ilustración, se hace una consulta parawww.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 realizar una prueba y una verificación exhaustivas del proceso DNS. Al finalizar, escriba **exit** para salir de la utilidad nslookup.

El servicio Protocolo de configuración dinámica de host (DHCP, Dynamic Host Configuration Protocol) 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, gateway y otros parámetros de redes IP. Esto se denomina “direccionamiento dinámico”. La alternativa al direccionamiento dinámico es el direccionamiento estático. Al utilizar el direccionamiento estático, el administrador de red introduce manualmente la información de la dirección IP en los hosts de red.

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

En redes locales más grandes, o donde los usuarios cambian con frecuencia, se prefiere asignar direcciones con DHCP. Es posible que los nuevos usuarios tengan computadoras portátiles y necesiten una conexión; otros pueden tener estaciones de trabajo nuevas que deben estar conectadas. En lugar de que el administrador de red asigne direcciones IP para cada estación de trabajo, es más eficaz que las direcciones IP se asignen automáticamente mediante el DHCP.

Las direcciones distribuidas por DHCP no se asignan de forma permanente a los hosts, sino que solo se conceden por un cierto período. Si el host se apaga o se desconecta de la red, la dirección regresa al pool para volver a utilizarse. Esto es especialmente útil en el caso de los usuarios móviles que entran en una red y salen de ella. 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 conecta el hardware, ya sea por cable o por LAN inalámbrica.

DHCP permite el acceso a Internet por medio de zonas de cobertura inalámbrica en aeropuertos o cafeterías. Cuando un dispositivo inalámbrico ingresa a una zona de cobertura, el cliente DHCP del dispositivo entra en contacto con el servidor de DHCP local mediante una conexión inalámbrica, y el servidor de DHCP asigna una dirección IP al dispositivo.

Como lo muestra la figura, varios tipos de dispositivos pueden ser servidores de DHCP cuando ejecutan software de servicio de DHCP. En la mayoría de las redes medianas a grandes, el servidor de DHCP suele ser un servidor local dedicado con base en una PC. En las redes domésticas, el servidor de DHCP suele estar ubicado en el router local que conecta la red doméstica al ISP. Los hosts locales reciben la información de la dirección IP directamente del router local. El router local recibe una dirección IP del servidor de DHCP en el ISP.

DHCP puede representar un riesgo a la seguridad porque cualquier dispositivo conectado a la red puede recibir una dirección. Este riesgo hace que la seguridad física sea un factor determinante para el uso del direccionamiento dinámico o manual. Tanto el direccionamiento dinámico como el estático tienen un lugar en el diseño de red. Muchas redes utilizan tanto el direccionamiento estático como el DHCP. DHCP se utiliza para hosts de uso general, como los dispositivos para usuarios finales, mientras que el direccionamiento estático se utiliza para dispositivos de red, como gateways, switches, servidores e impresoras.

Sin DHCP los usuarios tienen que introducir manualmente la dirección IP, la máscara de subred y otros parámetros de red para poder unirse a esta. 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 (concedidas) en lugar de estáticas (asignadas en forma permanente), las direcciones en desuso regresan automáticamente al pool para que se vuelvan a asignar. Como se muestra en la ilustración, cuando un dispositivo configurado con DHCP se inicia o se conecta a la red, el cliente transmite un mensaje de descubrimiento de DHCP (DHCPDISCOVER) para identificar cualquier servidor de DHCP disponible en la red. Un servidor de DHCP responde con un mensaje de oferta de DHCP (DHCPOFFER), que ofrece una concesión al cliente. El mensaje de oferta contiene la dirección IP y la máscara de subred que se deben asignar, la dirección IP del servidor DNS y la dirección IP del gateway predeterminado. La oferta de concesión también incluye la duración de esta.

El cliente puede recibir varios mensajes DHCPOFFER si hay más de un servidor de DHCP en la red local; por lo tanto, debe elegir entre ellos y enviar un mensaje de solicitud de DHCP (DHCPREQUEST) que identifique el servidor explícito y la oferta de concesión que el cliente acepta. Un cliente también puede optar por solicitar una dirección previamente asignada por el servidor.

Suponiendo que la dirección IP solicitada por el cliente, u ofrecida por el servidor, aún está disponible, el servidor devuelve un mensaje de acuse de recibo de DHCP (DHCPACK) que le informa al cliente que finalizó la concesión. Si la oferta ya no es válida, quizá debido a que hubo un tiempo de espera o a que otro cliente tomó la concesión, entonces el servidor seleccionado responde con un mensaje de acuse de recibo negativo de DHCP (DHCPNAK). Si se devuelve un mensaje DHCPNAK, entonces el proceso de selección debe volver a comenzar con la transmisión de un nuevo mensaje DHCPDISCOVER. Una vez que el cliente tiene la concesión, se debe renovar mediante otro mensaje DHCPREQUEST antes de que expire.

El servidor de DHCP asegura que todas las direcciones IP sean únicas (no se puede asignar la misma dirección IP a dos dispositivos de red diferentes de 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 direcciones a los clientes que no necesitan una dirección estática.

El protocolo de transferencia de archivos (FTP) es otro protocolo de capa de aplicación que se utiliza comúnmente. El protocolo FTP se desarrolló para permitir las transferencias de datos entre un cliente y un servidor. Un cliente FTP es una aplicación que se ejecuta en una PC y que se utiliza para insertar y extraer datos en un servidor que ejecuta un demonio FTP (FTPd).

Como se muestra en la ilustración, para transferir datos correctamente, FTP requiere dos conexiones entre el cliente y el servidor, una para los comandos y las respuestas y la otra para la transferencia de archivos propiamente dicha:

* El cliente establece la primera conexión al servidor para el tráfico de control, que está constituido por comandos del cliente y respuestas del servidor.
* El cliente establece la segunda conexión al servidor para la transferencia de datos propiamente dicha. Esta conexión se crea cada vez que hay datos para transferir.

La transferencia de datos se puede producir en ambas direcciones. El cliente puede descargar (extraer) datos del servidor o subir datos a él (insertarlos).

La capa de aplicación es responsable del acceso directo a los procesos subyacentes que administran y transmiten la comunicación a través de la red. Esta capa funciona como origen y destino de las comunicaciones a través de las redes de datos, independientemente del tipo de red de datos que se utilice. De hecho, los avances en la forma en que nos conectamos mediante redes tienen un impacto directo en el tipo de aplicaciones que están en desarrollo.

Las tendencias como Traiga su propio dispositivo (BYOD), el acceso desde cualquier lugar, la virtualización y las conexiones de máquina a máquina (m2m) abrieron el camino hacia una nueva generación de aplicaciones. Se estima que para el año 2020 habrá aproximadamente 50 000 millones de dispositivos conectados. Solo en 2010 se desarrollaron más de 350 000 aplicaciones, de las que se realizaron más de tres millones de descargas. Todo esto conduce a un mundo de conexiones intuitivas entre personas, procesos, datos y los elementos que están en la red.

Mediante el uso de etiquetas inteligentes y de la conectividad avanzada para digitalizar productos que no son de tecnología inteligente (desde bicicletas y botellas hasta refrigeradores y automóviles) y para conectarlos a Internet, las personas y las compañías podrán interactuar en formas nuevas e inimaginables. Los objetos podrán recopilar, recibir y enviar información a usuarios y a otros objetos conectados. Como se muestra en la ilustración, esta nueva ola de desarrollo de Internet se conoce como Internet de las cosas.

En la actualidad, existen más de 100 millones de máquinas expendedoras, vehículos, detectores de humo y otros dispositivos que ya comparten información automáticamente, una cifra que los analistas de mercado de [Berg Insight](http://www.berginsight.com/)   esperan que suba a 360 millones para el año 2016. Actualmente, las fotocopiadoras que cuentan con un módulo M2M pueden pedir tóner y papel nuevos en forma automática, o avisar a los técnicos sobre una falla; incluso pueden indicarles qué piezas deben traer.

La explosión masiva de aplicaciones se debe, en gran medida, al genio del enfoque en capas para el procesamiento de datos a través de una red. Concretamente, si se mantiene la funcionalidad de la capa de aplicación separada de la funcionalidad del transporte de datos, los protocolos de capa de aplicación se pueden modificar, y se pueden desarrollar nuevas aplicaciones, sin que el desarrollador deba preocuparse por el procedimiento de obtención de datos a través de la red. Esa es la funcionalidad de otras capas y, por lo tanto, de otros desarrolladores.

Como se muestra en la ilustración, cuando una aplicación envía una solicitud a una aplicación de servidor, la capa de aplicación construye el mensaje, pero después se pasa por las diversas funcionalidades de la capa en el cliente para entregarse. Mientras se traslada por el stack, cada capa inferior encapsula los datos con un encabezado que contiene los protocolos de comunicación para esa capa. Estos protocolos, que se implementan en los hosts emisores y receptores, interactúan para proporcionar una entrega extremo a extremo de las aplicaciones a través de la red.

Por ejemplo, los protocolos como el HTTP, admiten el envío de páginas Web a dispositivos finales. Ahora que conocemos todas las diversas capas y sus funcionalidades, podemos seguir la solicitud de una página Web de un cliente desde el servidor Web para ver cómo funciona cada una de estas funcionalidades individuales en forma completa y conjunta.

Si se utiliza el modelo TCP/IP, un proceso completo de comunicación incluye los siguientes seis pasos:

**Creación de los datos**

El primer paso es la creación de datos en la capa de aplicación del dispositivo final de origen que inicia la comunicación. En este caso, después de crear la solicitud del cliente Web, conocida como HTTP GET, esos datos se codifican, comprimen y encriptan, si es necesario. De esto se encarga el protocolo de capa de aplicación del modelo TCP/IP, pero incluye la funcionalidad descrita por las capas de aplicación, presentación y sesión del modelo OSI. La capa de aplicación envía estos datos como un stream a la capa de transporte.

**Segmentación y encapsulación inicial**

El siguiente paso es la segmentación y la encapsulación de los datos a medida que pasan por el stack de protocolos. En la capa de transporte, el mensaje HTTP GET se divide en partes más pequeñas y fáciles de manejar. A cada parte del mensaje se le agrega un encabezado de capa de transporte. Dentro del encabezado de la capa de transporte, hay indicadores que establecen cómo reconstruir el mensaje. También se incluye un identificador, el número de puerto 80. Este se utiliza para avisar al servidor de destino que se quiere enviar el mensaje a su aplicación de servidor Web. También se agrega un puerto de origen generado aleatoriamente para asegurarse de que el cliente pueda seguir la comunicación de retorno y reenviarla a la aplicación cliente correcta.

**Direccionamiento**

A continuación, se agregan los identificadores de dirección a los segmentos, como se muestra en la ilustración. Así como hay capas múltiples de protocolos que preparan los datos para la transmisión a su destino, hay capas múltiples de direccionamiento para asegurar su entrega. La función de la capa de red es agregar el direccionamiento que permite la transferencia de los datos desde el host que los origina hasta el host que los utiliza. La capa de red logra esto mediante el encapsulamiento de cada segmento en el encabezado del paquete IP. El encabezado del paquete IP contiene las direcciones IP del dispositivo de origen y de destino. (La dirección IP del dispositivo de destino se determina generalmente mediante un proceso de aplicación anterior conocido como “servicio de nombre de dominio”). La combinación de la dirección IP de origen y de destino con el número de puerto de origen y de destino se conoce como “socket”. El socket se utiliza para identificar el servidor y el servicio que solicita el cliente.

**Preparación para el transporte**

Después de agregar el direccionamiento IP, el paquete se traslada a la capa de acceso a la red para generar datos en los medios, como se muestra en la ilustración. Para que esto ocurra, la capa de acceso a la red primero debe preparar el paquete para la transmisión; para eso, lo coloca en una trama con un encabezado y un tráiler. Esta trama incluye la dirección física del host de origen y la dirección física del salto siguiente en la ruta al destino final. Esto equivale a la funcionalidad de la capa 2, o la capa de enlace de datos, del modelo OSI. La Capa 2 está relacionada con la entrega de los mensajes en una red local única. La dirección de la Capa 2 es exclusiva en la red local y representa la dirección del dispositivo final en el medio físico. En una LAN que utiliza Ethernet, esta dirección se denomina dirección de Control de acceso a los medios (MAC). Una vez que la capa de acceso a la red preparó la trama con las direcciones de origen y de destino, codifica la trama en bits y luego en impulsos eléctricos o en destellos de luz que se envían a través de los medios de red.

**Transporte de datos**

Los datos se transportan mediante la internetwork, que está compuesta por medios y por cualquier dispositivo intermedio. Como el mensaje encapsulado se transmite a través de la red, puede viajar a través de diferentes medios y tipos de red. La capa de acceso a la red especifica las técnicas para colocar la trama en los medios y quitarla de ellos, lo que se conoce también como “método de control de acceso al medio”.

Si el host de destino está en la misma red que el host de origen, el paquete se envía entre dos hosts en el medio local sin la necesidad de un router. Sin embargo, si el host de destino y el host de origen no están en la misma red, el paquete se puede transportar a través de muchas redes, en muchos tipos diferentes de medios y a través de muchos routers. Cuando pasa por la red, la información contenida en la trama no se modifica.

En los límites de cada red local, un dispositivo de red intermedio, por lo general, un router, desencapsula la trama para leer la dirección de host de destino incluida en el encabezado del paquete. Los routers utilizan la porción del identificador de red de esta dirección para determinar qué ruta utilizar para llegar al host de destino. Una vez que se determina la ruta, el router encapsula el paquete en una nueva trama y lo envía al siguiente salto del trayecto hacia el dispositivo final de destino.

**Entrega de datos a la aplicación de destino correcta**

Por último, se recibe la trama en el dispositivo final de destino. Cuando los datos pasan por el stack en el dispositivo de destino, estos se desencapsulan y se vuelven a armar. Los datos pasan continuamente por las capas, de la capa de acceso a la red a la capa de red y a la capa de transporte, hasta que llegan a la capa de aplicación para ser procesados. Pero ¿cómo puede estar seguro el dispositivo de que se identifique el proceso de aplicación correcto?

Como se muestra en la ilustración, recuerde que en la capa de transporte, la información que contiene el encabezado PDU identifica el proceso o el servicio específico que se ejecuta en el dispositivo del host de destino que funcionará con los datos. Los hosts, sean clientes o servidores en Internet, pueden ejecutar múltiples aplicaciones de red simultáneamente. Las personas que utilizan PC suelen tener un cliente de correo electrónico que se ejecuta al mismo tiempo que el explorador Web, un programa de mensajería instantánea, algún streaming media y, tal vez, incluso algún juego. Todos estos programas ejecutándose en forma separada son ejemplos de procesos individuales.

Ver una página Web invoca al menos un proceso de red. Hacer clic en un hipervínculo hace que un explorador Web se comunique con un servidor Web. Al mismo tiempo, en segundo plano, es posible que un cliente de correo electrónico envíe o reciba un correo y un colega o amigo envíe un mensaje instantáneo.

Piense en una computadora que tiene sólo una interfaz de red. Todos los streams de datos que crean las aplicaciones que se ejecutan en la PC entran y salen a través de esa interfaz; no obstante, los mensajes instantáneos no aparecen de repente en medio de documentos del procesador de textos y los correos electrónicos no aparecen en la interfaz de un juego.

Esto es porque los procesos induviduales que se ejecutan en los hosts de origen y destino se comunican unos con otros. Cada aplicación o servicio se representa por un número de puerto en la Capa 4. Un diálogo único entre dispositivos se identifica con un par de números de puerto de origen y de destino de Capa 4 que son representativos de las dos aplicaciones de comunicación. Cuando los datos se reciben en el host, se examina el número de puerto para determinar qué aplicación o proceso es el destino correcto de los datos.

Un recurso de entretenimiento para ayudar a visualizar los conceptos de red es la película animada “Warriors of the Net” (Guerreros de la red), de TNG Media Lab. Antes de ver el video, se debe tener en cuenta lo siguiente: Primero, en cuanto a los conceptos que ha aprendido en este capítulo, piense en qué momento del video está en la LAN, en la WAN, en intranet o en Internet, y cuáles son los dispositivos finales vs. los dispositivos intermedios, cómo se aplican los modelos OSI y TCP/IP y qué protocolos están involucrados.

En segundo lugar, si bien se hace referencia a los números de puerto 21, 23, 25, 53 y 80 de forma explícita en el video, se hace referencia a las direcciones IP solo de forma implícita, ¿puede ver dónde? ¿Dónde se pudieron involucrar las direcciones MAC en el video?

Finalmente, aunque todas las animaciones con frecuencia tienen simplificaciones en ellas, hay un error categórico en el video. Aproximadamente a los cinco minutos se afirma lo siguiente: “Lo que el Sr. IP hace cuando no recibe un acuse de recibo es enviar un paquete de reemplazo.” Esta no es una función del protocolo de Internet de capa 3, que es “poco confiable”, y de entrega de máximo esfuerzo, sino que es una función del protocolo TCP de la capa de transporte.

Descargue la película de [http://www.warriorsofthe.net](http://www.warriorsofthe.net/)  .

La capa de aplicación es responsable del acceso directo a los procesos subyacentes que administran y transmiten la comunicación a través de la red humana. Esta capa sirve como origen y destino de las comunicaciones a través de las redes de datos. Las aplicaciones, los protocolos y los servicios de la capa de aplicación permiten a los usuarios interactuar con la red de datos de manera significativa y eficaz.

* Las aplicaciones son programas informáticos con los que el usuario interactúa y que inician el proceso de transferencia de datos a solicitud del usuario.
* Los servicios son programas en segundo plano que proporcionan conexión entre la capa de aplicación y las capas inferiores del modelo de red.
* Los protocolos proporcionan una estructura de reglas y procesos acordados que garantizan que los servicios que se ejecutan en un dispositivo particular puedan enviar y recibir datos de una variedad de dispositivos de red diferentes.

Un cliente puede solicitar a un servidor la entrega de datos por la red, o se puede solicitar entre dispositivos que funcionan en una disposición P2P, donde la relación cliente/servidor se establece según el dispositivo de origen y de destino en ese momento. Los mensajes se intercambian entre los servicios de la capa de aplicación en cada dispositivo final de acuerdo con las especificaciones del protocolo para establecer y utilizar estas relaciones.

Por ejemplo, los protocolos como el HTTP, admiten el envío de páginas Web a dispositivos finales. SMTP y POP admiten el envío y la recepción de correo electrónico. SMB y FTP permiten compartir archivos a los usuarios. Las aplicaciones P2P facilitan a los consumidores la tarea de compartir medios sin inconvenientes de una manera distribuida. DNS resuelve los nombres utilizados para referirse a los recursos de red en direcciones numéricas utilizables por la red. Las nubes son ubicaciones ascendentes remotas que almacenan datos y aplicaciones host, de modo que los usuarios no requieran tantos recursos locales y para que puedan acceder al contenido sin inconvenientes desde distintos dispositivos en cualquier ubicación.

Todos estos elementos funcionan conjuntamente, en la capa de aplicación. La capa de aplicación permite que los usuarios trabajen y jueguen a través de Internet.