FUNDAMENTOS DE REDES.
TEMAS 3 Y 4.

ÍNDICE

TEMA 3

1. Las funciones de la capa de aplicación

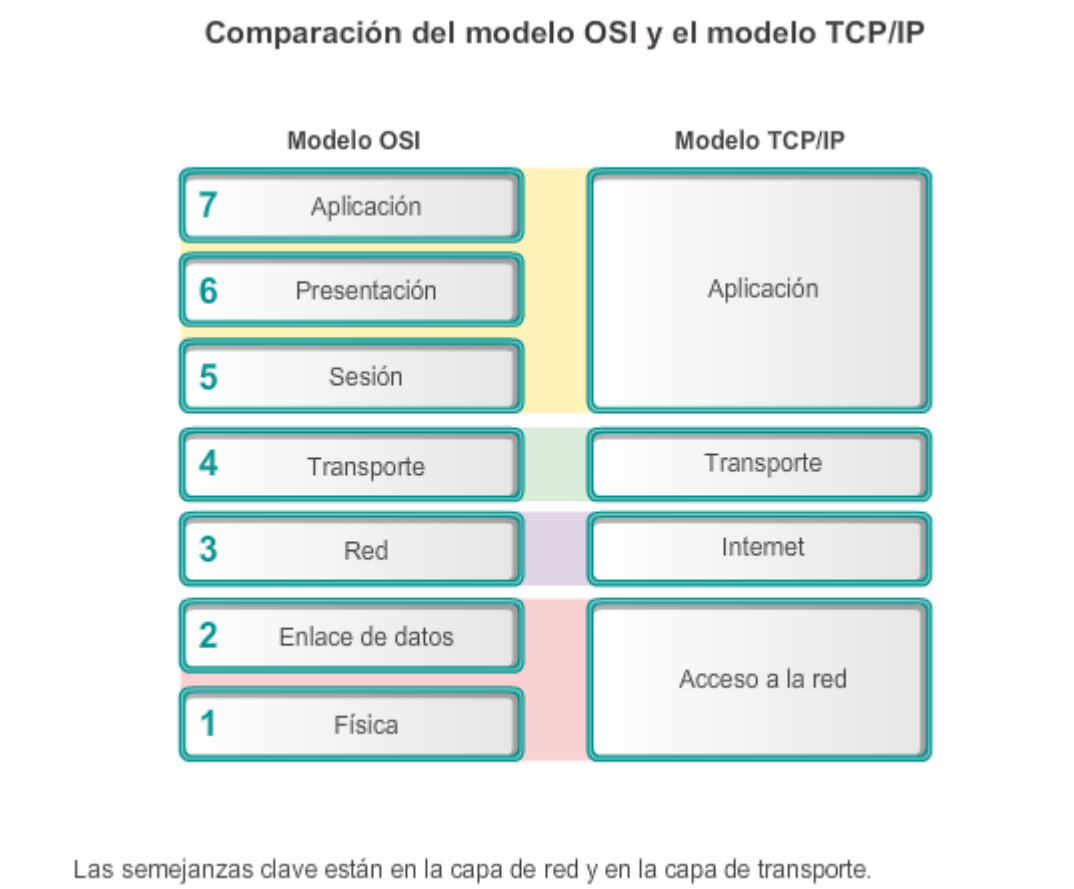


Ilustración 1

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.

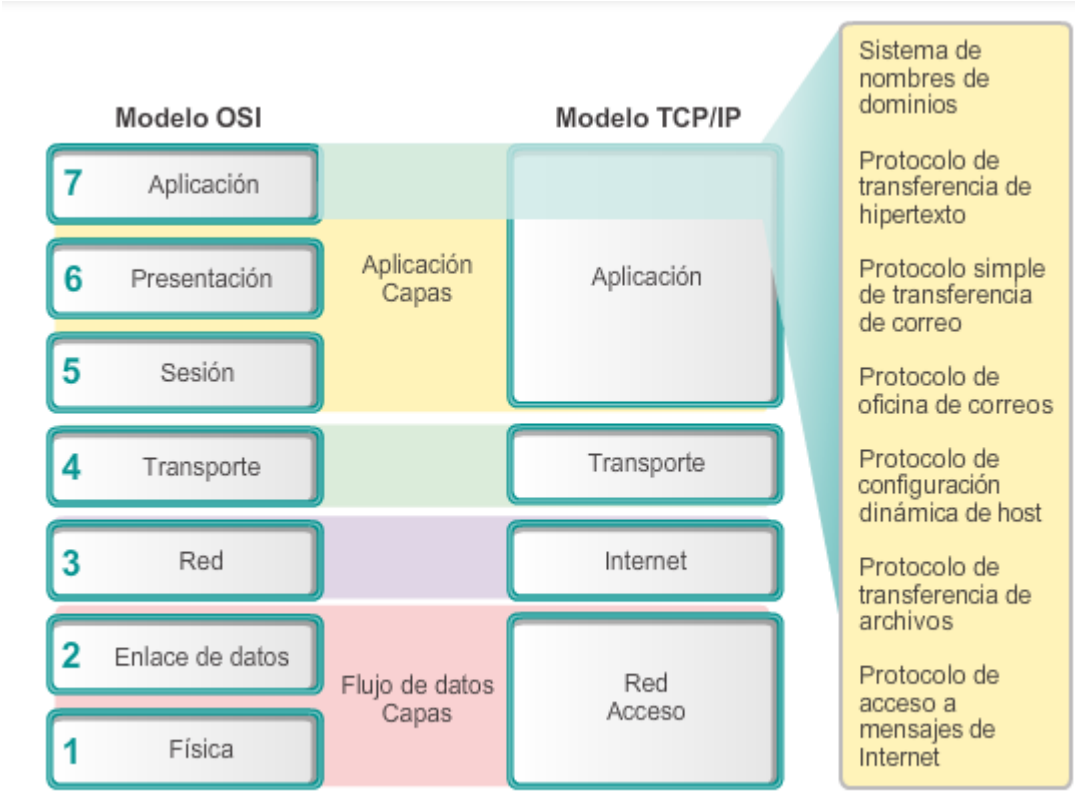


Ilustración 2

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).

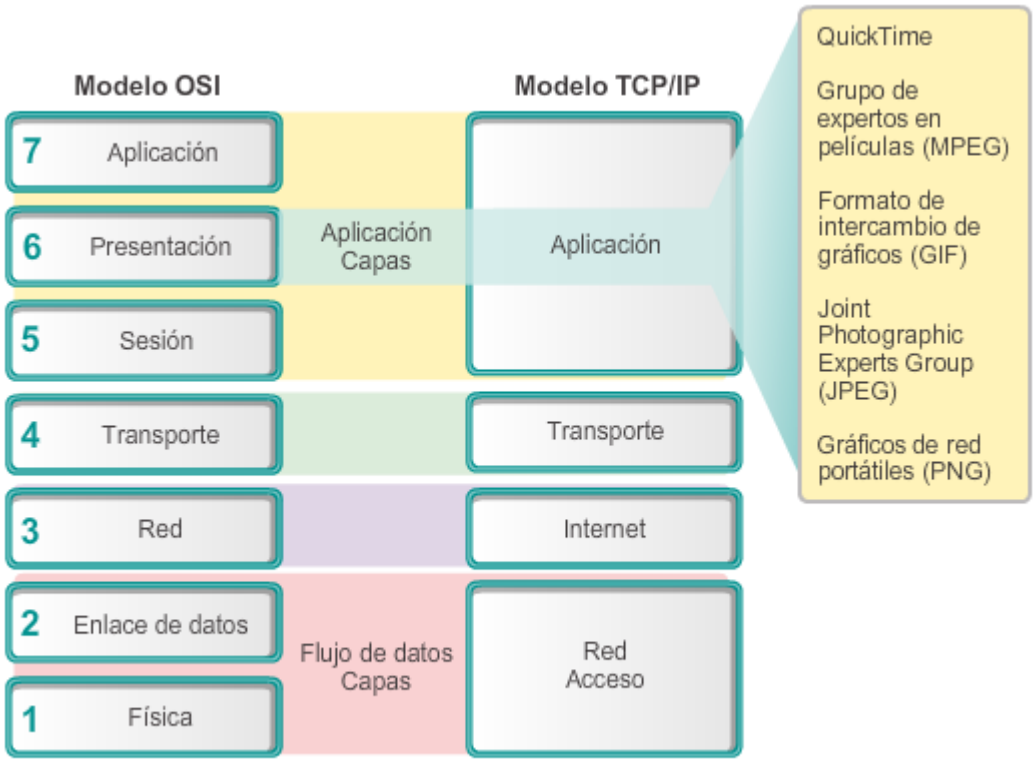


Ilustración 3

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.

2. Modelo de aplicaciones y servicios de red.

2.1 Modelo P2P

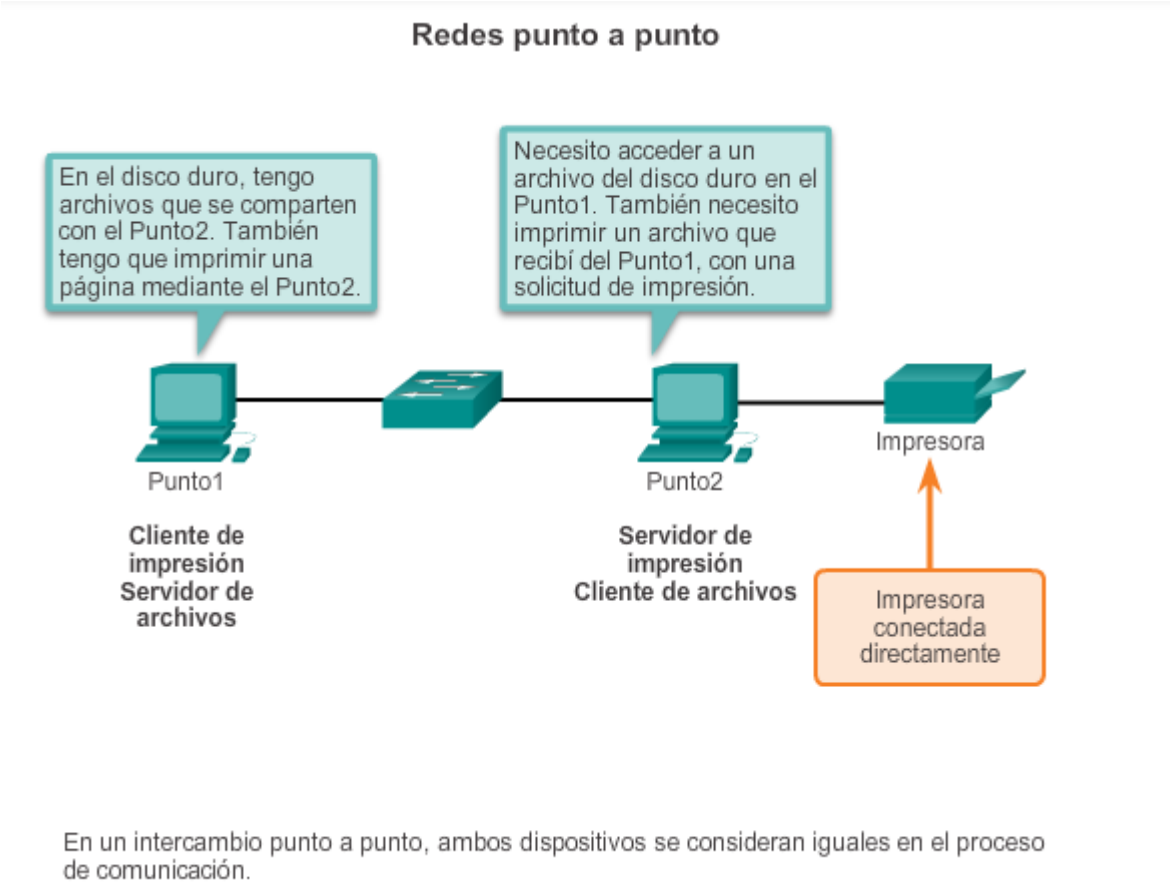


Ilustración 4

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.

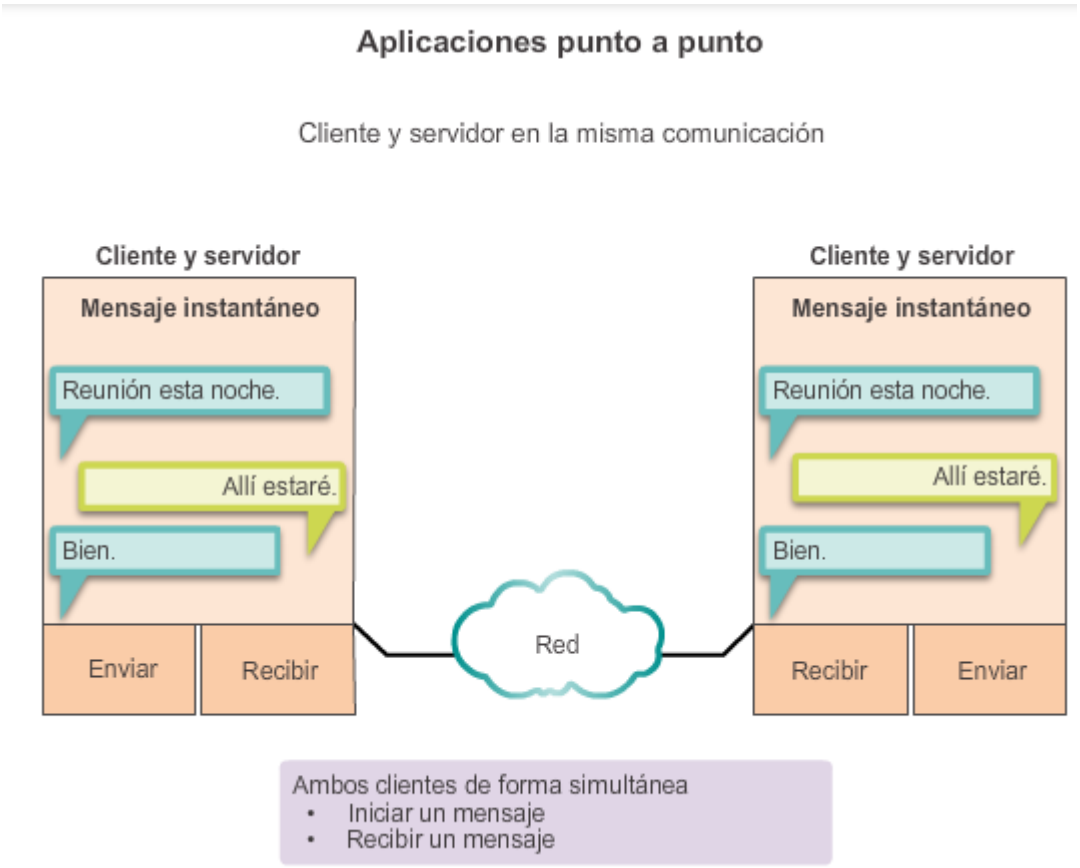
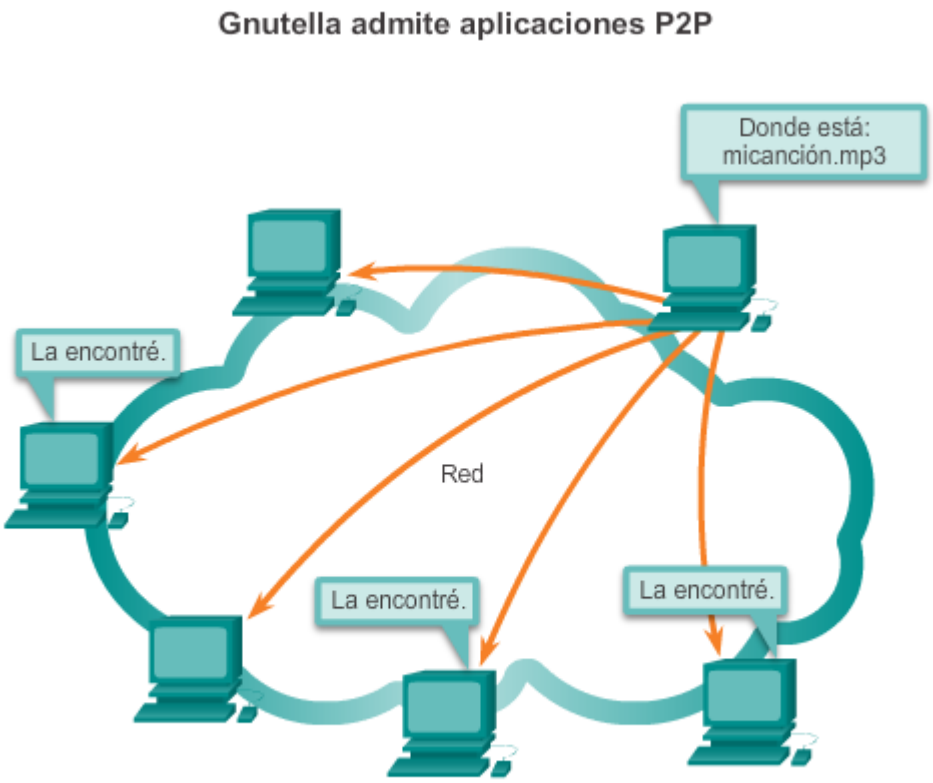


Ilustración 5

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.



Gnutella permite que las aplicaciones P2P busquen recursos compartidos entre puntos.

Ilustración 6

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.

2.2 Modelo cliente/servidor

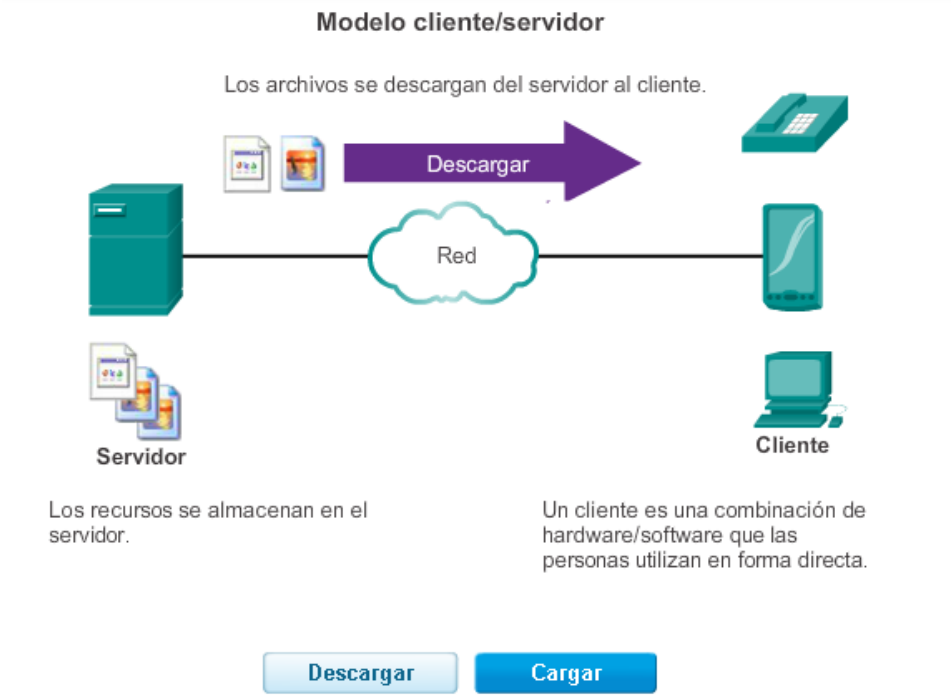


Ilustración 7

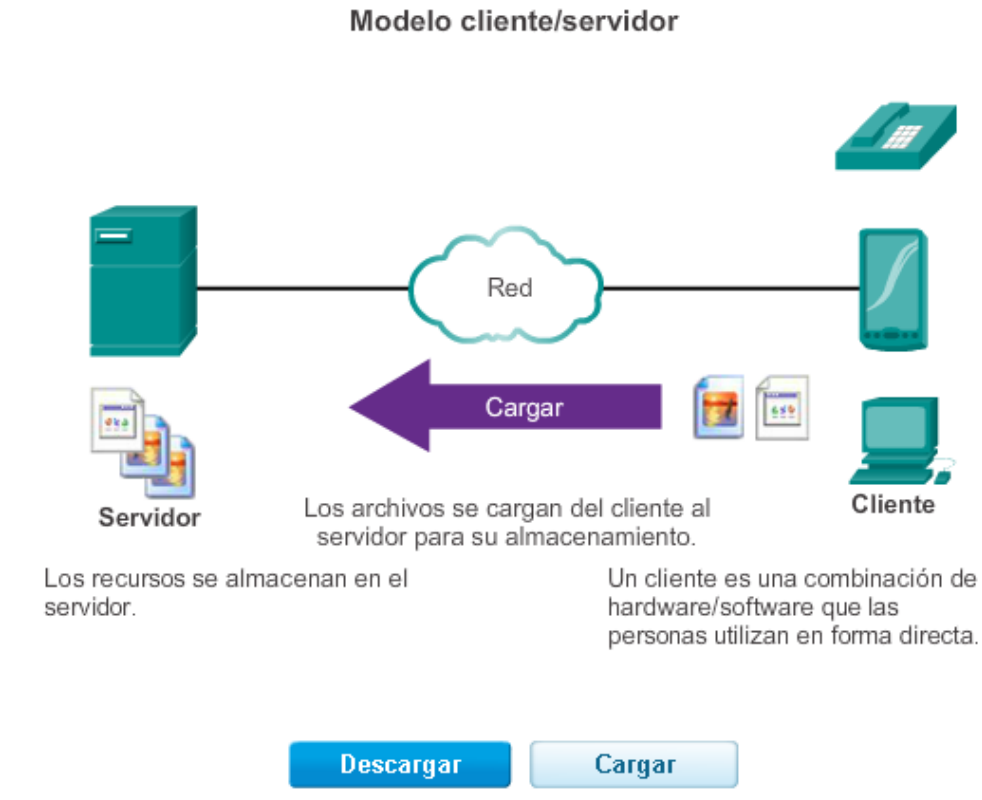


Ilustración 8

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”.

3. Protocolos y servicios de capa de aplicación conocidos

3.1 World Wide Web: HTTP y HTTPS.

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:

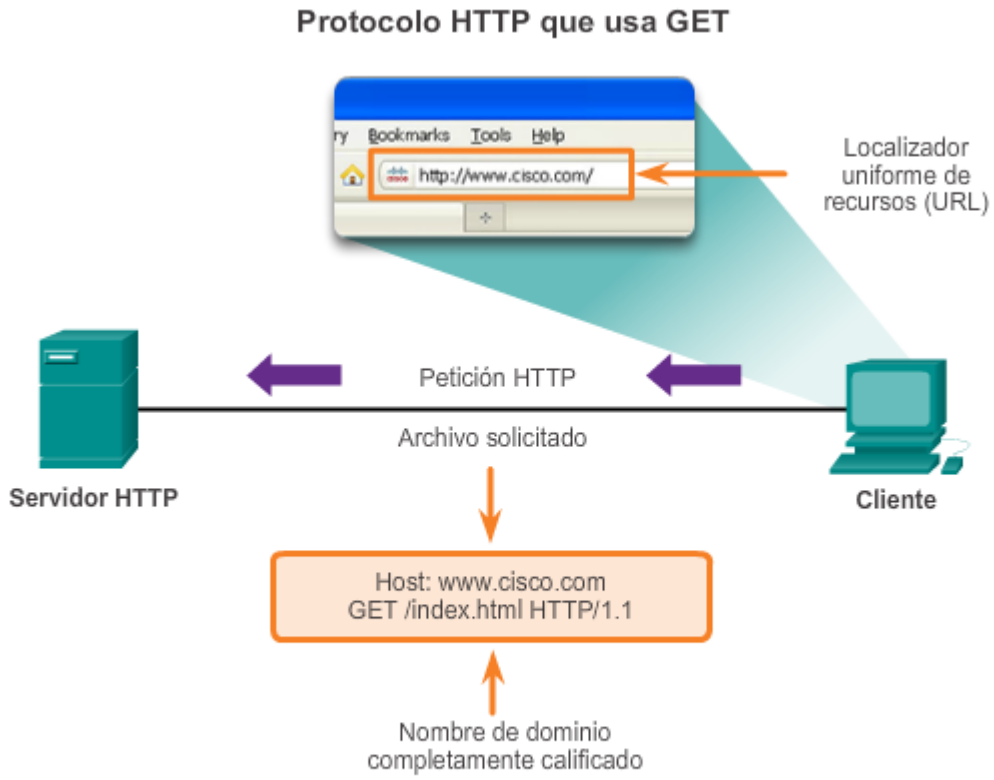
1. **http** (el protocolo o esquema)

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.

NOTA: Para ver lo que muestran las figuras, pinchar aquí.



Al escribir "http://www.cisco.com" en la barra de dirección de un explorador Web, se genera el mensaje HTTP "GET".

Ilustración 9

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”.

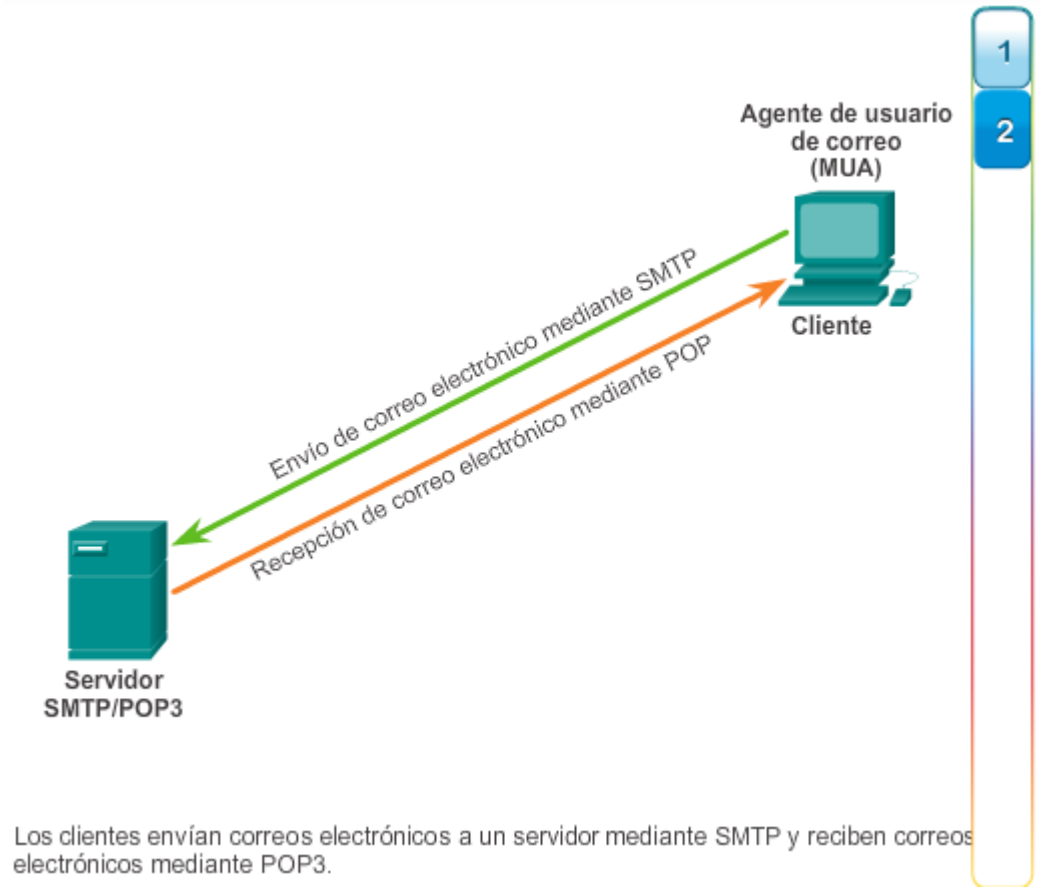
Los mensajes POST y PUT se utilizan para subir datos al servidor Web. Por ejemplo, cuando el usuario introduce datos en un formulario que está integrado en una página Web (p. ej., cuando se completa una solicitud de pedido), el mensaje POST se envía al servidor Web. En el mensaje POST, se incluyen los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

3.2 Correo electrónico: SMTP, POP e IMAP.



Los clientes envían correos electrónicos a un servidor mediante SMTP y reciben correos electrónicos mediante POP3.

Ilustración 10

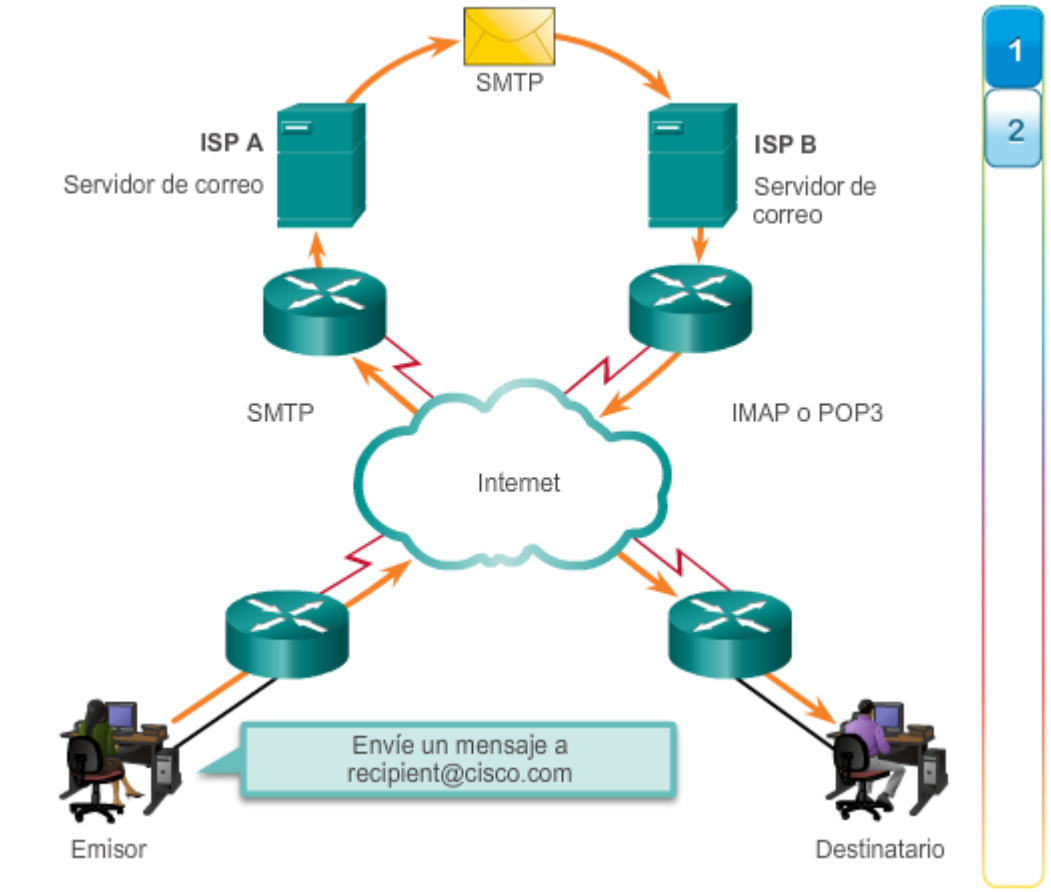


Ilustración 11

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.

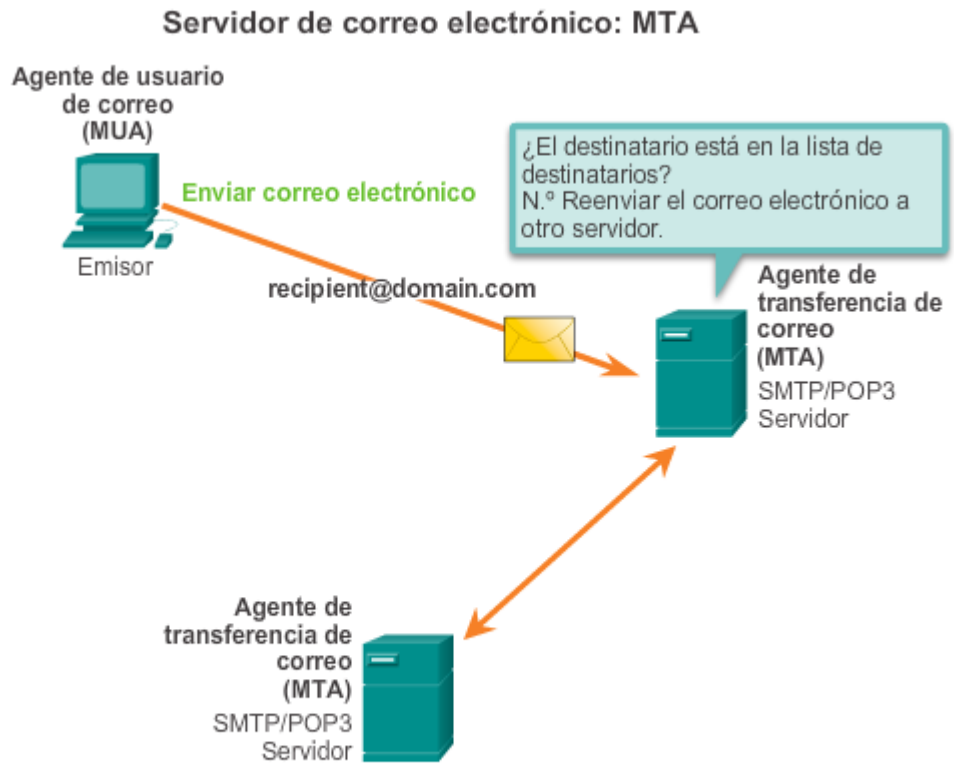
El correo electrónico es un método para almacenar y enviar que se utiliza para enviar, almacenar y recuperar mensajes electrónicos a través de una red. Los mensajes de correo electrónico se guardan en bases de datos en servidores de correo. A menudo, los ISP mantienen servidores de correo que admiten 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 proceso de agente de transferencia de correo rige el envío de correo electrónico entre los servidores y los clientes.

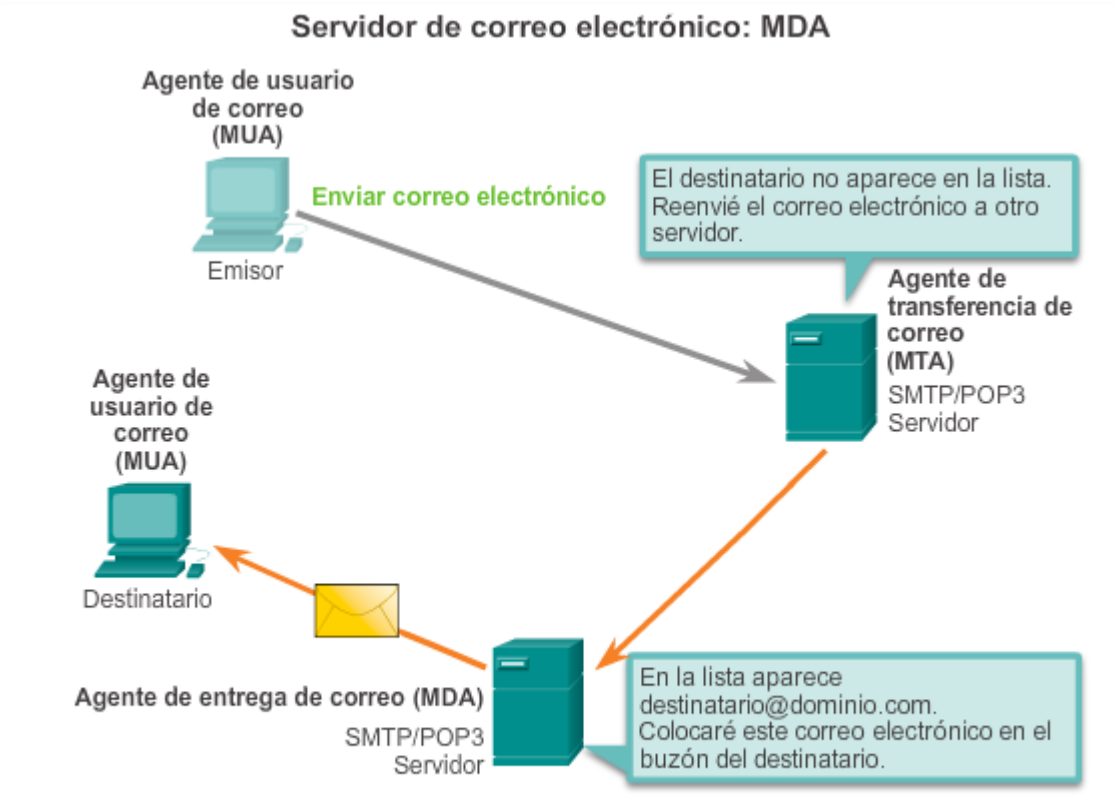
Ilustración 12

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 proceso de agente de entrega de correo rige la entrega de correo electrónico entre los servidores y los clientes.

Ilustración 13

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.

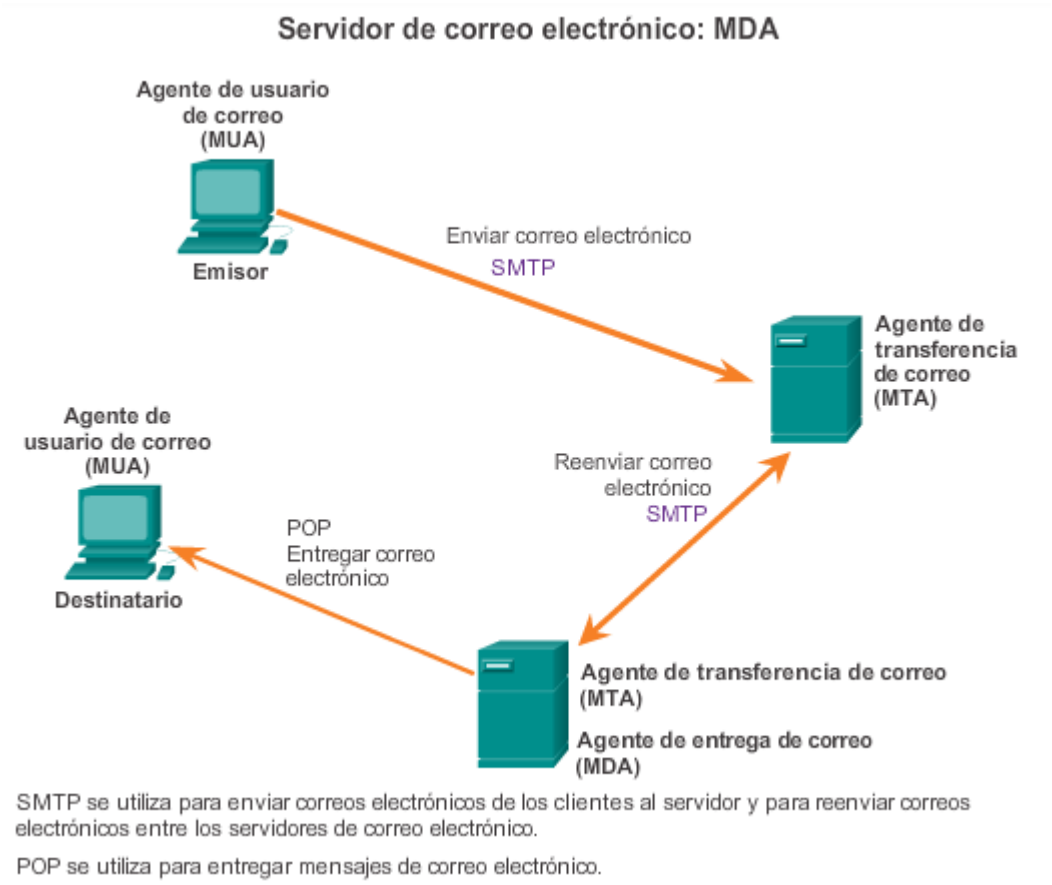


Ilustración 14

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.

3.3 Sistema de Nombre de Dominio: DNS.

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, 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, 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.

NOTA: Para seguir las figuras pincha aquí.

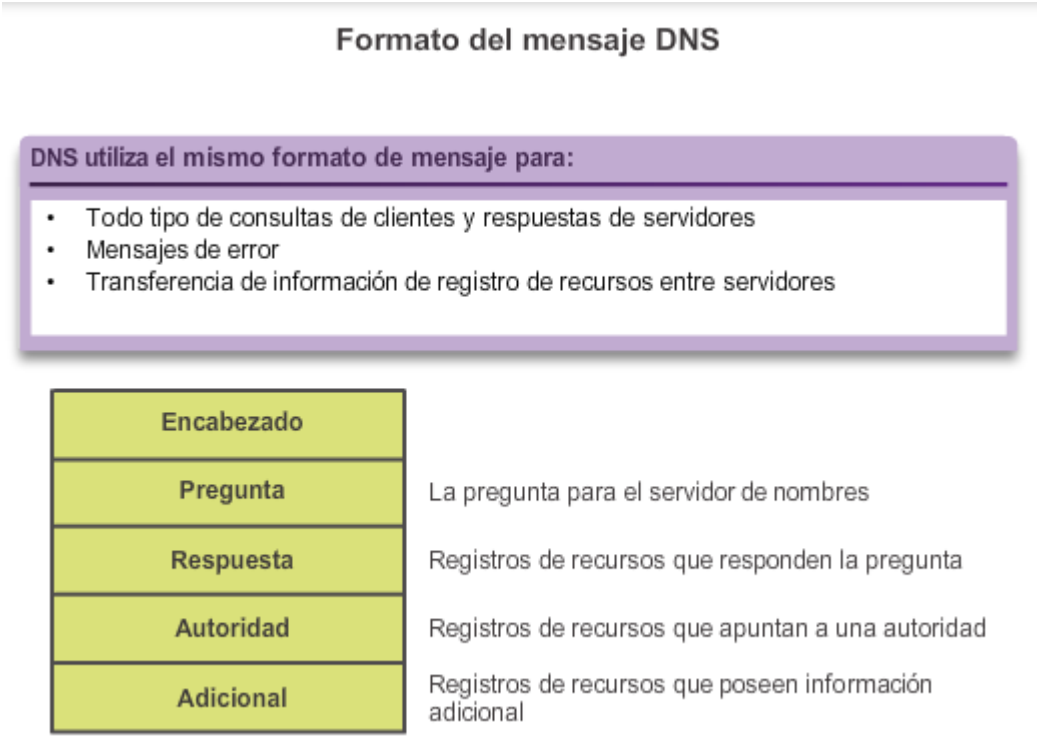


Ilustración 15

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.

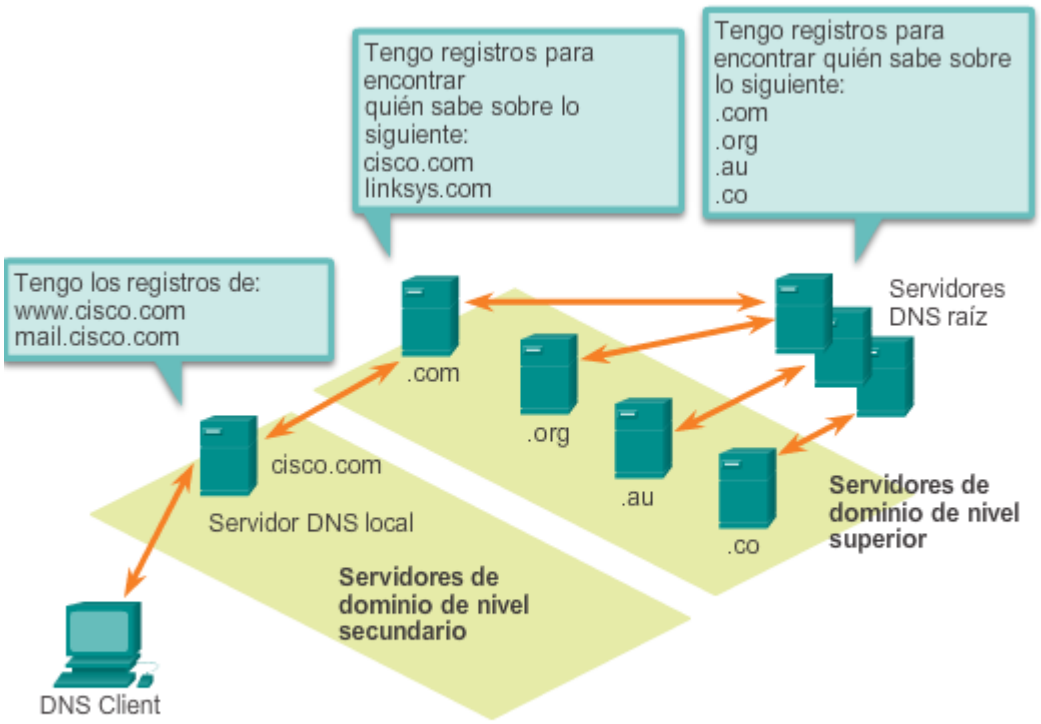
Algunos de estos tipos de registros son:

- **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.



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

Ilustración 16

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.

3.4 Configuración Automática de Host: DHCP.

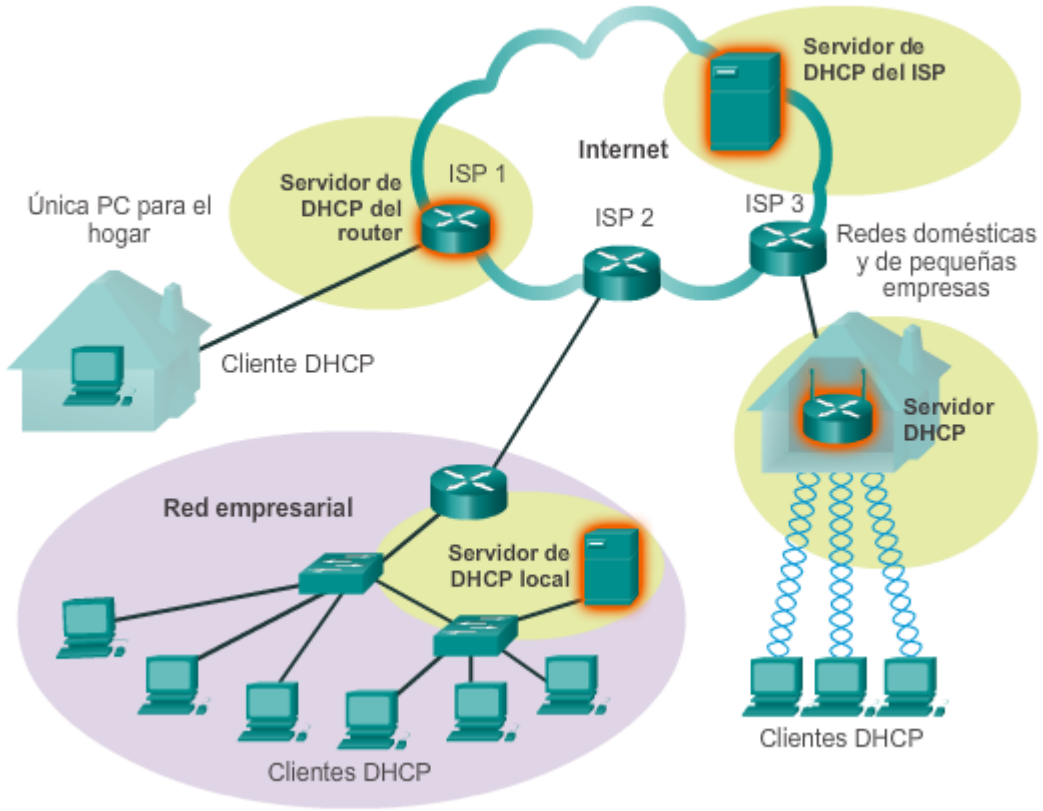


Ilustración 17

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.

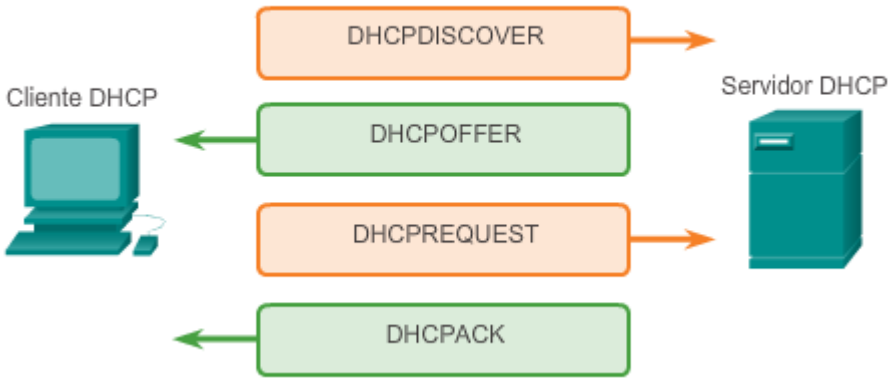


Ilustración 18

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.

3.5 Transferencia de archivos: FTP.

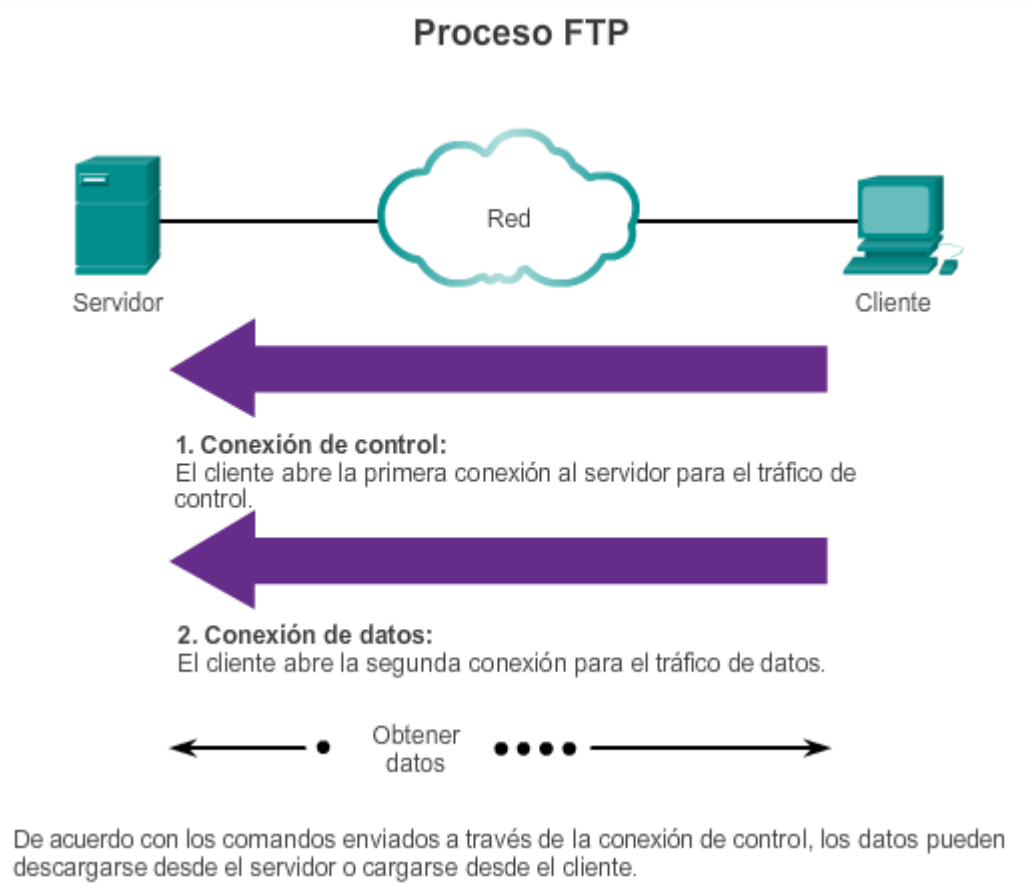


Ilustración 19

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).

3.6 Compartir archivos e impresoras: SMB.

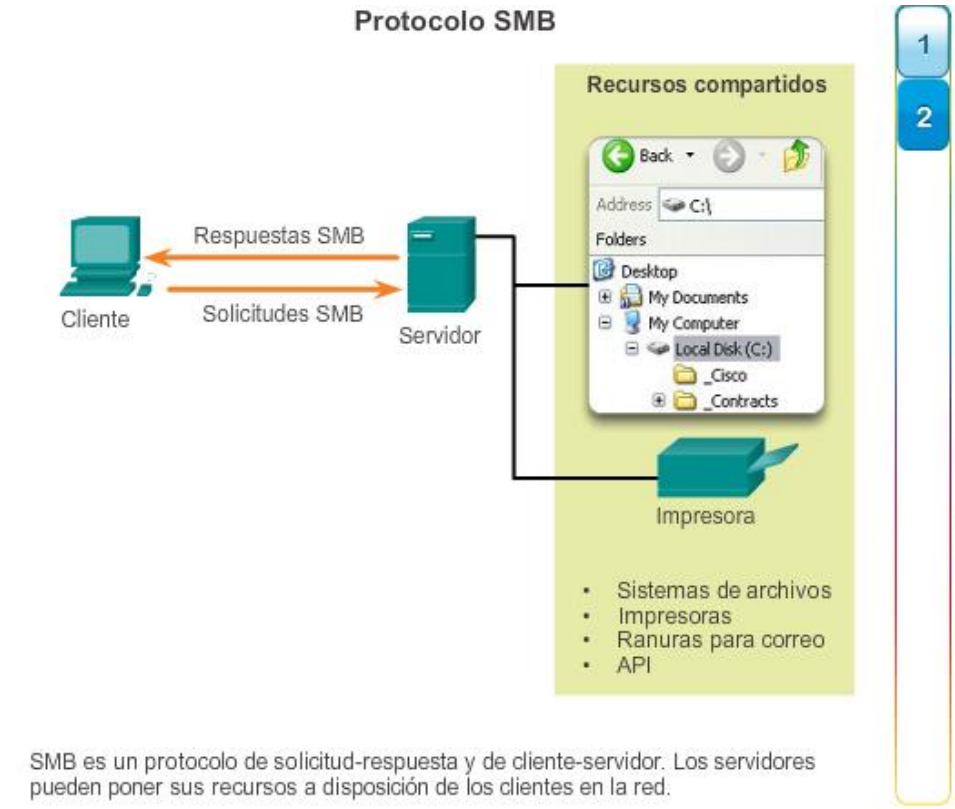


Ilustración 20

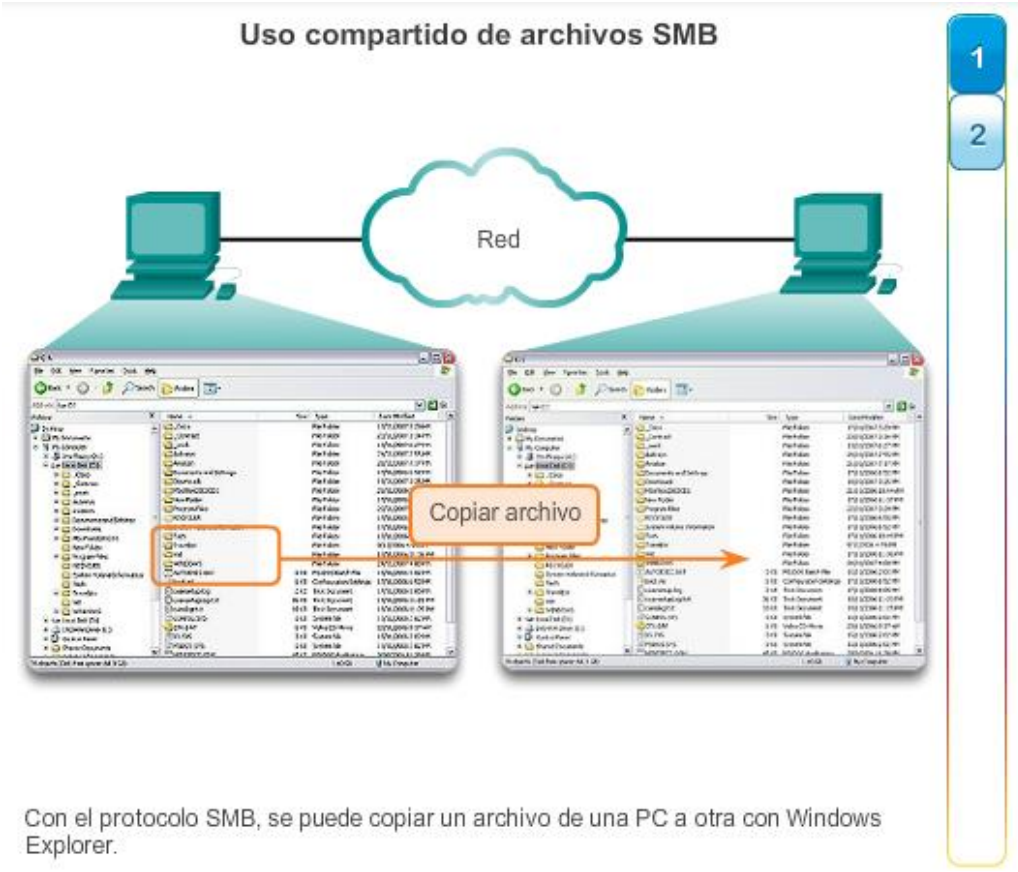


Ilustración 21

El bloque de mensajes del servidor (SMB) es un protocolo de intercambio de archivos cliente/servidor que desarrolló IBM a fines de la década de los ochenta para describir la estructura de los recursos de red compartidos, como archivos, directorios, impresoras y puertos serie. Es un protocolo de solicitud-respuesta.

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 de un parámetro de tamaño variable y un componente de datos.

Los mensajes de SMB pueden:

- Iniciar, autenticar y terminar sesiones
- Controlar el acceso a los archivos y a la impresora
- Autorizar una aplicación para enviar o recibir mensajes para o de otro dispositivo

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 de software Windows 2000, 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. A partir de Windows 2000, todos los productos subsiguientes de Microsoft utilizan la convención de nomenclatura DNS, que permite que los protocolos TCP/IP admitan directamente el uso compartido de recursos de SMB, como se muestra en la figura 1. El proceso de intercambio de archivos de SMB entre equipos Windows se muestra en la figura 2.

A diferencia del uso compartido de archivos que admite el protocolo de transferencia de archivos (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 sistemas operativos LINUX y UNIX también proporcionan un método de intercambio de recursos con redes de Microsoft mediante una versión del SMB llamado SAMBA. Los sistemas operativos Macintosh de Apple también admiten recursos compartidos utilizando el protocolo SMB.

3.7 Internet de las cosas (IoT): M2M

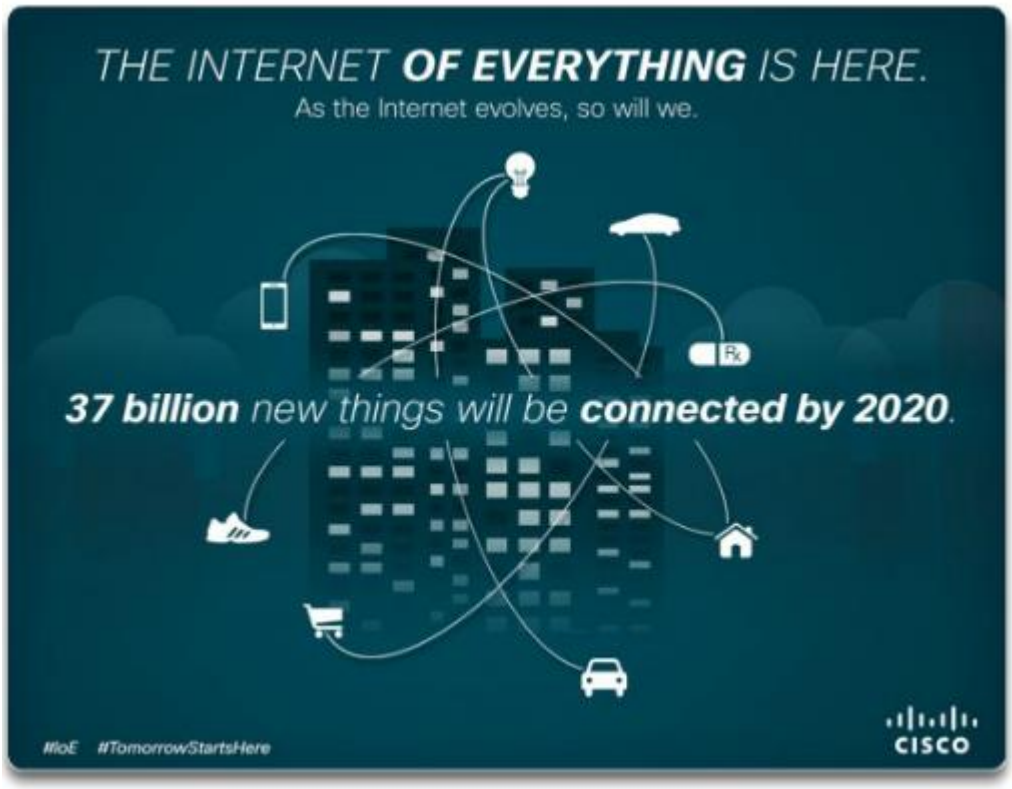


Ilustración 22

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.

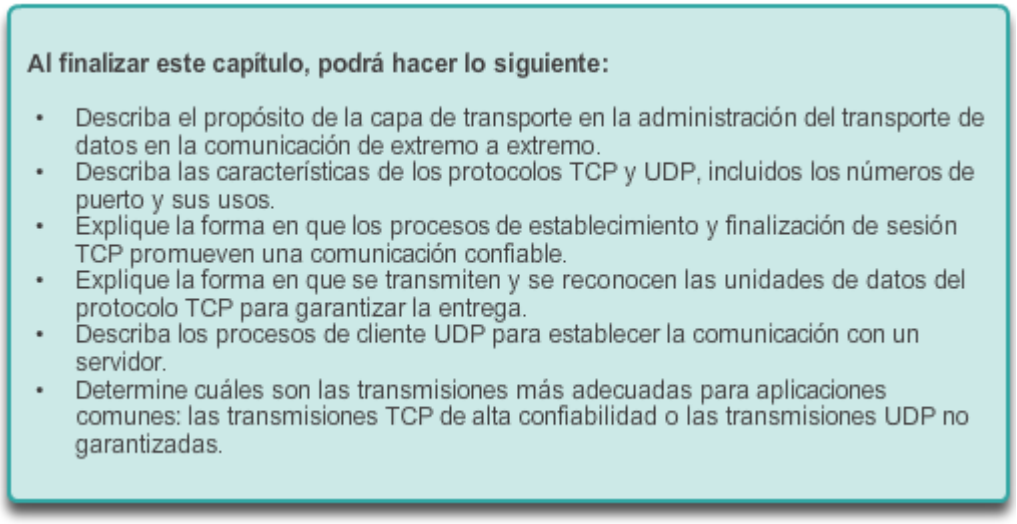
Las tendencias como Traiga su propio dispositivo (BYOD), el acceso desde cualquier lugar, la virtualización y las conexiones de máquina a máquina (m2m) abrieron el camino hacia una nueva generación de aplicaciones. Se estima que para el año 2020 habrá aproximadamente 50 000 millones de dispositivos conectados. 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 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.

TEMA 4

1. Funciones de la capa de transporte.



T4. Ilustración 1

Las redes de datos e Internet brindan soporte a la red humana por medio del suministro de comunicación confiable entre personas. En un único dispositivo, las personas pueden utilizar varias aplicaciones y diversos servicios, como correo electrónico, la Web y la mensajería instantánea, para enviar mensajes o recuperar información. Las aplicaciones, como los clientes de correo electrónico, los exploradores Web y los clientes de mensajería instantánea, permiten que las personas usen PC y redes para enviar mensajes y encontrar información.

Los datos de cada una de estas aplicaciones se empaquetan, se transportan y se entregan a la aplicación correspondiente en el dispositivo de destino. Los procesos que se describen en la capa de transporte del modelo OSI aceptan los datos de la capa de aplicación y los preparan para el direccionamiento en la capa de red. La capa de transporte **prepara** los datos para transmitirlos a través de la red. La PC de origen se comunica con una PC receptora para decidir cómo dividir los datos en **segmentos**, cómo asegurarse de que ninguno de los segmentos se pierda y cómo verificar si llegan todos los segmentos. Al considerar la capa de transporte, imagínese un departamento de envíos que prepara un único pedido de varios paquetes para entregar.

En este capítulo, se examina el rol de la capa de transporte en el encapsulamiento de datos de aplicación que utiliza la capa de red. La capa de transporte incluye también las siguientes funciones:

- Permite que varias aplicaciones, como el envío de correo electrónico y las redes sociales, se puedan comunicar a través la red al mismo tiempo en un único dispositivo.
- Asegura que, si es necesario, la aplicación correcta reciba todos los datos con confianza y en orden.
- Emplea mecanismos de manejo de errores.

Objetivos de aprendizaje

Al completar este capítulo, usted podrá:

- Explicar la necesidad de la capa de transporte.
- Identificar la función de la capa de transporte a medida que provee la transferencia de datos de extremo a extremo entre las aplicaciones.
- Describir la función de dos protocolos de la capa de transporte TCP/IP: TCP y UDP.
- Explicar las funciones clave de la capa de transporte, incluso la confiabilidad, el direccionamiento de puerto y la segmentación.
- Explicar cómo cada TCP y UDP maneja las funciones clave.
- Identificar cuándo es apropiado usar TCP o UDP y proveer ejemplos de aplicaciones que usan cada protocolo.

T4. Ilustración 2

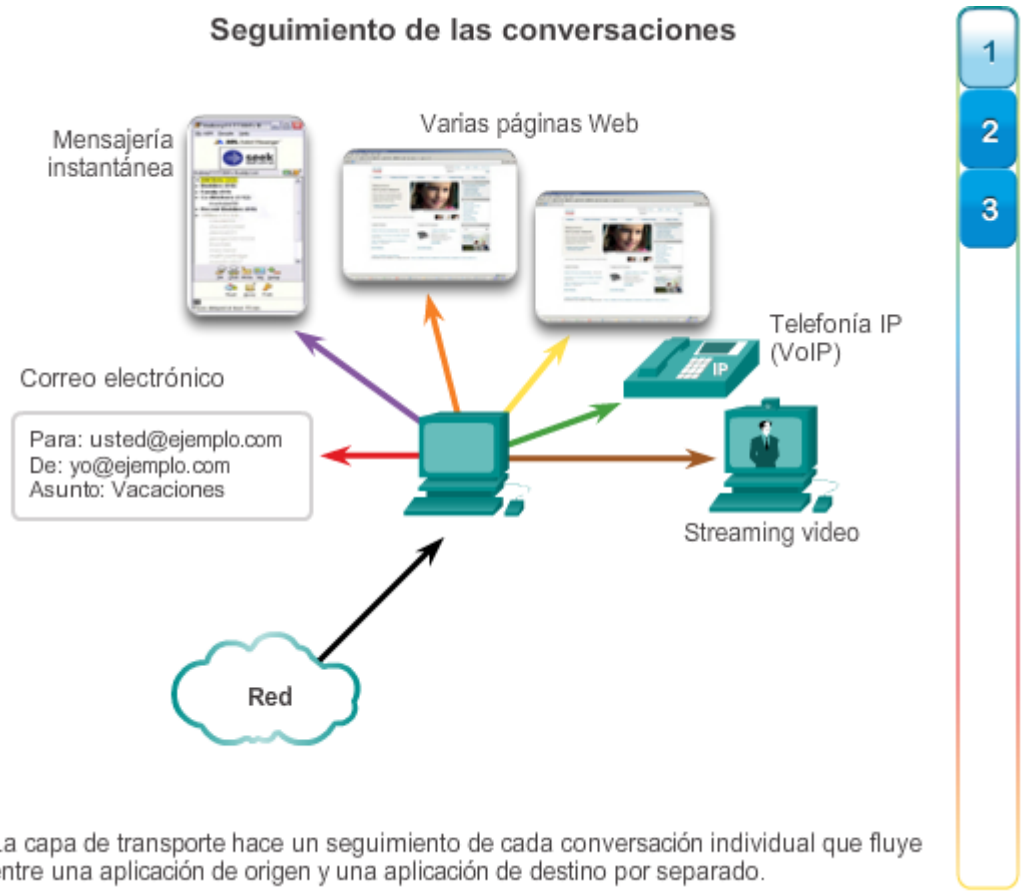
La capa de transporte es responsable de establecer una sesión de comunicación temporal entre dos aplicaciones y de transmitir datos entre ellas. Las aplicaciones generan los datos que se envían de una aplicación en un host de origen a una aplicación a un host de destino, independientemente del tipo de host de destino, el tipo de medios a través de los que deben viajar los datos, la ruta que toman los datos, la congestión en un enlace o el tamaño de la red. Como se muestra en la ilustración, la capa de transporte es el enlace entre la capa de aplicación y las capas inferiores que son responsables de la transmisión a través de la red.

La capa de transporte proporciona un método para entregar datos a través de la red de una manera que garantiza que estos se puedan volver a unir correctamente en el extremo receptor. La capa de transporte permite la segmentación de datos y proporciona el control necesario para rearmar estos segmentos en los distintos streams de comunicación. En el protocolo TCP/IP, estos procesos de

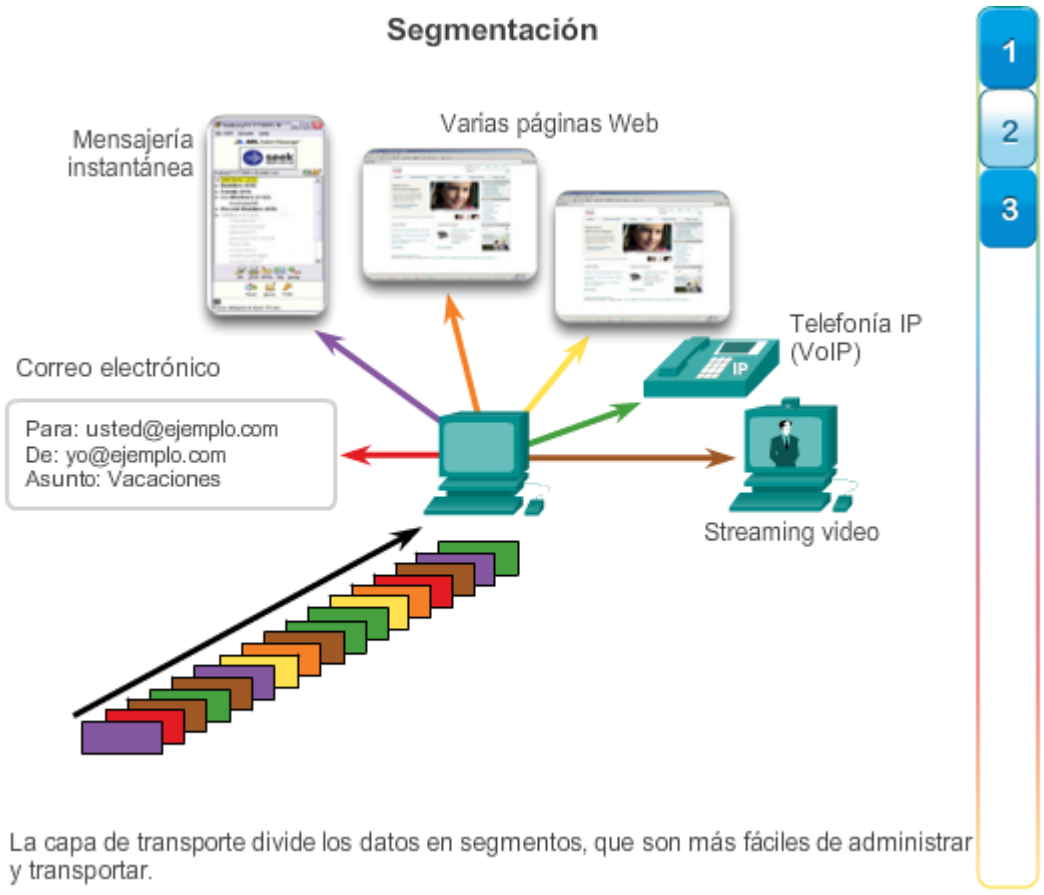
segmentación y rearmado se pueden lograr utilizando dos protocolos muy diferentes de la capa de transporte: el protocolo de control de transmisión (TCP) y el protocolo de datagramas de usuario (UDP).

Las principales responsabilidades de los protocolos de la capa de transporte son las siguientes:

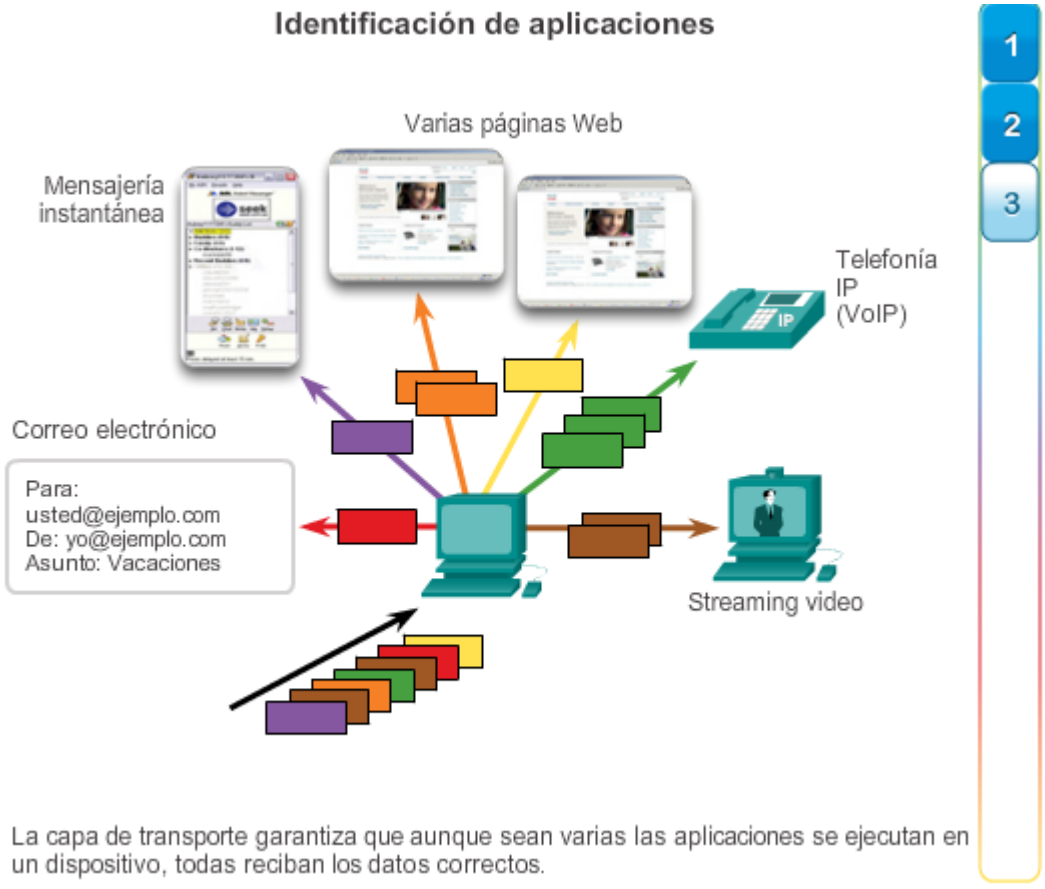
- Rastreo de comunicación individual entre aplicaciones en los hosts de origen y destino
- División de los datos en segmentos para su administración y reunificación de los datos segmentados en streams de datos de aplicación en el destino
- Identificación de la aplicación correspondiente para cada stream de comunicación.



T4. Ilustración 3



T4. Ilustración 4



T4. Ilustración 5

Rastreo de conversaciones individuales

En la capa de transporte, cada conjunto de datos particular que fluye entre una aplicación de origen y una de destino se conoce como “conversación” (figura 1). Un host puede tener varias aplicaciones que se comunican a través de la red de forma simultánea. Cada una de estas aplicaciones se comunica con una o más aplicaciones en uno o más hosts remotos. Es responsabilidad de la capa de transporte mantener y hacer un seguimiento de todas estas conversaciones.

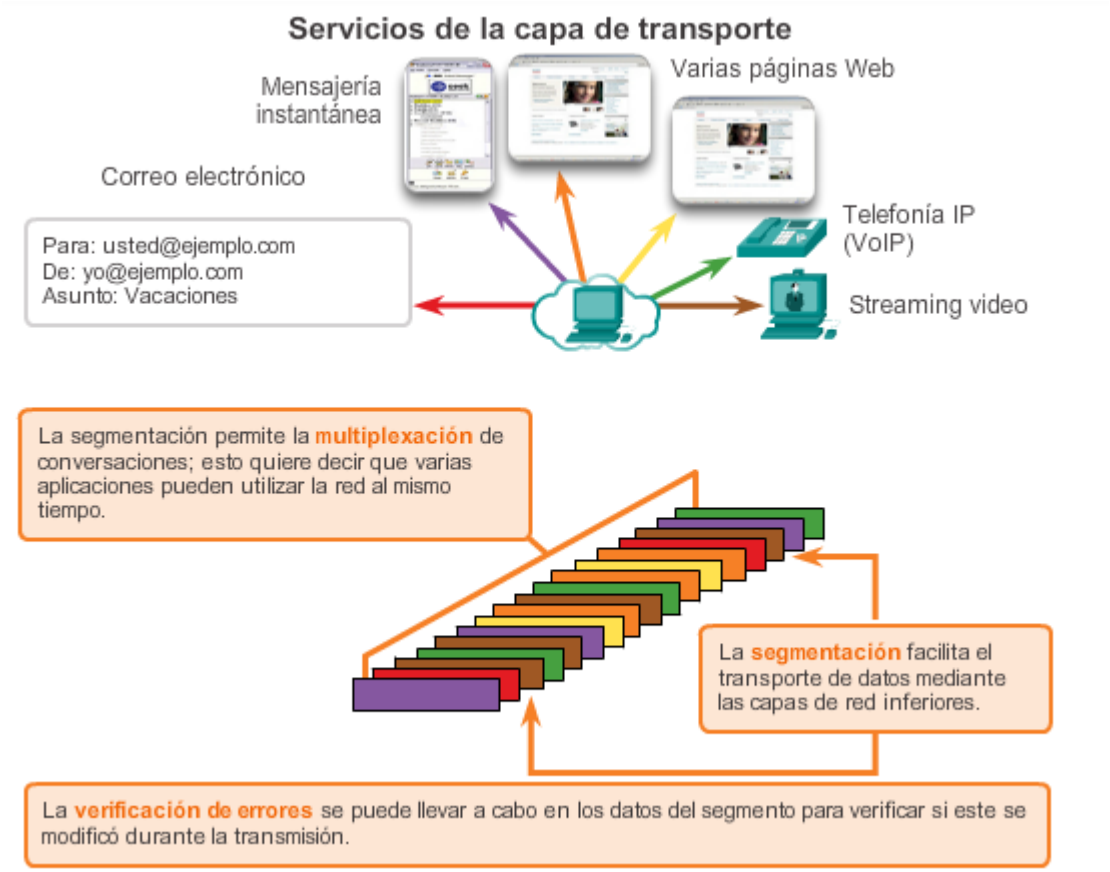
Segmentación de datos y rearmado de segmentos

Se deben preparar los datos para el envío a través de los medios en partes manejables. La mayoría de las redes tienen un límite de la cantidad de datos que se puede incluir en un solo paquete. Los protocolos de la capa de transporte tienen servicios que segmentan los datos de aplicación en bloques de datos de un tamaño apropiado (figura 2). Estos servicios incluyen la encapsulación necesaria en cada porción de datos. Se agrega un encabezado a cada bloque de datos para el rearmado. Este encabezado se utiliza para hacer un seguimiento del stream de datos.

En el destino, la capa de transporte debe poder reconstruir las porciones de datos en un stream de datos completo que sea útil para la capa de aplicación. Los protocolos en la capa de transporte describen cómo se utiliza la información del encabezado de dicha capa para rearmar las porciones de datos en streams para pasarlos a la capa de aplicación.

Identificación de aplicaciones

Puede haber muchas aplicaciones o servicios que se ejecutan en cada host de la red. Para pasar streams de datos a las aplicaciones adecuadas, la capa de transporte debe identificar la aplicación objetivo (figura 3). Para lograr esto, la capa de transporte asigna un identificador a cada aplicación. Este identificador se denomina “número de puerto”. A todos los procesos de software que requieran acceder a la red se les asigna un número de puerto exclusivo en ese host. La capa de transporte utiliza puertos para identificar la aplicación o el servicio.



T4. Ilustración 6

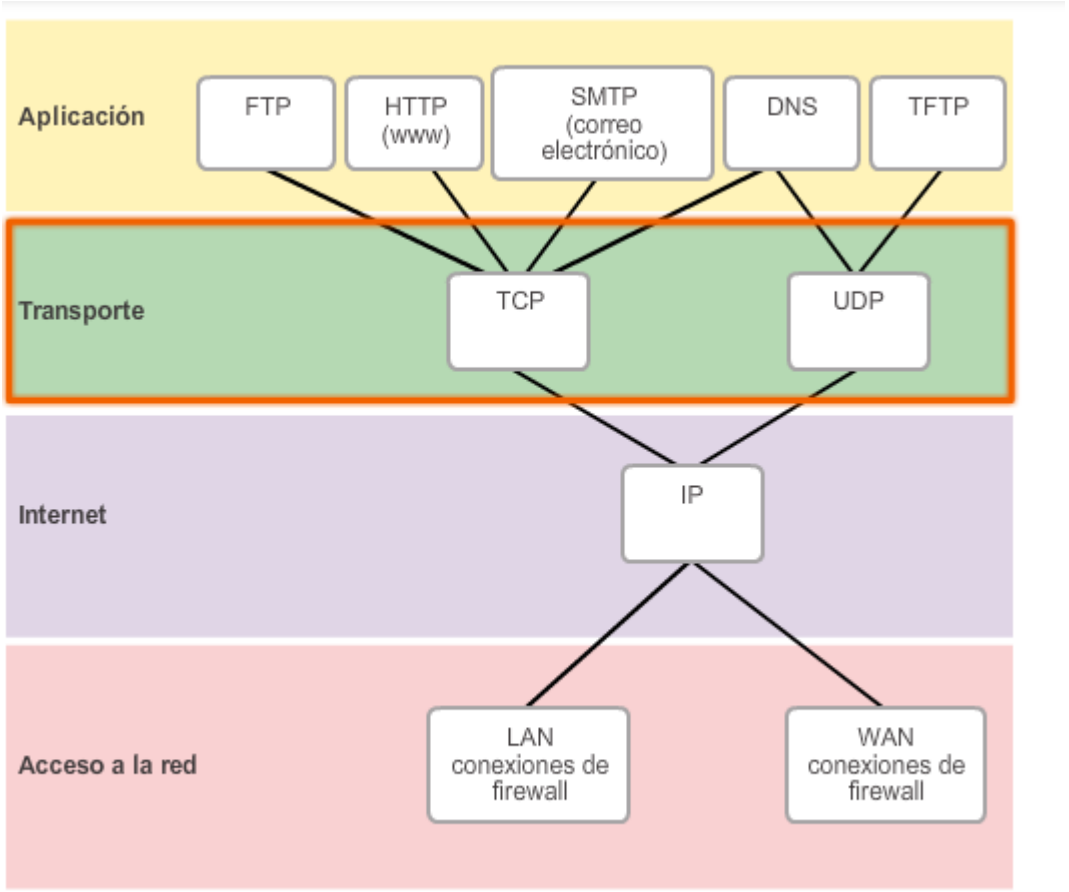
Multiplexación de conversaciones

El envío de algunos tipos de datos (por ejemplo, un streaming video) a través de una red, como un stream completo de comunicación, podría utilizar todo el ancho de banda disponible e impedir que se produzcan otras comunicaciones al mismo tiempo. También dificulta la recuperación de errores y la retransmisión de datos dañados.

En la ilustración, se muestra que la segmentación de los datos en partes más pequeñas permite que se entrelacen (multiplexen) varias comunicaciones de distintos usuarios en la misma red. La segmentación de los datos según los protocolos de la capa de transporte también proporciona los medios para enviar y recibir datos cuando se ejecutan varias aplicaciones a la vez en una PC.

Sin la segmentación, solo podría recibir datos una aplicación. Por ejemplo, con un streaming video, los medios se consumirían por completo por ese stream de comunicación en lugar de compartirse. No podría recibir correos electrónicos, chatear por mensajería instantánea o visitar páginas Web mientras mira el video.

Para identificar cada segmento de datos, la capa de transporte agrega al segmento un encabezado que contiene datos binarios. Este encabezado contiene campos de bits. Los valores de estos campos permiten que los distintos protocolos de la capa de transporte lleven a cabo diferentes funciones de administración de la comunicación de datos.

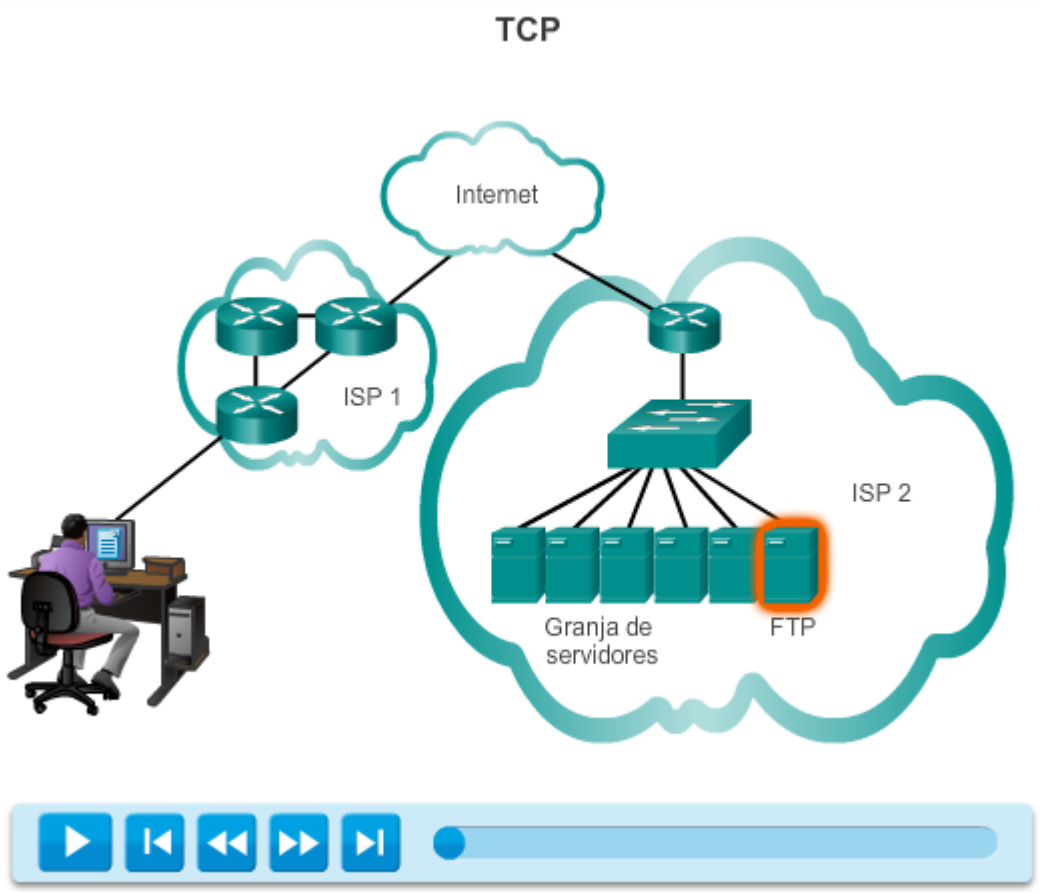


T4. Ilustración 7

La capa de transporte también es responsable de administrar los requisitos de confiabilidad de las conversaciones. Las diferentes aplicaciones tienen diferentes requisitos de confiabilidad de transporte.

IP se ocupa solo de la estructura, el direccionamiento y el enrutamiento de paquetes. IP no especifica la manera en que se lleva a cabo la entrega o el transporte de los paquetes. Los protocolos de transporte especifican la manera en que se transfieren los mensajes entre los hosts. TCP/IP proporciona dos protocolos de la capa de transporte: el protocolo de control de transmisión (TCP) y el protocolo de datagramas de usuario (UDP), como se muestra en la ilustración. IP utiliza estos protocolos de transporte para habilitar la comunicación y la transferencia de datos entre los hosts.

TCP se considera un protocolo de la capa de transporte confiable y completo, lo que garantiza que todos los datos lleguen al destino. En cambio, UDP es un protocolo de la capa de transporte muy simple que no proporciona confiabilidad.



T4. Ilustración 8. **Pincha en la imagen para ver la animación.**

Como se indicó anteriormente, TCP se considera un protocolo de transporte confiable, lo que significa que incluye procesos para garantizar la entrega confiable entre aplicaciones mediante el uso de entrega con acuse de recibo. La función del protocolo de transporte TCP es similar al envío de paquetes de los que se hace un seguimiento de origen a destino. Si se divide un pedido de FedEx en varios envíos, el cliente puede revisar en línea el orden de la entrega.

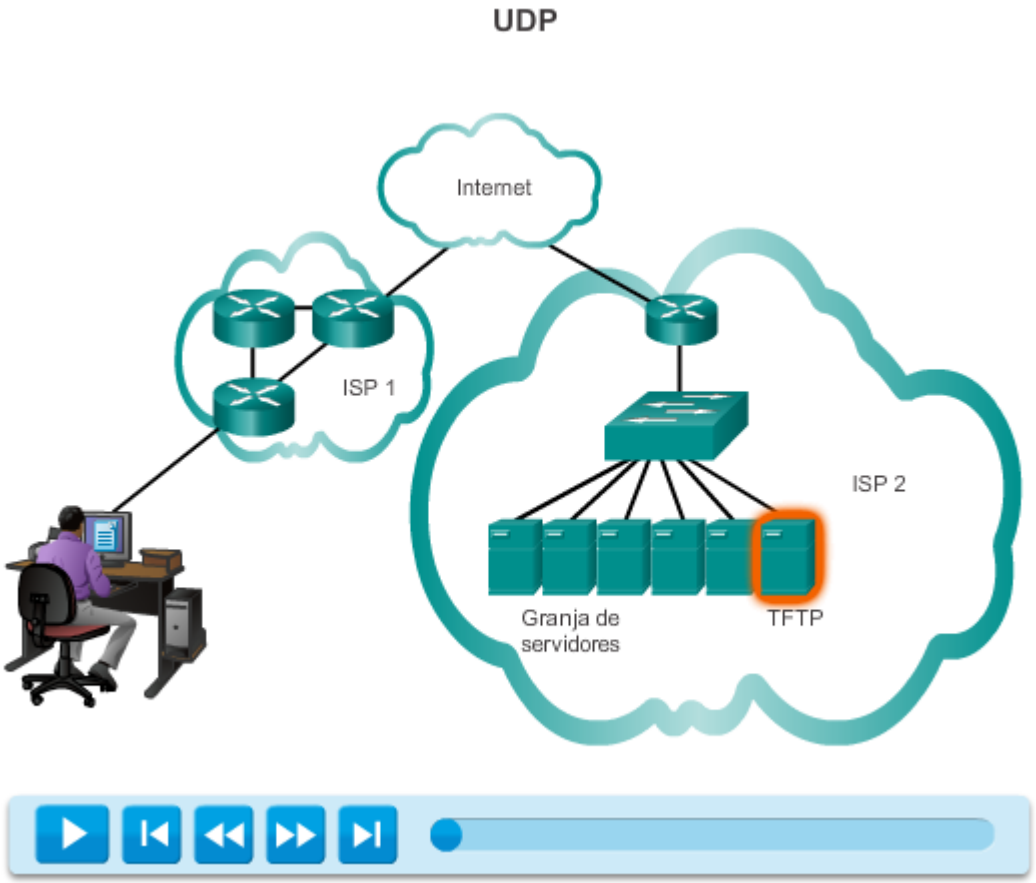
Con TCP, las tres operaciones básicas de confiabilidad son las siguientes:

- Seguimiento de segmentos de datos transmitidos
- Acuse de recibo de datos
- Retransmisión de cualquier dato sin acuse de recibo

TCP divide el mensaje en partes pequeñas, conocidas como segmentos. Los segmentos se numeran en secuencia y se pasan al proceso IP para armarse en paquetes. TCP realiza un seguimiento del número de segmentos que se enviaron a un host específico desde una aplicación específica. Si el emisor no recibe un acuse de recibo antes del transcurso de un período determinado, supone que los segmentos se perdieron y los vuelve a transmitir. Sólo se vuelve a enviar la parte del mensaje que se perdió, no todo el mensaje. En el host receptor, TCP se encarga de rearmar los segmentos del mensaje y de pasarlos a la aplicación. El protocolo de transferencia de archivos (FTP) y el protocolo de transferencia de hipertexto (HTTP) son ejemplos de las aplicaciones que utilizan TCP para garantizar la entrega de datos.

Haga clic en el botón Reproducir en la ilustración para ver una animación de los segmentos TCP que se transmiten del emisor al receptor.

Estos procesos de confiabilidad generan una sobrecarga adicional en los recursos de la red debido a los procesos de acuse de recibo, rastreo y retransmisión. Para admitir estos procesos de confiabilidad, se intercambian más datos de control entre los hosts emisores y receptores. Esta información de control está incluida en un encabezado TCP.



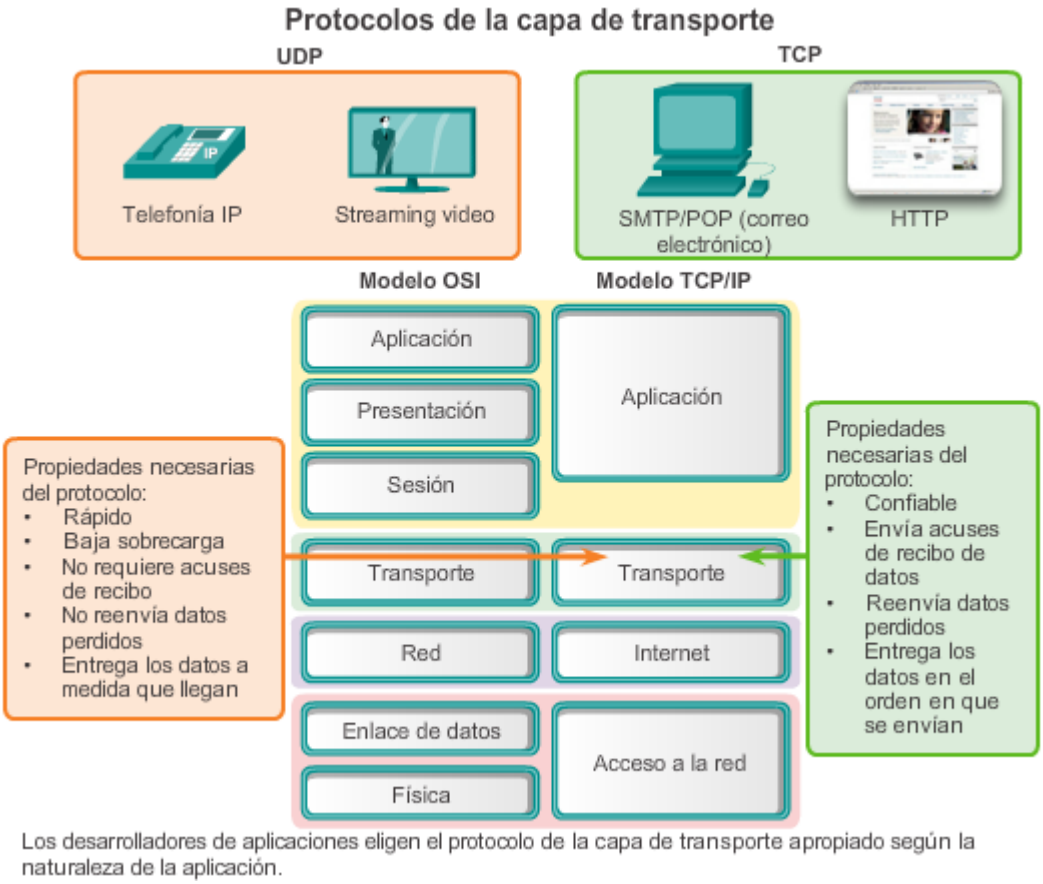
T4. Ilustración 9. Para ver la animación pincha en la imagen.

Si bien las funciones de confiabilidad de TCP proporcionan una comunicación más sólida entre aplicaciones, también representan una sobrecarga adicional y pueden provocar demoras en la transmisión. Existe una compensación entre el valor de la confiabilidad y la carga que implica para los recursos de la red. La imposición de sobrecarga para garantizar la confiabilidad para algunas aplicaciones podría reducir la utilidad a la aplicación e incluso ser perjudicial para esta. En estos casos, UDP es un protocolo de transporte mejor.

UDP proporciona solo las funciones básicas para entregar segmentos de datos entre las aplicaciones adecuadas, con muy poca sobrecarga y revisión de datos. El protocolo UDP se conoce como protocolo de entrega de máximo esfuerzo. En el contexto de redes, la entrega de máximo esfuerzo se denomina “poco confiable”, porque no hay acuse de recibo que indique que los datos se recibieron en el destino. Con UDP, no existen procesos de capa de transporte que informen al emisor si la entrega se produjo correctamente.

El proceso de UDP es similar al envío por correo de una carta simple sin registrar. El emisor de la carta no sabe si el receptor está disponible para recibir la carta ni la oficina de correos es responsable de hacer un seguimiento de la carta o de informar al emisor si esta no llega a destino.

Haga clic en el botón Reproducir en la ilustración para ver una animación de los segmentos UDP que se transmiten del emisor al receptor.



T4. Ilustración 10

Tanto TCP como UDP son protocolos de transporte válidos. Según los requisitos de la aplicación, se puede utilizar uno de estos protocolos de transporte y, en ocasiones, se pueden utilizar ambos. Los desarrolladores de aplicaciones deben elegir qué tipo de protocolo de transporte es adecuado según los requisitos de las aplicaciones.

Para algunas aplicaciones, los segmentos deben llegar en una secuencia muy específica para que se puedan procesar correctamente. Con otras aplicaciones, todos los datos se deben recibir en forma completa para poder considerarse útiles. En ambos casos, se utiliza TCP como protocolo de transporte. Por ejemplo, las aplicaciones, como las bases de datos, los exploradores Web y los clientes de correo electrónico, requieren que todos los datos que se envían lleguen a destino en su formato original. Todos los datos perdidos pueden corromper una comunicación y dejarla incompleta o ilegible. Por lo tanto, estas aplicaciones están diseñadas para utilizar TCP. Los gastos de red adicionales se consideran necesarios para estas aplicaciones.

En otros casos, una aplicación puede tolerar cierta pérdida de datos durante la transmisión a través de la red, pero no se admiten retrasos en la transmisión. UDP es la mejor opción para estas aplicaciones, ya que se requiere menos sobrecarga de red. Con aplicaciones como streaming audio, video y voz sobre IP (VoIP), es preferible utilizar UDP. Los acuses de recibo reducirían la velocidad de la entrega, y las retransmisiones no son recomendables.

Por ejemplo, si uno o dos segmentos de un stream de video no llegan al destino, se interrumpe momentáneamente el stream. Esto puede representar distorsión en la imagen, pero quizá ni el usuario lo note. Por otro lado, la imagen en un streaming video se degradaría en gran medida si el dispositivo de destino tuviera que dar cuenta de los datos perdidos y demorar el stream mientras

espera las retransmisiones. En este caso, es mejor producir el mejor video posible con los segmentos recibidos y prescindir de la confiabilidad.

La radio a través de Internet es otro ejemplo de aplicación que utiliza UDP. Si parte del mensaje se pierde durante su transmisión por la red, no se vuelve a transmitir. Si se pierden algunos paquetes, el oyente podrá escuchar una breve interrupción en el sonido. Si se utilizara TCP y se volvieran a enviar los paquetes perdidos, la transmisión haría una pausa para recibirlos, y la interrupción sería más notoria.

T4. Ilustración 11

La capa de transporte debe poder separar y administrar varias comunicaciones con diferentes necesidades de requisitos de transporte. Tome como ejemplo un usuario conectado a una red en un dispositivo final. El usuario envía y recibe correo electrónico y mensajes instantáneos, visita sitios Web y realiza una llamada telefónica de voz sobre IP (VoIP) simultáneamente. Cada una de estas aplicaciones envía y recibe datos a través de la red al mismo tiempo, a pesar de los diferentes requisitos de confiabilidad. Además, los datos de la llamada telefónica no están dirigidos al explorador Web y el texto de un mensaje instantáneo no aparece en un correo electrónico.

Por motivos de confiabilidad, los usuarios necesitan que un correo electrónico o una página Web se reciba y presente por completo para que la información se considere útil. Por lo general, se permiten leves retrasos en la carga de correo electrónico o de páginas Web, siempre y cuando el producto final se muestre en su totalidad y de forma correcta. En este ejemplo, la red administra el reenvío o reemplazo de la información que falta y no muestra el producto final hasta que se hayan recibido y armado todos los datos.

En cambio, la pérdida ocasional de partes pequeñas de una conversación telefónica se puede considerar aceptable. Incluso si se descartan partes pequeñas de algunas palabras, se puede deducir el audio que falta del contexto de la conversación o solicitar que la otra persona repita lo que dijo. Si la red administrara y reenviara segmentos faltantes, se prefiere lo mencionado anteriormente a los retrasos que se producen. En este ejemplo, es el usuario y no la red quien administra el reenvío o reemplazo de la información que falta.

Como se muestra en la ilustración, para que TCP y UDP administren estas conversaciones simultáneas con diversos requisitos, los servicios basados en UDP y TCP deben hacer un seguimiento de las diversas aplicaciones que se comunican. Para diferenciar los segmentos y datagramas para cada aplicación, tanto TCP como UDP cuentan con campos de encabezado que pueden identificar de manera exclusiva estas aplicaciones. Estos identificadores únicos son números de puertos.

T4. Ilustración 12

En el encabezado de cada segmento o datagrama, hay un puerto origen y uno de destino. El número de puerto de origen es el número para esta comunicación asociado con la aplicación que origina la comunicación en el host local. Como se muestra en la ilustración, el número de puerto de destino es el número para esta comunicación relacionada con la aplicación de destino en el host remoto.

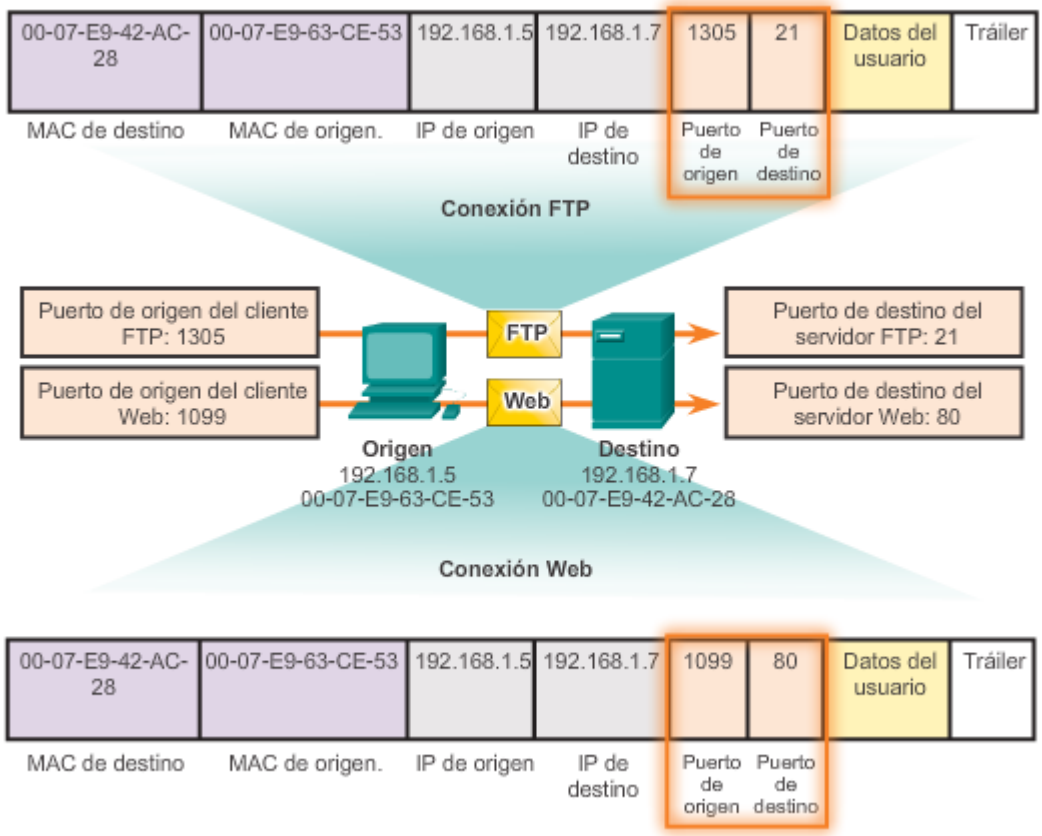
Cuando se envía un mensaje utilizando TCP o UDP, los protocolos y servicios solicitados se identifican con un número de puerto. Un puerto es un identificador numérico de cada segmento, que se utiliza para realizar un seguimiento de conversaciones específicas y de servicios de destino solicitados. Cada mensaje que envía un host contiene un puerto de origen y un puerto de destino.

Puerto de destino

El cliente coloca un número de puerto de destino en el segmento para informar al servidor de destino el servicio solicitado. Por ejemplo: el puerto 80 se refiere a HTTP o al servicio Web. Cuando un cliente especifica el puerto 80 en el puerto de destino, el servidor que recibe el mensaje sabe que se solicitan servicios Web. Un servidor puede ofrecer más de un servicio simultáneamente. Por ejemplo, puede ofrecer servicios Web en el puerto 80 al mismo tiempo que ofrece el establecimiento de una conexión FTP en el puerto 21.

Puerto de origen

El número de puerto de origen es generado de manera aleatoria por el dispositivo emisor para identificar una conversación entre dos dispositivos. Esto permite establecer varias conversaciones simultáneamente. En otras palabras, un dispositivo puede enviar varias solicitudes de servicio HTTP a un servidor Web al mismo tiempo. El seguimiento de las conversaciones por separado se basa en los puertos de origen.



T4. Ilustración 13

Los puertos de origen y de destino se colocan dentro del segmento. Los segmentos se encapsulan dentro de un paquete IP. El paquete IP contiene la dirección IP de origen y de destino. La combinación de las direcciones IP de origen y de destino y de los números 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. Miles de hosts se comunican a diario con millones de servidores diferentes. Los sockets identifican esas comunicaciones.

La combinación del número de puerto de la capa de transporte y de la dirección IP de la capa de red del host identifica de manera exclusiva un proceso de aplicación en particular que se ejecuta en un dispositivo host individual. Esta combinación se denomina socket. Un par de sockets, que consiste en las direcciones IP de origen y destino y los números de puertos, también es exclusivo e identifica la conversación específica entre los dos hosts.

Un socket de cliente puede ser parecido a esto, donde 1099 representa el número de puerto de origen: 192.168.1.5:1099

El socket en un servidor Web podría ser el siguiente: 192.168.1.7:80

Juntos, estos dos sockets se combinan para formar un par de sockets: 192.168.1.5:1099, 192.168.1.7:80

Con la creación de sockets, se conocen los extremos de la comunicación, de modo que los datos puedan moverse desde una aplicación en un host hacia una aplicación en otro host. Los sockets permiten que los procesos múltiples que se ejecutan en un cliente se distingan entre sí. También permiten la diferenciación de múltiples conexiones a un proceso de servidor.

El puerto de origen de la solicitud de un cliente se genera de manera aleatoria. El número de puerto actúa como dirección de retorno para la aplicación que realiza la solicitud. La capa de transporte hace un seguimiento de este puerto y de la aplicación que generó la solicitud de manera que cuando se devuelva una respuesta, esta se envíe a la aplicación correcta. El número de puerto de la aplicación que realiza la solicitud se utiliza como número de puerto de destino en la respuesta que vuelve del servidor.

La Agencia de asignación de números por Internet (IANA) asigna números de puerto. IANA es un organismo normativo responsable de asegurar diferentes estándares de direccionamiento.

Existen diferentes tipos de números de puerto, como se muestra en la figura 1:

- **Puertos bien conocidos (números del 0 al 1023):** estos números se reservan para servicios y aplicaciones. Se utilizan comúnmente para aplicaciones como HTTP (servidor Web), protocolo de acceso a mensajes de Internet (IMAP) o protocolo simple de transferencia de correo (SMTP) (servidor de correo electrónico) y Telnet. Al definir estos puertos bien conocidos para las aplicaciones de los servidores, las aplicaciones cliente se pueden programar para solicitar una conexión a ese puerto en particular y el servicio relacionado.

- **Puertos registrados (números del 1024 al 49151):** estos números de puerto se asignan a procesos o aplicaciones del usuario. Principalmente, estos procesos son aplicaciones individuales que el usuario elige instalar en lugar de aplicaciones comunes que recibiría un número de puerto bien conocido. Cuando no se utilizan para un recurso del servidor, un cliente puede seleccionar estos puertos de forma dinámica como su puerto de origen.

- **Puertos dinámicos o privados (números 49152 a 65535):** también conocidos como puertos efímeros, generalmente se los asigna de forma dinámica a las aplicaciones cliente cuando el cliente inicia una conexión a un servicio. El puerto dinámico suele utilizarse para identificar la aplicación cliente durante la comunicación, mientras que el cliente utiliza el puerto bien conocido para identificar el servicio que se solicita en el servidor y conectarse a dicho servicio. No es común que un cliente se conecte a un servicio mediante un puerto dinámico o privado (aunque algunos programas de intercambio de archivos punto a punto lo hacen).

En la figura 2, se muestran algunos puertos bien conocidos y registrados comunes en TCP. En la figura 3, se muestran algunos puertos bien conocidos y registrados comunes en UDP.

Uso de TCP y UDP

Algunas aplicaciones pueden utilizar tanto TCP como UDP (figura 4). Por ejemplo, el bajo gasto de UDP permite que DNS atienda rápidamente varias solicitudes de clientes. Sin embargo, a veces el envío de la información solicitada puede requerir la confiabilidad de TCP. En este caso, el número de puerto bien conocido (53) lo utilizan ambos protocolos con este servicio.

Hay una lista de números de puerto y de aplicaciones asociadas en el sitio Web organizacional de la IANA.

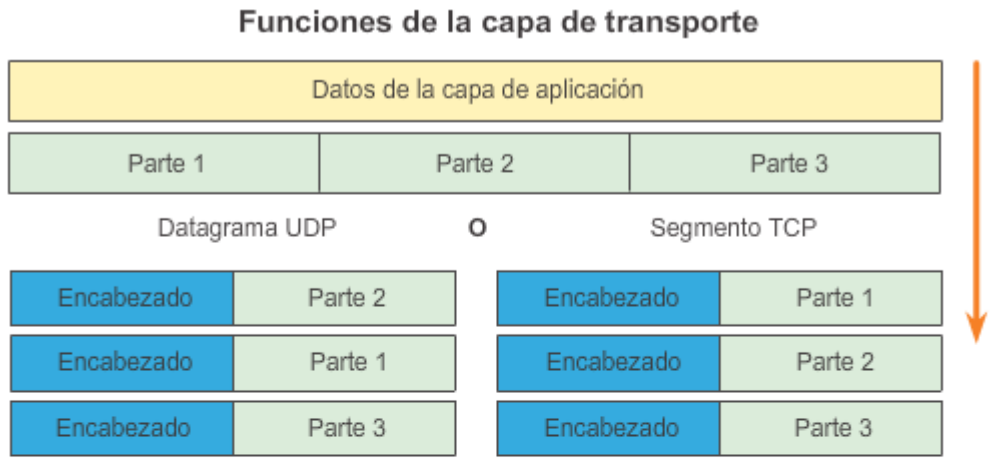
NOTA: para ver la información de las figuras, pincha aquí.

A veces es necesario conocer las conexiones TCP activas que están abiertas y en ejecución en el host de red. Netstat es una utilidad de red importante que puede usarse para verificar esas conexiones. Netstat indica el protocolo que se está usando, la dirección y el número de puerto locales, la dirección y el número de puerto externos y el estado de la conexión.

Las conexiones TCP desconocidas pueden presentar una amenaza de seguridad grave, ya que pueden indicar que hay algo o alguien conectado al host local. Además, las conexiones TCP innecesarias pueden consumir recursos valiosos del sistema y, por lo tanto, enlentecer el rendimiento del host. Netstat debe utilizarse para examinar las conexiones abiertas de un host cuando el rendimiento parece estar comprometido.

Existen muchas opciones útiles para el **comando**netstat. Haga clic en los botones en las figuras 1 a 5 para conocer la información que se muestra en los diferentes resultados del comando **netstat**.

NOTA: para ver la información de las figuras, pincha aquí.



T4. Ilustración 14

En un capítulo anterior, se explicó la forma en que se construyen las unidades de datos del protocolo (PDU) mediante la transmisión de datos de una aplicación a través de los diversos protocolos para crear una PDU que después se transmita en el medio. En el host de destino, este proceso se revierte hasta que los datos se puedan transferir a la aplicación.

Algunas aplicaciones transmiten grandes cantidades de datos; en algunos casos, muchos gigabytes. Resultaría poco práctico enviar todos estos datos en una sola gran sección. No puede transmitirse ningún otro tráfico de red mientras se envían estos datos. Una gran sección de datos puede tardar minutos y hasta horas en enviarse. Además, si hubiese errores, se perdería el archivo de datos completo o habría que volver a enviarlo. Los dispositivos de red no cuentan con buffers de memoria lo suficientemente grandes como para almacenar esa cantidad de datos durante la

transmisión o recepción. El límite varía según la tecnología de red y el medio físico específico en uso.

La división de datos de aplicación en segmentos asegura que estos se transmitan dentro de los límites de los medios y que los datos de diferentes aplicaciones se puedan multiplexar en los medios.

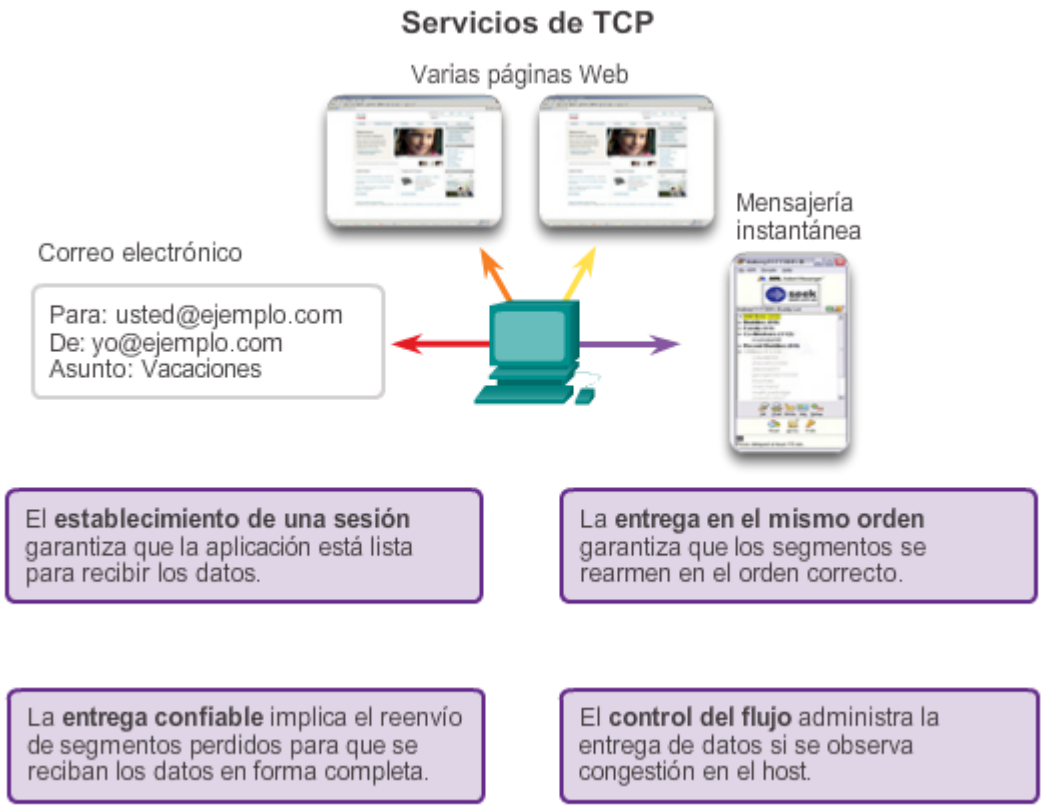
TCP y UDP: manejo distinto de la segmentación

Como se muestra en la ilustración, cada encabezado del segmento TCP contiene un número de secuencia que permite que las funciones de la capa de transporte en el host de destino vuelvan a armar segmentos en el orden en que se transmitieron. Esto asegura que la aplicación de destino tiene los datos en la misma forma que el emisor la planeó.

Aunque los servicios que utilizan UDP rastrean también las conversaciones entre las aplicaciones, no se encargan del orden en que se transmite la información ni de mantener una conexión. No existe número de secuencia en el encabezado UDP. UDP es un diseño simple y genera menos carga que TCP, lo que produce una transferencia de datos más rápida.

La información puede llegar en un orden distinto del de la transmisión, ya que los distintos paquetes pueden tomar diferentes rutas a través de la red. Una aplicación que utiliza UDP debe tolerar el hecho de que los datos no lleguen en el orden en el que fueron enviados.

2. Protocolo de control de transmisiones (TCP)



T4. Ilustración 15

Para entender con propiedad las diferencias entre TCP y UDP, es importante comprender la manera en que cada protocolo implementa las funciones específicas de confiabilidad y la forma en que realizan el seguimiento de las comunicaciones.

Protocolo de control de transmisión (TCP)

TCP se describió inicialmente en RFC 793. Además de admitir funciones básicas de segmentación y rearmado de datos, TCP, como se muestra en la ilustración, también proporciona lo siguiente:

- Conversaciones orientadas a la conexión mediante el establecimiento de sesiones
- Entrega confiable
- Reconstrucción de datos ordenada
- Control del flujo

Establecimiento de una sesión

TCP es un protocolo orientado a la conexión. Un protocolo orientado a la conexión es uno que negocia y establece una conexión (o sesión) permanente entre los dispositivos de origen y de destino antes de reenviar tráfico. El establecimiento de sesión prepara los dispositivos para que se comuniquen entre sí. Mediante el establecimiento de sesión, los dispositivos negocian la cantidad de tráfico que se puede reenviar en un momento determinado, y los datos que se comunican entre ambos se pueden administrar detenidamente. La sesión se termina solo cuando se completa toda la comunicación.

Entrega confiable

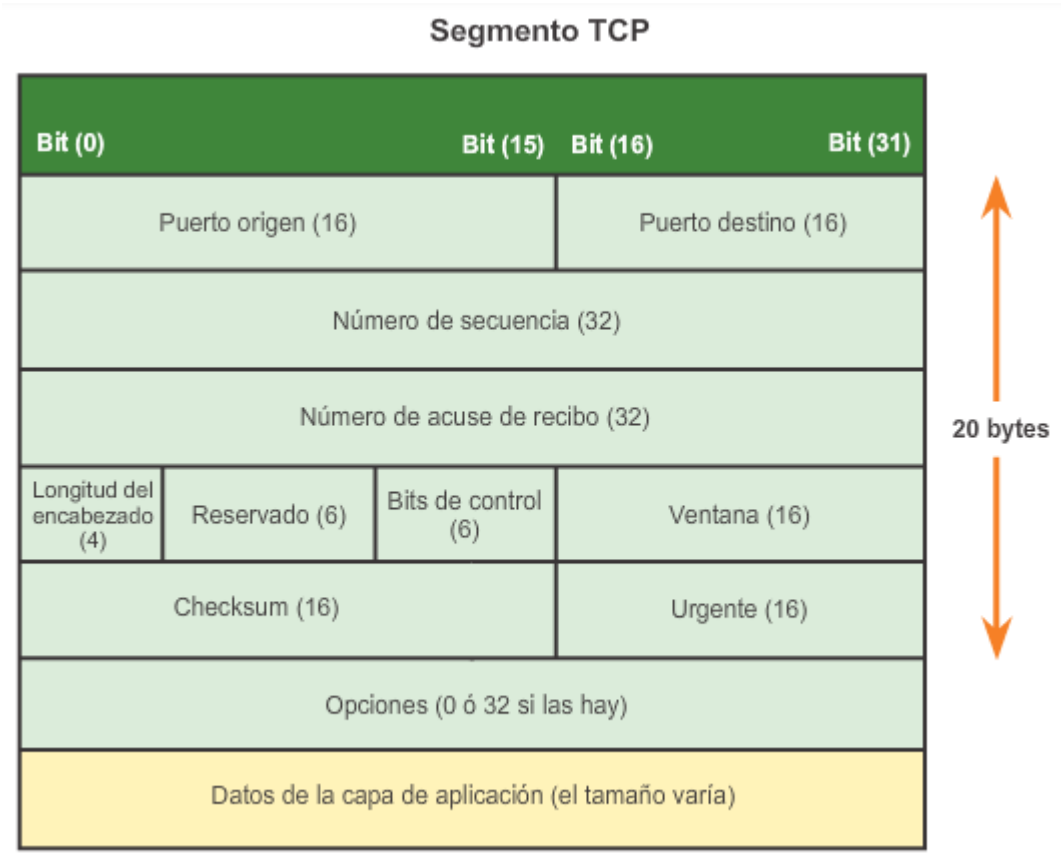
TCP puede implementar un método para garantizar la entrega confiable de los datos. En términos de redes, confiabilidad significa asegurar que cada sección de datos que envía el origen llegue al destino. Por varias razones, es posible que una sección de datos se corrompa o se pierda por completo a medida que se transmite a través de la red. TCP puede asegurar que todas las partes lleguen a destino al hacer que el dispositivo de origen retransmita los datos perdidos o dañados.

Entrega en el mismo orden

Los datos pueden llegar en el orden equivocado, debido a que las redes pueden proporcionar varias rutas que pueden tener diferentes velocidades de transmisión. Al numerar y secuenciar los segmentos, TCP puede asegurar que estos se rearmen en el orden correcto.

Control de flujo

Los hosts de la red cuentan con recursos limitados, como memoria o ancho de banda. Cuando TCP advierte que estos recursos están sobrecargados, puede solicitar que la aplicación emisora reduzca la velocidad del flujo de datos. Esto lo lleva a cabo TCP, que regula la cantidad de datos que transmite el origen. El control de flujo puede evitar la pérdida de segmentos en la red y evitar la necesitad de la retransmisión.



T4. Ilustración 16

Una vez que TCP establece una sesión, puede hacer un seguimiento de la conversación dentro de esa sesión. Debido a la capacidad de TCP de hacer un seguimiento de conversaciones reales, se lo considera un protocolo con estado. Un protocolo con estado es un protocolo que realiza el seguimiento del estado de la sesión de comunicación. Por ejemplo, cuando se transmiten datos mediante TCP, el emisor espera que el destino acuse recibo de los datos. TCP hace un seguimiento de la información que se envió y de la que se acusó de recibo. Si no se acusa recibo de los datos, el emisor supone que no llegaron y los vuelve a enviar. La sesión con estado comienza con el establecimiento de sesión y finaliza cuando se cierra la sesión con terminación de sesión.

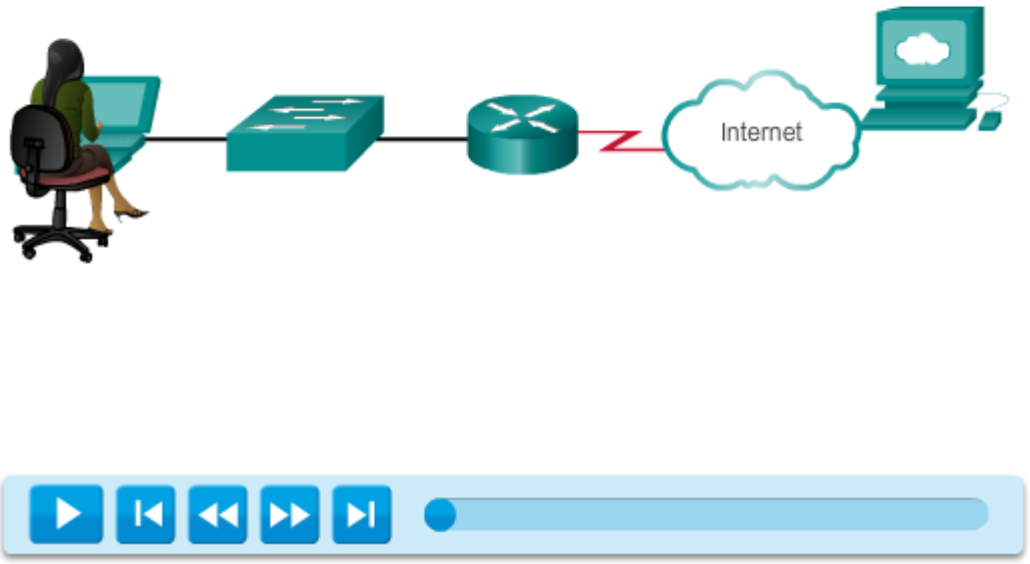
Nota: el mantenimiento de esta información de estado requiere recursos que no son necesarios para un protocolo sin estado, como UDP.

TCP genera sobrecarga adicional para obtener estas funciones. Como se muestra en la ilustración, cada segmento TCP tiene 20 bytes de sobrecarga en el encabezado que encapsula los datos de la capa de aplicación. Este tipo de segmento es mucho más largo que un segmento UDP, que solo tiene 8 bytes de sobrecarga. La sobrecarga adicional incluye lo siguiente:

- **Número de secuencia (32 bits):** se utiliza para rearmar datos.
- **Número de acuse de recibo (32 bits):** indica los datos que se recibieron.
- **Longitud del encabezado (4 bits):** conocido como “desplazamiento de datos”. Indica la longitud del encabezado del segmento TCP.
- **Reservado (6 bits):** este campo está reservado para el futuro.

- **Bits de control (6 bits):** incluye códigos de bit, o indicadores, que indican el propósito y la función del segmento TCP.
- **Tamaño de la ventana (16 bits):** indica la cantidad de segmentos que se puedan aceptar por vez.
- **Checksum (16 bits):** se utiliza para la verificación de errores en el encabezado y los datos del segmento.
- **Urgente (16 bits):** indica si la información es urgente.

Algunos ejemplos de aplicaciones que utilizan TCP son los exploradores Web, el correo electrónico y las transferencias de archivos.



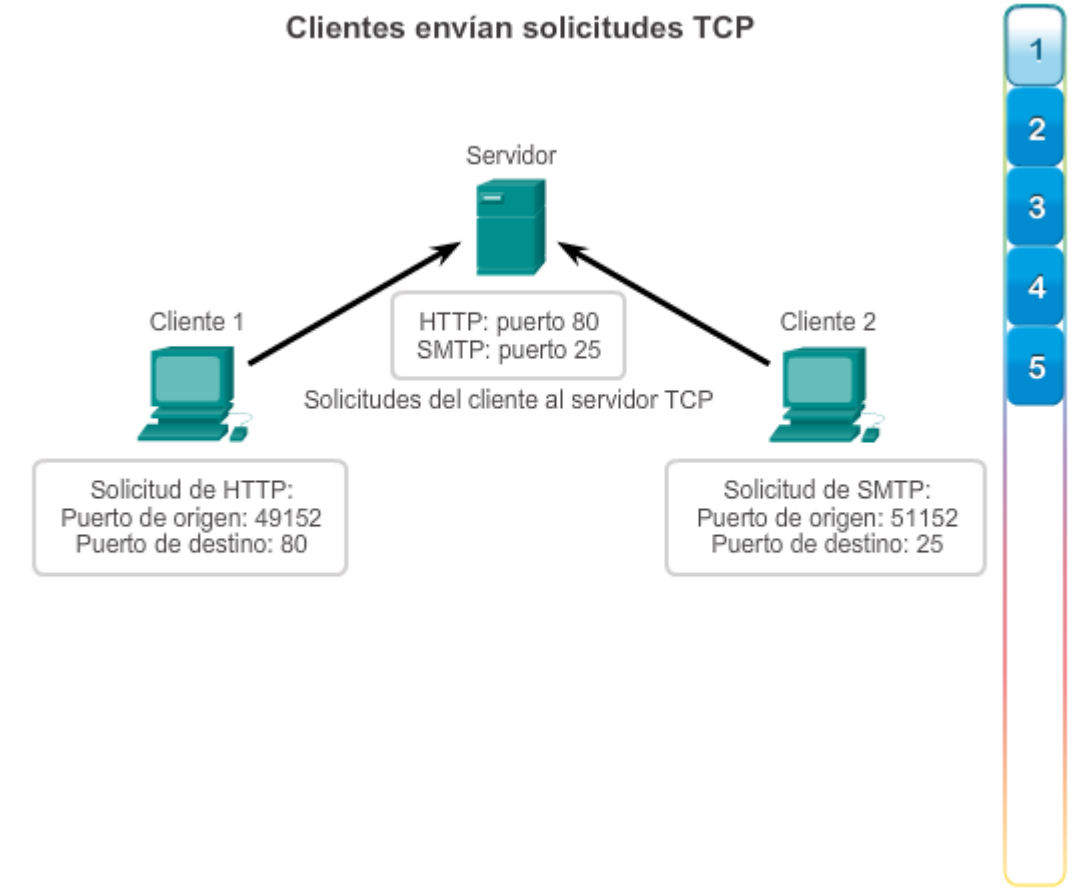
T4. Ilustración 17. **Para ver la animación, pincha en la imagen.**

La diferencia clave entre TCP y UDP es la confiabilidad. La confiabilidad de la comunicación TCP se obtiene con el uso de sesiones orientadas a la conexión. Antes de que un host que utiliza TCP envíe datos a otro host, TCP inicia un proceso para crear una conexión con el destino. Esta conexión con estado permite hacer un seguimiento de una sesión o un stream de comunicación entre los hosts. Este proceso asegura que cada host tenga conocimiento del stream de comunicación y se prepare para este. Una conversación TCP requiere que se establezca una sesión entre hosts en ambas direcciones, como se muestra en la animación.

Una vez que se establece una sesión y que comienza la transferencia de datos, el destino envía acuses de recibo al origen por los segmentos que recibe. Estos acuses de recibo forman la base de la confiabilidad dentro de la sesión TCP. Cuando el origen recibe un acuse de recibo, reconoce que los datos se entregaron correctamente y puede dejar de rastrearlos. Si el origen no recibe el acuse de recibo dentro de un tiempo predeterminado, retransmite esos datos al destino.

Parte de la carga adicional que genera el uso de TCP es el tráfico de red generado por los acuses de recibo y las retransmisiones. El establecimiento de las sesiones genera sobrecarga en forma de segmentos adicionales que se intercambian. Hay también sobrecarga en los hosts indivuduales

creada por la necesidad de mantener un registro de los segmentos que esperan un acuse de recibo y por el proceso de retransmisión.



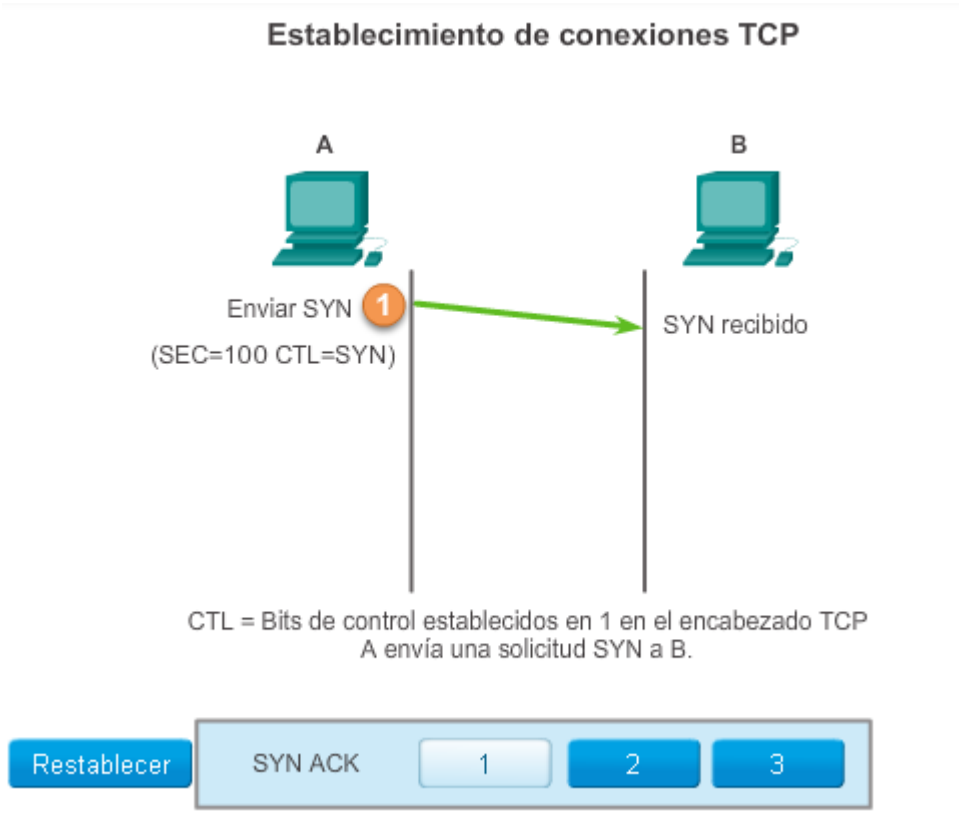
T4. Ilustración 18. **Para consultar las distintas figuras, pincha en la imagen.**

Los procesos de las aplicaciones se ejecutan en los servidores. Un único servidor puede ejecutar varios procesos de aplicaciones al mismo tiempo. Estos procesos esperan hasta que el cliente inicia comunicación con una solicitud de información u otros servicios.

Cada proceso de aplicación que se ejecuta en el servidor se configura para utilizar un número de puerto, ya sea predeterminado o de forma manual por el administrador del sistema. Un servidor individual no puede tener dos servicios asignados al mismo número de puerto dentro de los mismos servicios de la capa de transporte. Un host que ejecuta una aplicación de servidor Web y una de transferencia de archivos no puede configurar ambas para utilizar el mismo puerto (por ejemplo, el puerto TCP 8.080). Una aplicación de servidor activa asignada a un puerto específico se considera abierta, lo que significa que la capa de transporte acepta y procesa los segmentos dirigidos a ese puerto. Toda solicitud entrante de un cliente direccionada al socket correcto es aceptada y los datos se envían a la aplicación del servidor. Pueden existir varios puertos simultáneos abiertos en un servidor, uno para cada aplicación de servidor activa. Es común que un servidor proporcione más de un servicio al mismo tiempo, como un servidor Web y un servidor FTP.

Una manera de mejorar la seguridad en un servidor es restringir el acceso al servidor únicamente a aquellos puertos relacionados con los servicios y las aplicaciones a los que deben poder acceder los solicitantes autorizados.

Consulte las figuras 1 a 5 para ver la asignación típica de puertos de origen y de destino en las operaciones TCP de cliente y servidor.



T4. Ilustración 19. **Para consultar toda la animación, pincha en la imagen.**

En algunas culturas, cuando dos personas se conocen, generalmente se saludan dándose la mano. Ambas culturas entienden el acto de darse la mano como señal de un saludo amigable. Las conexiones en la red son similares. El primer enlace solicita la sincronización. El segundo enlace acusa recibo de la solicitud de sincronización inicial y sincroniza los parámetros de conexión en la dirección opuesta. El tercer segmento de enlace es un acuse de recibo que se utiliza para informarle al destino que ambos lados están de acuerdo en que se estableció una conexión.

Cuando dos hosts se comunican utilizando TCP, se establece una conexión antes de que puedan intercambiarse los datos. Luego de que se completa la comunicación, se cierran las sesiones y la conexión finaliza. Los mecanismos de conexión y sesión habilitan la función de confiabilidad de TCP. Vea en la figura los pasos para establecer y terminar una conexión del TCP.

Los hosts hacen un seguimiento de cada segmento de datos dentro de una sesión e intercambian información sobre qué datos se reciben mediante la información del encabezado TCP. TCP es un protocolo full-duplex, en el que cada conexión representa dos streams de comunicación unidireccionales, o sesiones. Para establecer la conexión los hosts realizan un protocolo de enlace de tres vías. Los bits de control en el encabezado TCP indican el progreso y estado de la conexión. Enlace de tres vías:

- Establece que el dispositivo de destino se presente en la red
- Verifica que el dispositivo de destino tenga un servicio activo y que acepte solicitudes en el número de puerto de destino que el cliente de origen intenta utilizar para la sesión
- Informa al dispositivo de destino que el cliente de origen intenta establecer una sesión de comunicación en dicho número de puerto

En las conexiones TCP, el cliente del host establece la conexión con el servidor. Los tres pasos en el establecimiento de una conexión TCP son:

Paso 1. El cliente de origen solicita una sesión de comunicación de cliente a servidor con el servidor.

Paso 2. El servidor acusa recibo de la sesión de comunicación de cliente a servidor y solicita una sesión de comunicación de servidor a cliente.

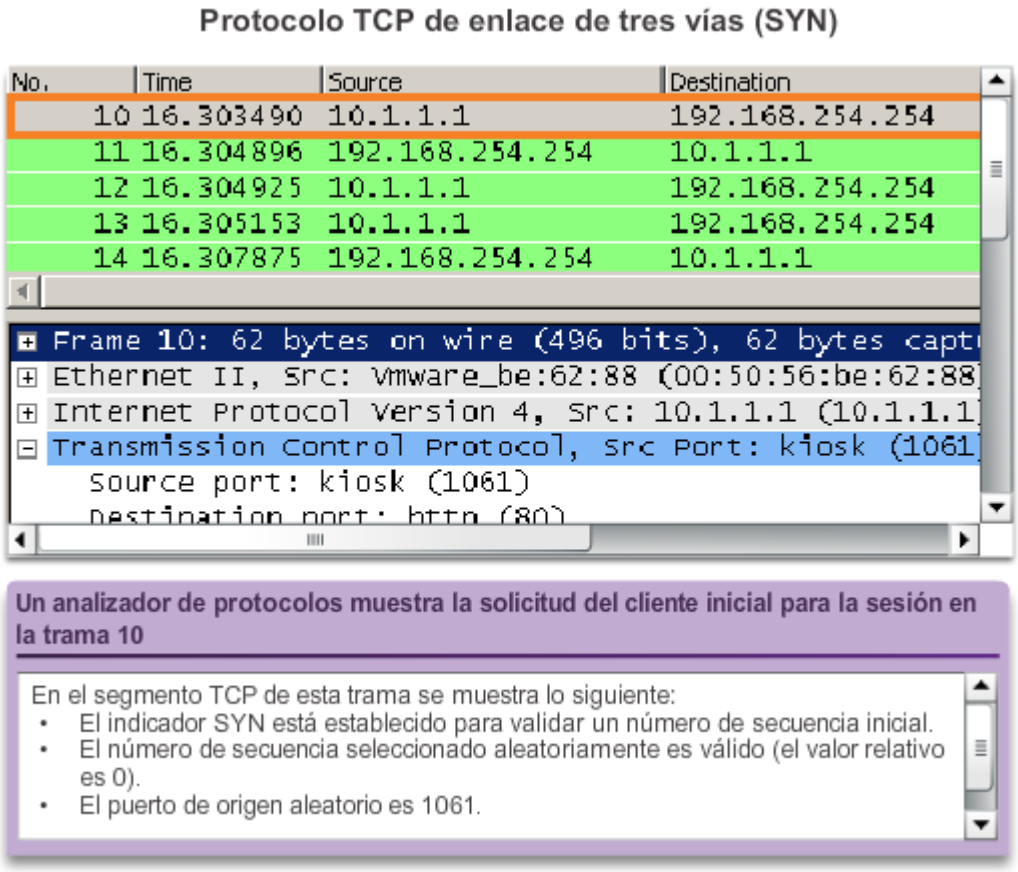
Paso 3. El cliente de origen acusa recibo de la sesión de comunicación de servidor a cliente.

En la ilustración, haga clic en los botones 1 a 3 para ver el establecimiento de la conexión TCP.

Para comprender el proceso de enlace de tres vías, observe los diversos valores que intercambian ambos hosts. Dentro del encabezado del segmento TCP, existen seis campos de 1 bit que contienen información de control utilizada para gestionar los procesos de TCP. Estos campos son los siguientes:

- **URG:** campo indicador urgente importante
- **ACK:** campo de acuse de recibo importante
- **PSH:** función de empuje
- **RST:** restablecer la conexión
- **SYN:** sincronizar números de secuencia
- **FIN:** no hay más datos del emisor

Los campos ACK y SYN son importantes para el análisis del protocolo de enlace de tres vías.



T4. Ilustración 20

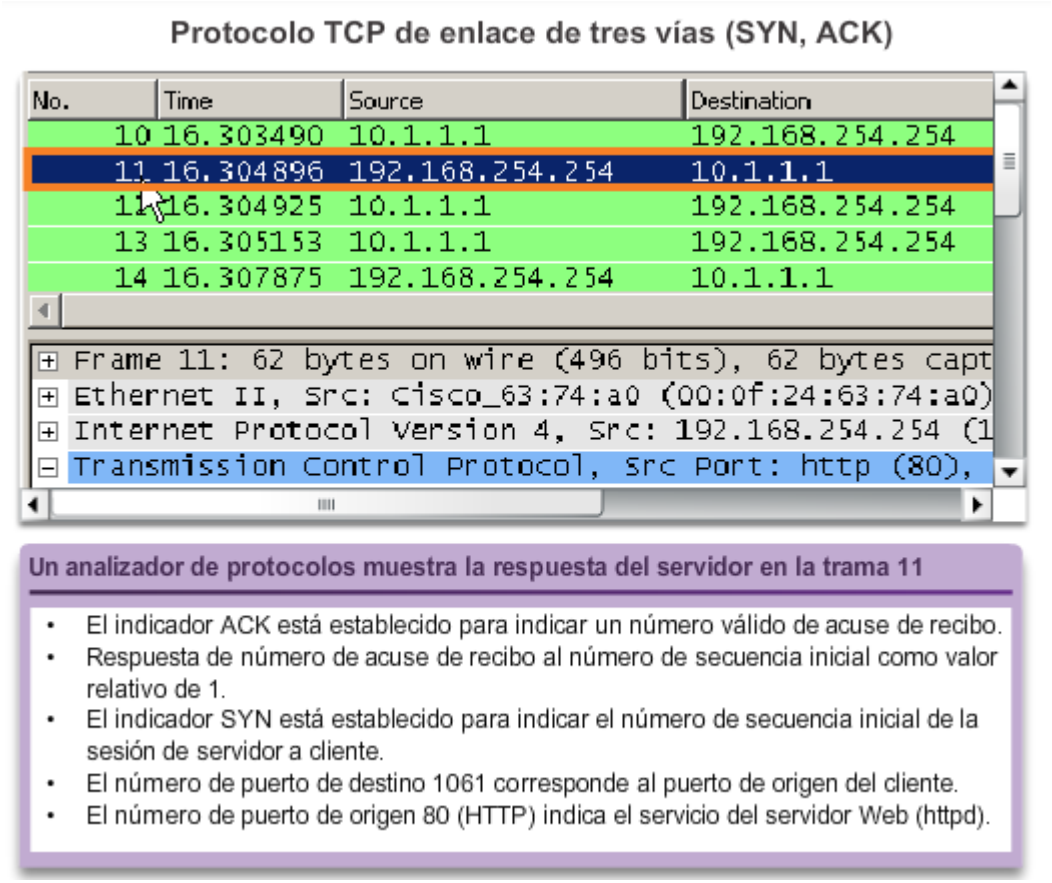
Mediante el resultado del software de análisis de protocolos, como los resultados de Wireshark, se puede examinar la operación del protocolo TCP de enlace de tres vías:

Paso 1: El cliente de origen solicita una sesión de comunicación de cliente a servidor con el servidor.

Un cliente TCP inicia un protocolo de enlace de tres vías al enviar un segmento con el indicador de control de sincronizar números de secuencia (SYN) establecido, lo que indica un valor inicial en el campo de número de secuencia en el encabezado. Este valor inicial para el número de secuencia, conocido como número de secuencia inicial (ISN), se elige de manera aleatoria y se utiliza para comenzar a rastrear el flujo de datos de esta sesión desde el cliente hasta el servidor. El ISN en el encabezado de cada segmento se incrementa en uno por cada byte de datos enviados desde el cliente hacia el servidor mientras continúa la conversación de datos.

Como se muestra en la figura, el resultado de un analizador de protocolos muestra el señalizador de control SYN y el número de secuencia relativa.

El indicador de control SYN está establecido y el número de secuencia relativa está en 0. Aunque el analizador de protocolos en el gráfico indique los valores relativos para los números de secuencia y de acuse de recibo, los verdaderos valores son números binarios de 32 bits. En la ilustración, se muestran los cuatro bytes representados en un valor hexadecimal.



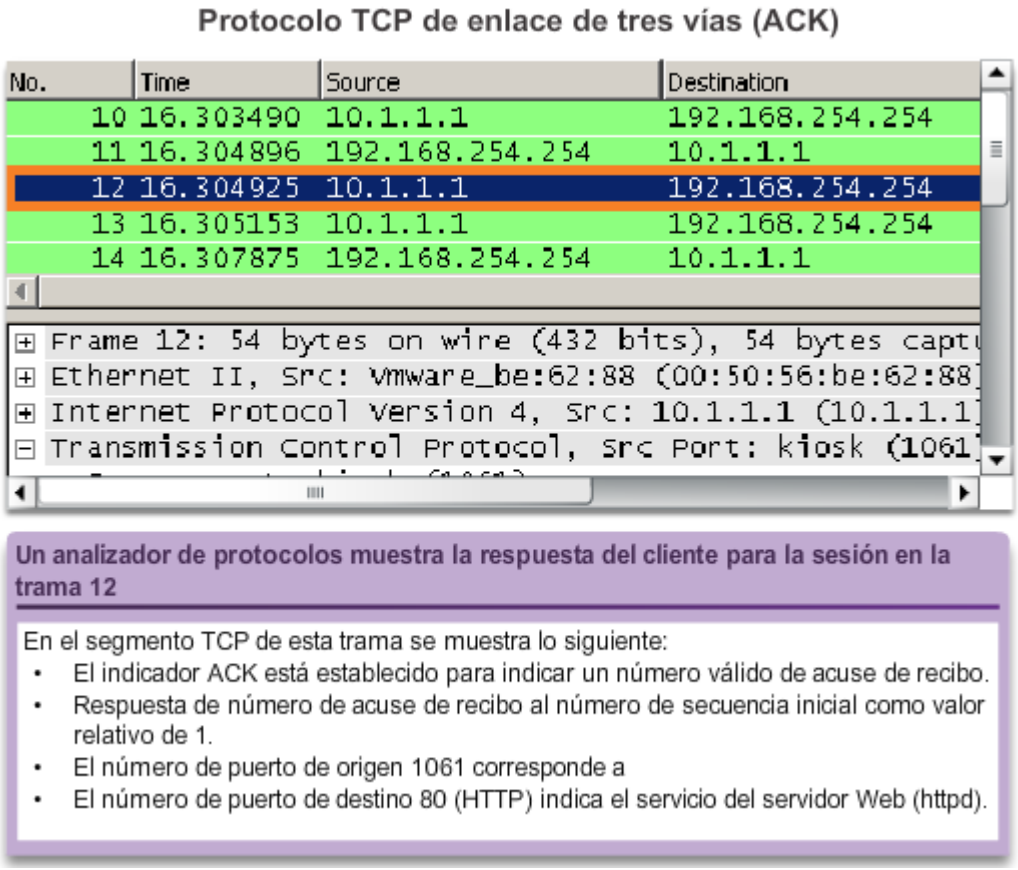
T4. Ilustración 21

Paso 2: El servidor reconoce la sesión de comunicación de cliente a servidor y solicita una sesión de comunicación de servidor a cliente.

El servidor TCP debe dar acuse de recibo del segmento SYN del cliente para establecer la sesión de cliente a servidor. Para hacerlo, el servidor envía un segmento al cliente con el indicador de acuse de recibo (ACK) establecido que indica que el número de acuse de recibo es significativo. Con este señalizador establecido en el segmento, el cliente interpreta esto como acuse de recibo de que el servidor ha recibido el SYN del cliente TCP.

El valor del campo de número de acuse de recibo es igual al ISN más 1. Esto establece una sesión del cliente al servidor. El indicador ACK permanece establecido para mantener el equilibrio de la sesión. Recuerde que la conversación entre el cliente y el servidor son, en realidad, dos sesiones unidireccionales: una del cliente al servidor y otra del servidor al cliente. En este segundo paso del protocolo de enlace de tres vías, el servidor debe iniciar la respuesta al cliente. Para comenzar esta sesión, el servidor utiliza el señalizador SYN de la misma manera en que lo hizo el cliente. Establece el señalizador de control SYN en el encabezado para establecer una sesión del servidor al cliente. El señalizador SYN indica que el valor inicial del campo de número de secuencia se encuentra en el encabezado. Este valor se utiliza para hacer un seguimiento del flujo de datos en esta sesión del servidor al cliente.

Como se muestra en la ilustración, el resultado del analizador de protocolos muestra que se establecieron los indicadores de control ACK y SYN y que se muestran los números de acuse de recibo y de secuencia relativa.



T4. Ilustración 22

Paso 3: El cliente de origen reconoce la sesión de comunicación de servidor a cliente.

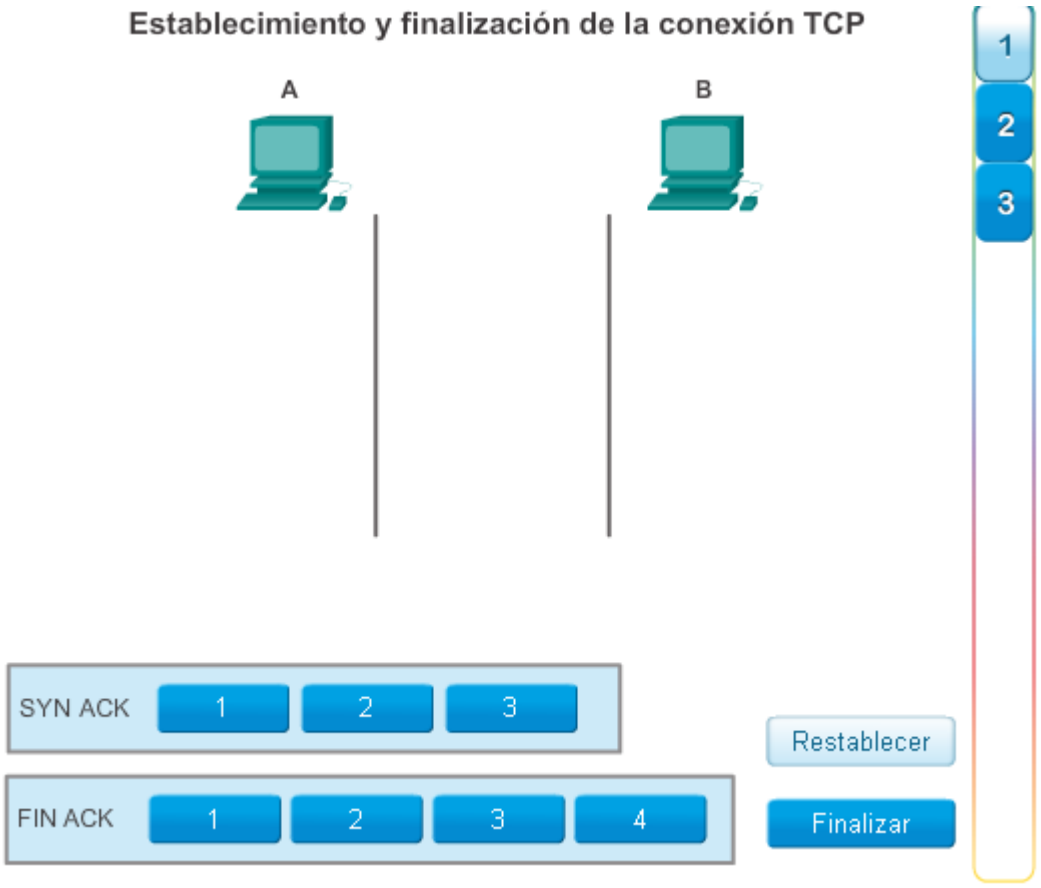
Por último, el cliente TCP responde con un segmento que contiene un ACK que actúa como respuesta al SYN de TCP enviado por el servidor. No existen datos de usuario en este segmento. El valor del campo de número de acuse de recibo contiene uno más que el ISN recibido del servidor. Una vez que se establecen ambas sesiones entre el cliente y el servidor, todos los segmentos adicionales que se intercambian en esta comunicación tendrán establecido el indicador ACK.

Como se muestra en la ilustración, el resultado del analizador de protocolos muestra el indicador de control ACK establecido y los números de acuse de recibo y de secuencia relativa.

Se puede añadir seguridad a la red de datos de la siguiente manera:

- Denegar el establecimiento de sesiones del TCP
- Permitir sólo sesiones que se establezcan para servicios específicos
- Permitir sólo tráfico como parte de sesiones ya establecidas

Estas medidas de seguridad se pueden implementar para todas las sesiones TCP o solo para las sesiones seleccionadas.



T4. Ilustración 23. **Para consultar toda la información, pincha en la imagen.**

Para cerrar una conexión, se debe establecer el indicador de control finalizar (FIN) en el encabezado del segmento. Para finalizar todas las sesiones TCP de una vía, se utiliza un enlace de dos vías, que consta de un segmento FIN y un segmento ACK. Por lo tanto, para terminar una única conversación que admite TCP, se requieren cuatro intercambios para finalizar ambas sesiones, como se muestra en la figura 1.

Nota: en esta explicación, los términos “cliente” y “servidor” se utilizan como referencia con fines de simplificación, pero el proceso de finalización lo pueden iniciar dos hosts cualesquiera que tengan una sesión abierta:

Paso 1: cuando el cliente no tiene más datos para enviar en el stream, envía un segmento con el indicador FIN establecido.

Paso 2: el servidor envía un ACK para acusar recibo del FIN y terminar la sesión de cliente a servidor.

Paso 3: el servidor envía un FIN al cliente para terminar la sesión de servidor a cliente.

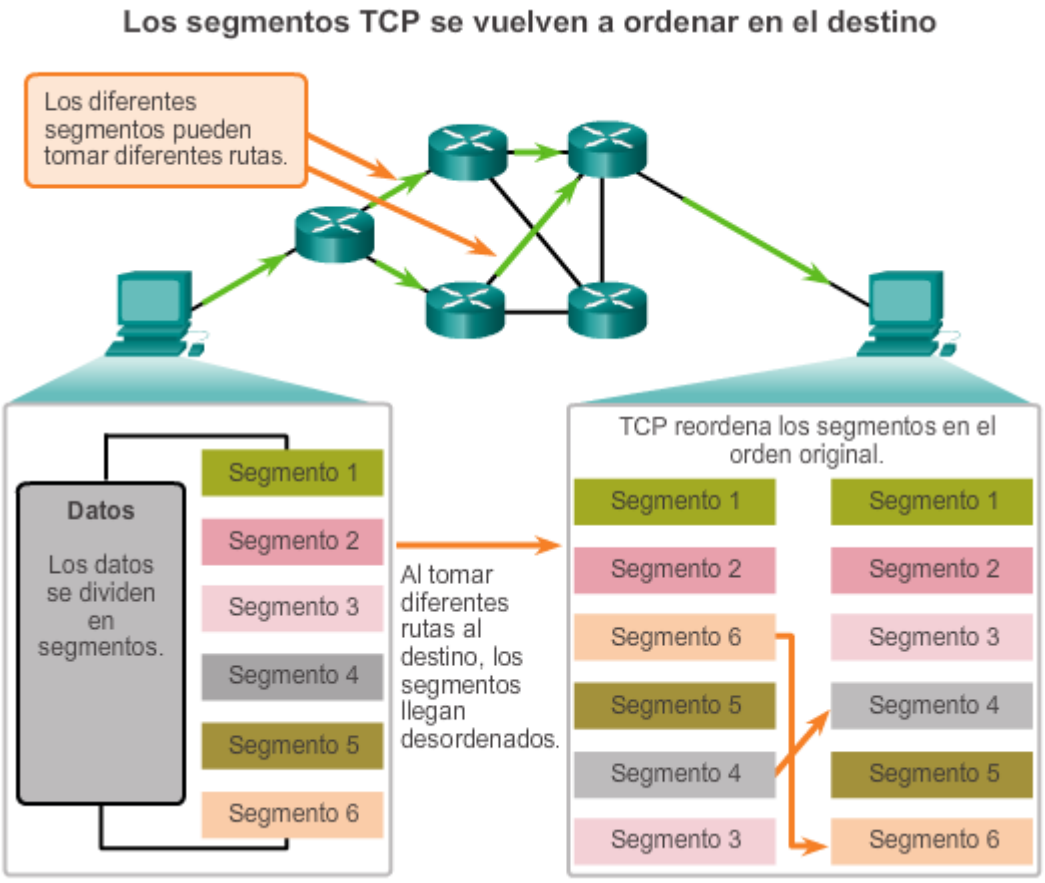
Paso 4: el cliente responde con un ACK para dar acuse de recibo del FIN desde el servidor.

Cuando el cliente no tiene más datos que transferir, establece el indicador FIN en el encabezado de un segmento. A continuación, el extremo servidor de la conexión envía un segmento normal que contiene datos con el indicador ACK establecido utilizando el número de acuse de recibo, lo que confirma que se recibieron todos los bytes de datos. Cuando se dio acuse de recibo de todos los segmentos, la sesión se cierra.

La sesión en la otra dirección se cierra con el mismo proceso. El receptor indica que no existen más datos para enviar estableciendo el señalizador FIN en el encabezado del segmento enviado al origen. Un acuse de recibo devuelto confirma que todos los bytes de datos se recibieron y que la sesión, a su vez, finalizó.

Consulte las figuras 2 y 3 para ver los indicadores de control FIN y ACK establecidos en el encabezado del segmento, lo que finaliza la sesión HTTP.

También es posible terminar la conexión por medio de un enlace de tres vías. Cuando el cliente no posee más datos para enviar, envía un señalizador FIN al servidor. Si el servidor tampoco tiene más datos para enviar, puede responder con los señalizadores FIN y ACK, combinando dos pasos en uno. A continuación, el cliente responde con un ACK.



T4. Ilustración 24

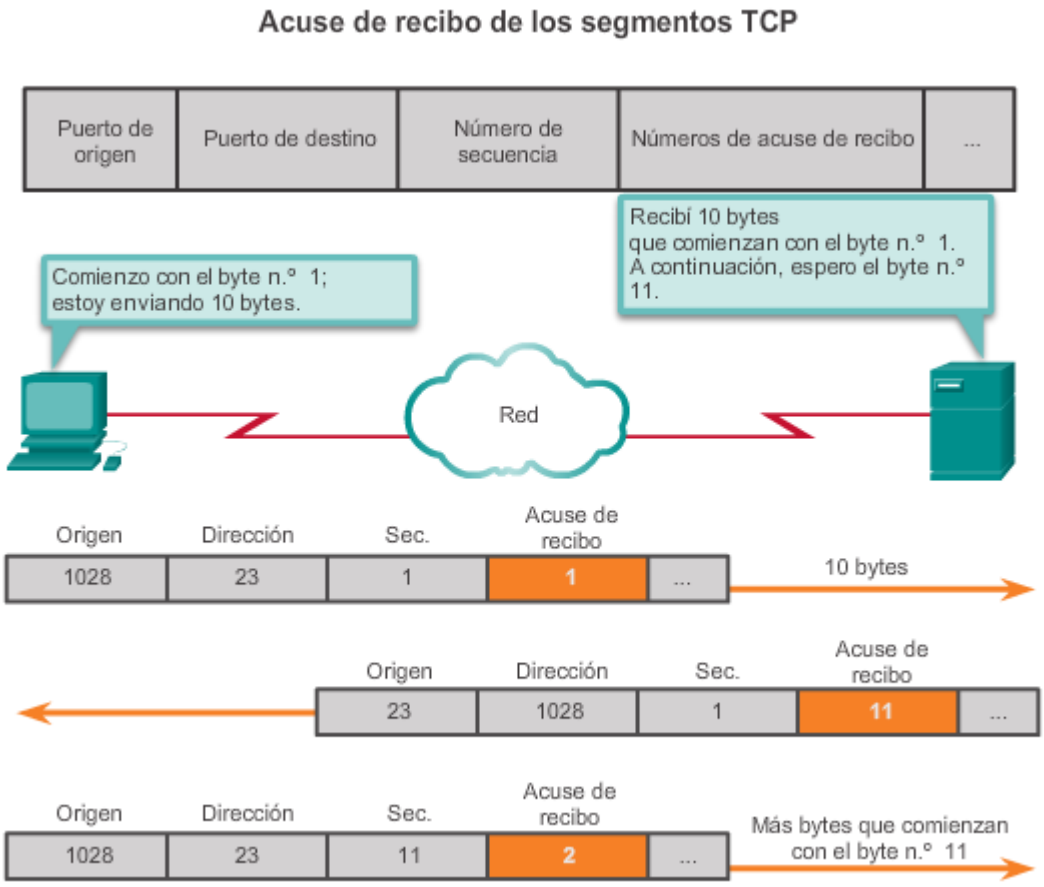
Reordenamiento de segmentos

Cuando los servicios envían datos mediante el TCP, los segmentos pueden llegar a su destino en desorden. Para que el receptor comprenda el mensaje original, los datos en estos segmentos se reensamblan en el orden original. Para lograr esto, se asignan números de secuencia en el encabezado de cada paquete.

Durante la configuración de la sesión, se establece un número de secuencia inicial (ISN). Este ISN representa el valor inicial para los bytes para esta sesión que se transmite a la aplicación receptora. A medida que se transmiten los datos durante la sesión, el número de secuencia se incrementa en el número de bytes que se han transmitido. Este seguimiento de bytes de datos permite identificar y dar acuse de recibo de cada segmento de manera exclusiva. Se pueden identificar segmentos perdidos.

Los números de secuencia de segmento habilitan la confiabilidad al indicar cómo rearmar y reordenar los segmentos recibidos, como se muestra en la ilustración.

El proceso TCP receptor coloca los datos del segmento en un búfer de recepción. Los segmentos se colocan en el orden de número de secuencia correcto y se pasan a la capa de aplicación cuando se rearman. Todos los segmentos que llegan con números de secuencia no contiguos se mantienen para su posterior procesamiento. A continuación, cuando llegan los segmentos con bytes faltantes, tales segmentos se procesan en orden.



T4. Ilustración 25

Confirmación de recepción de segmentos

Una de las funciones de TCP es garantizar que cada segmento llegue a destino. Los servicios de TCP en el host de destino envían un acuse de recibo de los datos que recibe la aplicación de origen.

El número de secuencia (SEQ) y el número de acuse de recibo (ACK) se utilizan juntos para confirmar la recepción de los bytes de datos contenidos en los segmentos transmitidos. El número de SEQ indica la cantidad relativa de bytes que se transmitieron en esta sesión, incluso los bytes en el segmento actual. TCP utiliza el número de ACK reenviado al origen para indicar el próximo byte que el receptor espera recibir. Esto se llama acuse de recibo de expectativa.

Se le informa al origen que el destino recibió todos los bytes de este stream de datos, hasta el byte especificado por el número de ACK, pero sin incluirlo. Se espera que el host emisor envíe un segmento que utiliza un número de secuencia que es igual al número de ACK.

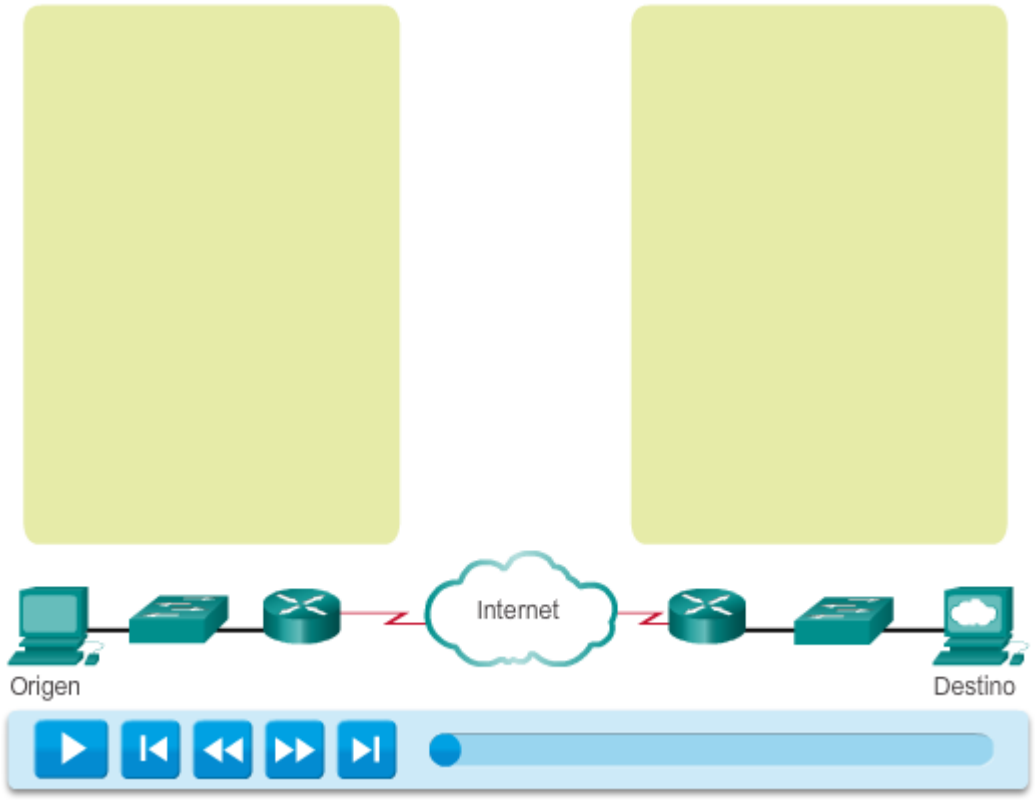
Recuerde que cada conexión son realmente dos sesiones de una vía. Los números de SEQ y ACK se intercambian en ambas direcciones.

En el ejemplo de la figura, el host de la izquierda envía datos al host de la derecha. Envía un segmento que contiene 10 bytes de datos para esta sesión y un número de secuencia igual a 1 en el encabezado.

El host receptor recibe el segmento en la capa 4 y determina que el número de secuencia es 1 y que tiene 10 bytes de datos. Luego el host envía un segmento de vuelta al host de la izquierda para acusar recibo de estos datos. En este segmento, el host establece el número de ACK en 11 para indicar que el siguiente byte de datos que espera recibir en esta sesión es el byte número 11. Cuando el host emisor recibe este acuse de recibo, puede enviar el próximo segmento que contiene datos para esta sesión a partir del byte 11.

En este ejemplo, si el host emisor tuviera que esperar el acuse de recibo de cada uno de los 10 bytes, la red tendría mucha sobrecarga. Para reducir la sobrecarga de estos acuses de recibo, pueden enviarse varios segmentos de datos y dar acuse de recibo de estos con un único mensaje de TCP en la dirección opuesta. Este acuse de recibo contiene un número de ACK que se basa en la cantidad total de bytes recibidos en la sesión. Por ejemplo, si se comienza con un número de secuencia 2000, si se reciben 10 segmentos de 1000 bytes cada uno, se devolverá al origen un número de ACK igual a 12 001.

La cantidad de datos que un origen puede transmitir antes de recibir un acuse de recibo se denomina “tamaño de la ventana”, que es un campo en el encabezado TCP que permite administrar datos perdidos y controlar el flujo.



T4. Ilustración 26. **Para ver la animación, pincha en la imagen.**

Manejo de segmentos perdidos

La pérdida de datos se produce en ocasiones, sin importar qué tan bien diseñada esté la red; por lo tanto, TCP proporciona métodos para administrar estas pérdidas de segmentos. Entre estos está un mecanismo para retransmitir segmentos con datos sin acuse de recibo.

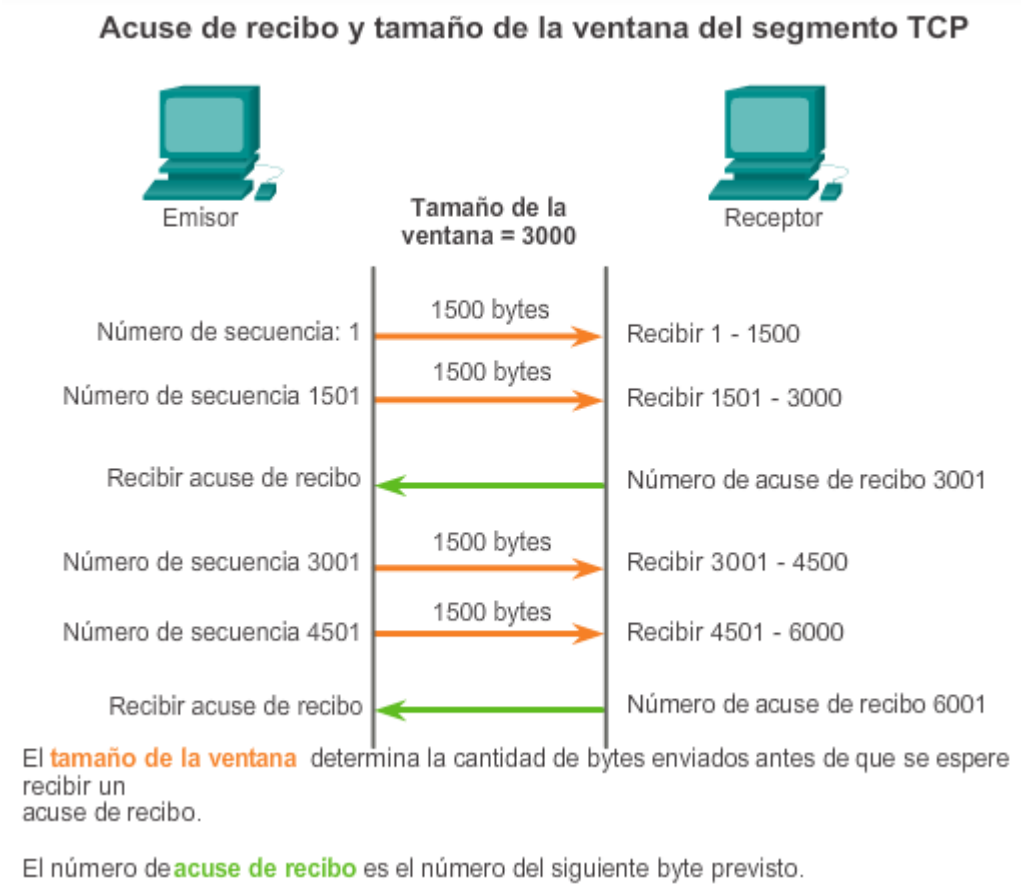
Un servicio de host de destino que utiliza TCP generalmente sólo da acuse de recibo de datos para bytes de secuencia continuos. Si faltan uno o más segmentos, solo se hace acuse de recibo de los datos en la primera secuencia contigua de bytes. Por ejemplo, si se reciben segmentos con números de secuencia de 1500 a 3000 y de 3400 a 3500, el número de ACK sería 3001. Esto se debe a que hay segmentos con números de SEQ de 3001 a 3399 que no se recibieron.

Cuando el TCP en el host de origen no recibe un acuse de recibo después de una cantidad de tiempo predeterminada, este vuelve al último número de ACK recibido y vuelve a transmitir los datos desde ese punto en adelante. La solicitud de comentarios (RFC) no especifica el proceso de retransmisión, pero se deja a criterio de la implementación particular del TCP.

Para una implementación de TCP típica, un host puede transmitir un segmento, colocar una copia del segmento en una cola de retransmisión e iniciar un temporizador. Cuando se recibe el acuse de recibo de los datos, se elimina el segmento de la cola. Si no se recibe el acuse de recibo antes de que el temporizador venza, el segmento es retransmitido.

Haga clic en el botón Reproducir en la ilustración para ver una animación de la retransmisión de segmentos perdidos.

En la actualidad, los hosts pueden emplear también una característica optativa llamada “acuses de recibo selectivos” (SACK). Si ambos hosts admiten los SACK, es posible que el destino acuse recibo de los bytes de segmentos discontinuos, y el host solo necesitará volver a transmitir los datos perdidos.



T4. Ilustración 27

Control de flujo

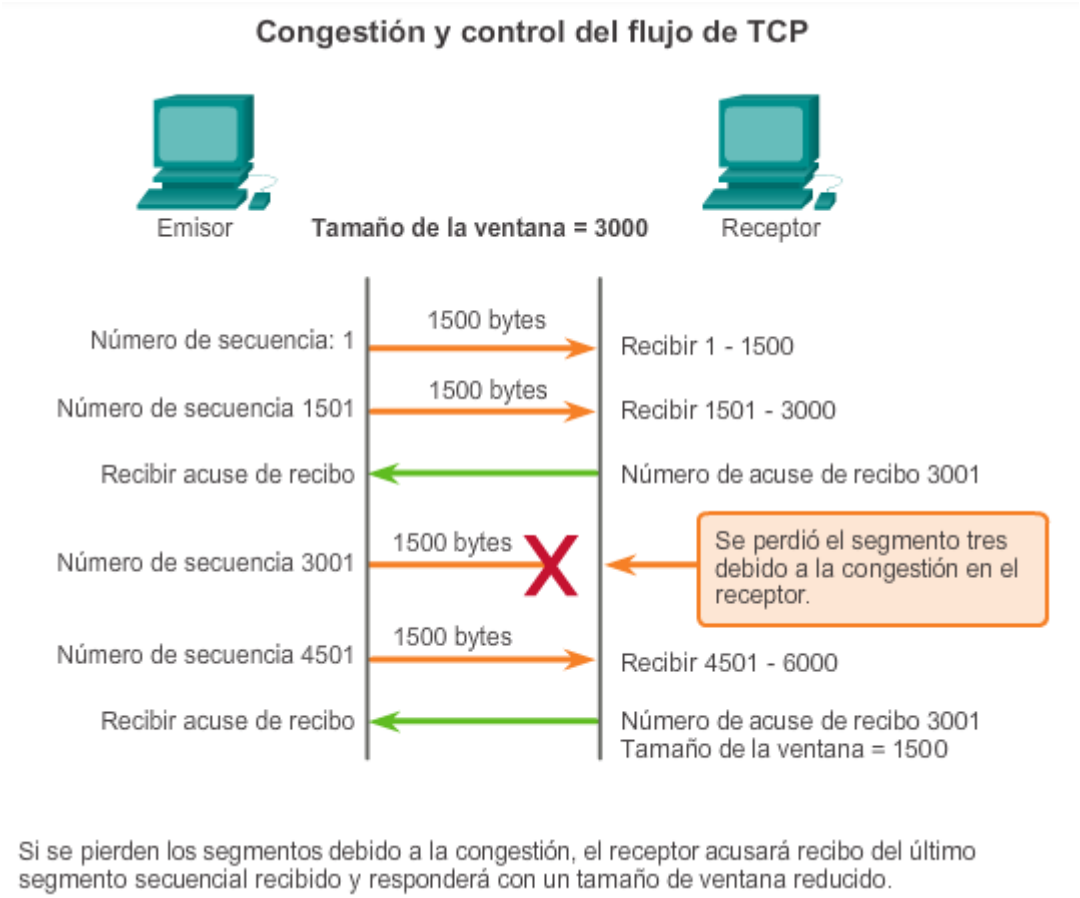
TCP también proporciona mecanismos para el control del flujo. El control del flujo permite mantener la confiabilidad de la transmisión de TCP mediante el ajuste de la velocidad del flujo de datos entre el origen y el destino para una sesión dada. El control del flujo se logra limitando la cantidad de segmentos de datos que se envían al mismo tiempo y solicitando acuses de recibo antes de enviar más segmentos.

Para lograr el control del flujo, lo primero que determina TCP es la cantidad de segmentos de datos que puede aceptar el dispositivo de destino. El encabezado TCP incluye un campo de 16 bits llamado “tamaño de la ventana”. Esta es la cantidad de bytes que el dispositivo de destino de una sesión TCP puede aceptar y procesar al mismo tiempo. El tamaño inicial de la ventana se acuerda durante el inicio de sesión entre el origen y el destino por medio del protocolo de enlace de tres vías. Una vez acordado el tamaño, el dispositivo de origen debe limitar la cantidad de segmentos de datos enviados al dispositivo de destino sobre la base del tamaño de la ventana. El dispositivo de origen puede continuar enviando más datos para la sesión solo cuando obtiene un acuse de recibo de los segmentos de datos recibidos.

Durante el retraso en la recepción del acuse de recibo, el emisor no envía ningún otro segmento. En los períodos en los que la red está congestionada o los recursos del host receptor están exigidos, la demora puede aumentar. A medida que aumenta esta demora, disminuye la tasa de transmisión efectiva de los datos para esta sesión. La disminución de velocidad en la transmisión de datos de cada sesión ayuda a reducir el conflicto de recursos en la red y en el dispositivo de destino cuando se ejecutan varias sesiones.

Ver la figura para obtener una representación simplificada del tamaño de la ventana y los acuses de recibo. En este ejemplo, el tamaño de la ventana inicial para una sesión TCP representada se establece en 3000 bytes. Cuando el emisor transmite 3000 bytes, espera por un acuse de recibo de los mismos antes de transmitir más segmentos para esta sesión. Una vez que el emisor obtiene este acuse de recibo del receptor, puede transmitir 3000 bytes adicionales.

TCP utiliza tamaños de ventana para tratar de aumentar la velocidad de transmisión hasta el flujo máximo que la red y el dispositivo de destino pueden admitir y, al mismo tiempo, minimizar las pérdidas y las retransmisiones.



Si se pierden los segmentos debido a la congestión, el receptor acusará recibo del último segmento secuencial recibido y responderá con un tamaño de ventana reducido.

T4. Ilustración 28

Reducción del tamaño de la ventana

Otra forma de controlar el flujo de datos es utilizar tamaños de ventana dinámicos. Cuando los recursos de la red son limitados, TCP puede reducir el tamaño de la ventana para lograr que los segmentos recibidos sean reconocidos con mayor frecuencia. Esto reduce de forma efectiva la velocidad de transmisión porque el origen espera que se de acuse de recibo de los datos con más frecuencia.

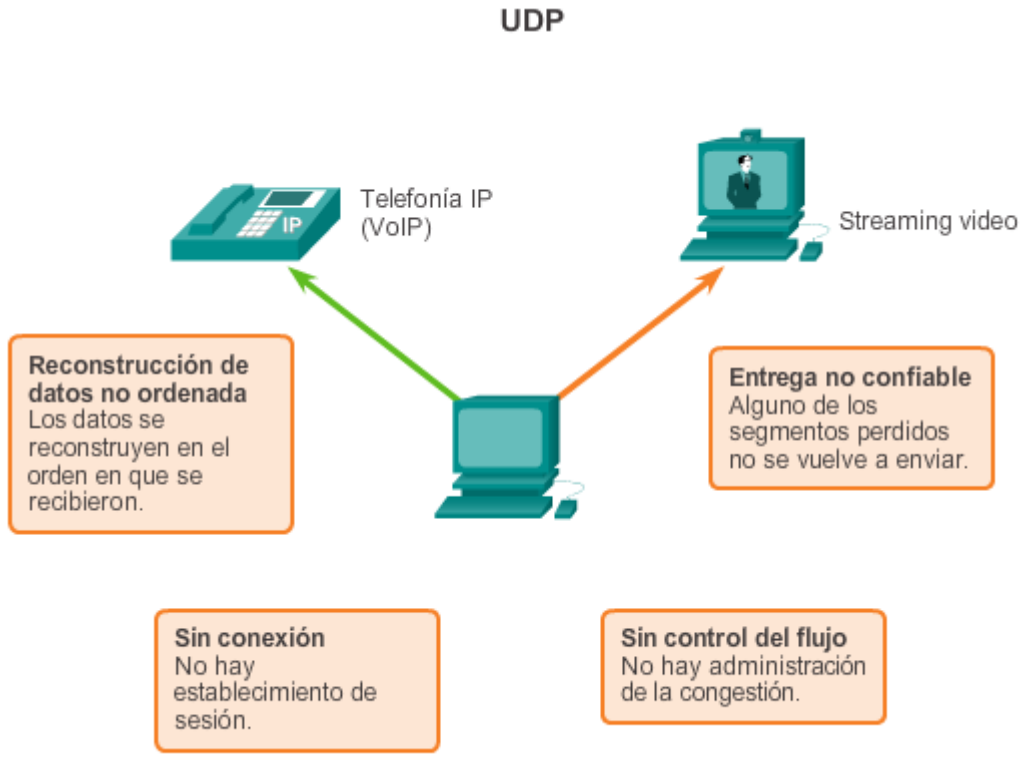
El host receptor envía el valor del tamaño de la ventana al host emisor para indicar la cantidad de bytes que puede recibir. Si el destino necesita disminuir la velocidad de comunicación debido, por ejemplo, a una memoria de búfer limitada, puede enviar un valor más pequeño del tamaño de la ventana al origen como parte del acuse de recibo.

Como se muestra en la ilustración, si un host receptor está congestionado, puede responder al host emisor con un segmento que especifique un tamaño reducido de la ventana. En esta ilustración, se muestra que se produjo la pérdida de uno de los segmentos. El receptor cambió el campo de la ventana en el encabezado TCP de los segmentos devueltos en esta conversación de 3000 a 1500. Esto hizo que el emisor redujera el tamaño de la ventana a 1500.

Después de un período de transmisión sin pérdidas de datos ni recursos limitados, el receptor comienza a aumentar el campo de la ventana, lo que reduce la sobrecarga en la red, ya que se deben enviar menos acuses de recibo. El tamaño de la ventana sigue aumentando hasta que se produce la pérdida de datos, lo que provoca que disminuya el tamaño de la ventana.

Este aumento y disminución dinámicos del tamaño de la ventana es un proceso continuo en TCP. En redes altamente eficaces, los tamaños de la ventana pueden ser muy grandes, porque no se pierden datos. En redes en las que la infraestructura subyacente está bajo presión, es probable que el tamaño de la ventana se mantenga pequeño.

3. Protocolo de datagrama de usuario (UDP)



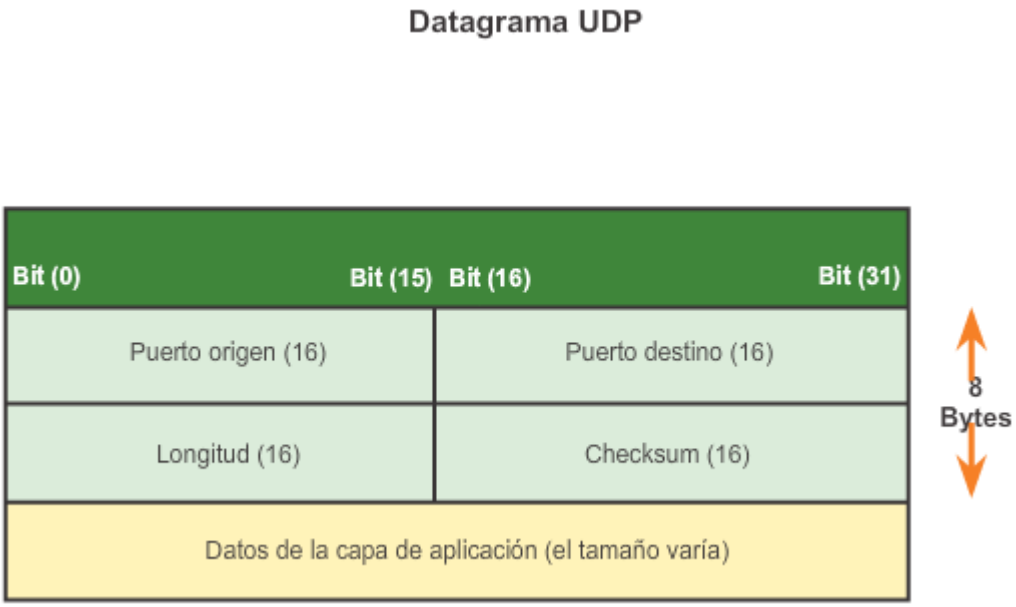
T4. Ilustración 29

Protocolo de datagramas de usuario (UDP)

UDP se considera un protocolo de transporte de máximo esfuerzo, descrito en RFC 768. UDP es un protocolo de transporte liviano que ofrece la misma segmentación y rearmado de datos que TCP, pero sin la confiabilidad y el control del flujo de TCP. UDP es un protocolo tan simple que, por lo general, se lo describe en términos de lo que no hace en comparación con TCP.

Como se muestra en la ilustración, las siguientes características describen a UDP:

- **Sin conexión:** UDP no establece una conexión entre los hosts antes de que se puedan enviar y recibir datos.
- **Entrega no confiable:** UDP no proporciona servicios para asegurar que los datos se entreguen con confianza. UDP no cuenta con procesos que hagan que el emisor vuelva a transmitir los datos que se pierden o se dañan.
- **Reconstrucción de datos no ordenada:** en ocasiones, los datos se reciben en un orden distinto del de envío. UDP no proporciona ningún mecanismo para rearmar los datos en su secuencia original. Los datos simplemente se entregan a la aplicación en el orden en que llegan.
- **Sin control del flujo:** UDP no cuenta con mecanismos para controlar la cantidad de datos que transmite el dispositivo de origen para evitar la saturación del dispositivo de destino. El origen envía los datos. Si los recursos en el host de destino se sobrecargan, es probable que dicho host descarte los datos enviados hasta que los recursos estén disponibles. A diferencia de TCP, en UDP no hay un mecanismo para la retransmisión automática de datos descartados.



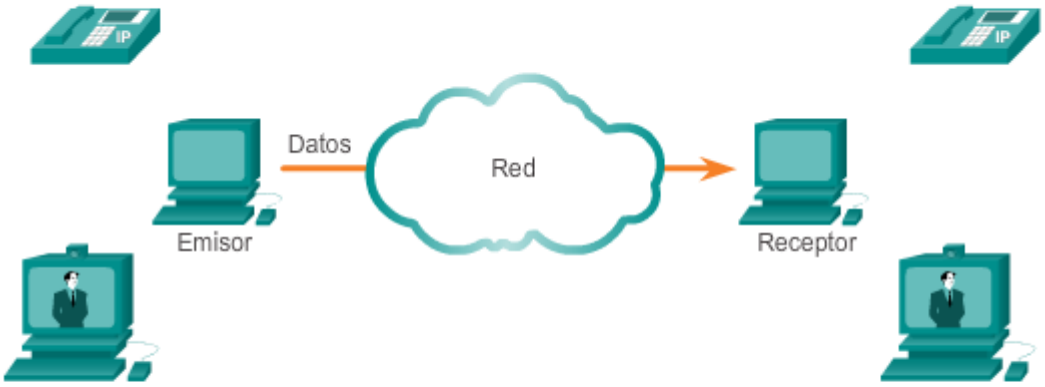
T4. Ilustración 30

Aunque UDP no incluye la confiabilidad y los mecanismos de control del flujo de TCP, como se muestra en la ilustración, la entrega de datos de baja sobrecarga de UDP lo convierte en un protocolo de transporte ideal para las aplicaciones que pueden tolerar cierta pérdida de datos. Las porciones de comunicación en UDP se llaman datagramas. El protocolo de la capa de transporte envía estos datagramas como máximo esfuerzo. Algunas aplicaciones que utilizan UDP son el Sistema de nombres de dominios (DNS), el streaming de video y la voz sobre IP (VoIP).

Uno de los requisitos más importantes para transmitir video en vivo y voz a través de la red es que los datos fluyan rápidamente. Las aplicaciones de video y de voz pueden tolerar cierta pérdida de datos con un efecto mínimo o imperceptible, y se adaptan perfectamente a UDP.

UDP es un protocolo sin estado, lo cual significa que ni el cliente ni el servidor están obligados a hacer un seguimiento del estado de la sesión de comunicación. Como se muestra en la ilustración, UDP no se ocupa de la confiabilidad ni del control del flujo. Los datos se pueden perder o recibir fuera de secuencia sin ningún mecanismo de UDP que pueda recuperarlos o reordenarlos. Si se requiere confiabilidad al utilizar UDP como protocolo de transporte, esta la debe administrar la aplicación.

Transporte de datos con baja sobrecarga de UDP



UDP no establece ninguna conexión antes de enviar datos.

UDP proporciona transporte de datos con baja sobrecarga, debido a que posee un encabezado de datagrama pequeño sin tráfico de administración de red.

T4. Ilustración 31

UDP es un protocolo simple que proporciona las funciones básicas de la capa de transporte. Tiene una sobrecarga mucho menor que TCP, ya que no está orientado a la conexión y no proporciona los mecanismos sofisticados de retransmisión, secuenciación y control del flujo que ofrecen confiabilidad.

Esto no significa que las aplicaciones que utiliza UDP sean siempre poco confiables ni que UDP sea un protocolo inferior. Solo quiere decir que estas funciones no las proporciona el protocolo de la capa de transporte, y se deben implementar aparte, si fuera necesario.

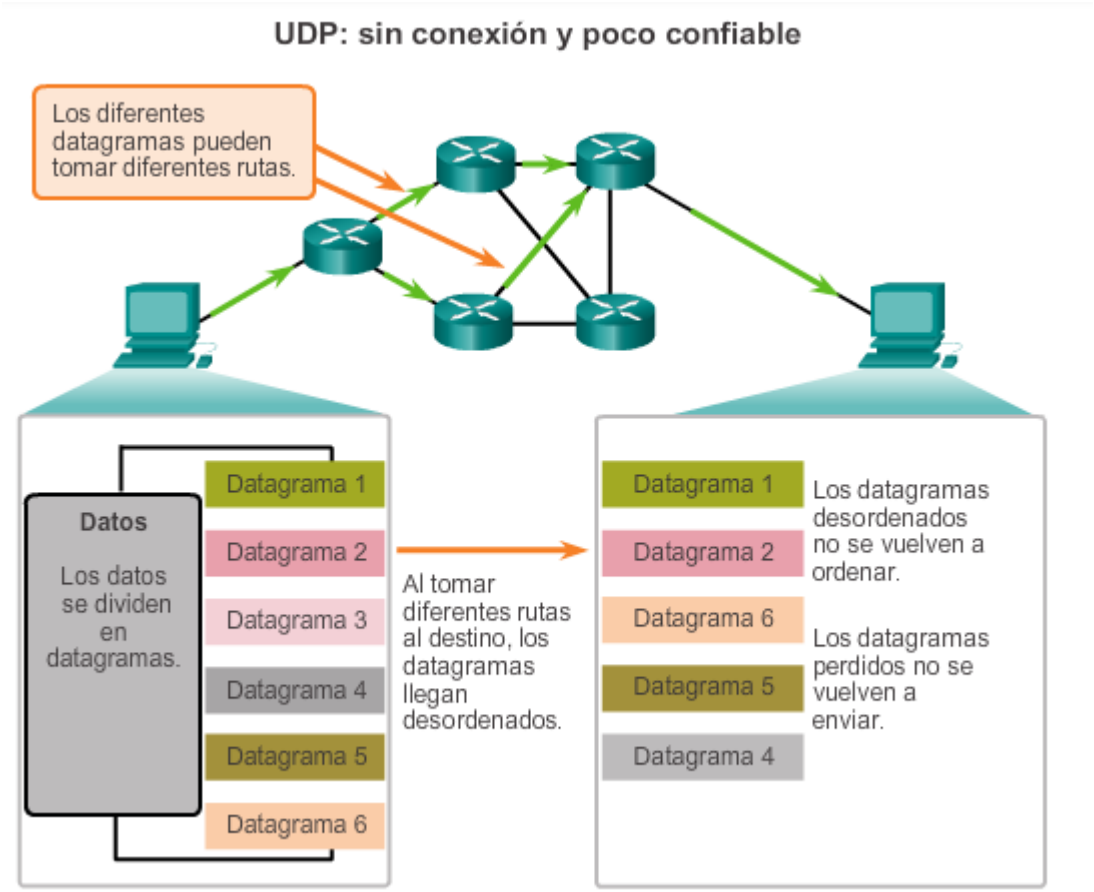
Pese a que es relativamente baja la cantidad total de tráfico UDP que puede encontrarse en una red típica, los protocolos clave de la capa de aplicación que utilizan UDP incluyen lo siguiente:

- Sistema de nombres de dominio (DNS)
- Protocolo simple de administración de red (SNMP, Simple Network Management Protocol)
- Protocolo de configuración dinámica de host (DHCP)
- Protocolo de información de enrutamiento (RIP)

- Protocolo de transferencia de archivos trivial (TFTP)
- Telefonía IP o voz sobre IP (VoIP)
- Juegos en línea

Algunas aplicaciones, como los juegos en línea o VoIP, pueden tolerar cierta pérdida de datos. Si estas aplicaciones utilizaran TCP, experimentarían largas demoras, ya que TCP detecta la pérdida de datos y los retransmite. Estas demoras serían más perjudiciales para el rendimiento de la aplicación que las pequeñas pérdidas de datos. Algunas aplicaciones, como DNS, simplemente reintentan el envío de la solicitud si no reciben ninguna respuesta; por lo tanto, no necesitan que TCP garantice la entrega de mensajes.

La baja sobrecarga del UDP es deseada por dichas aplicaciones.



T4. Ilustración 32

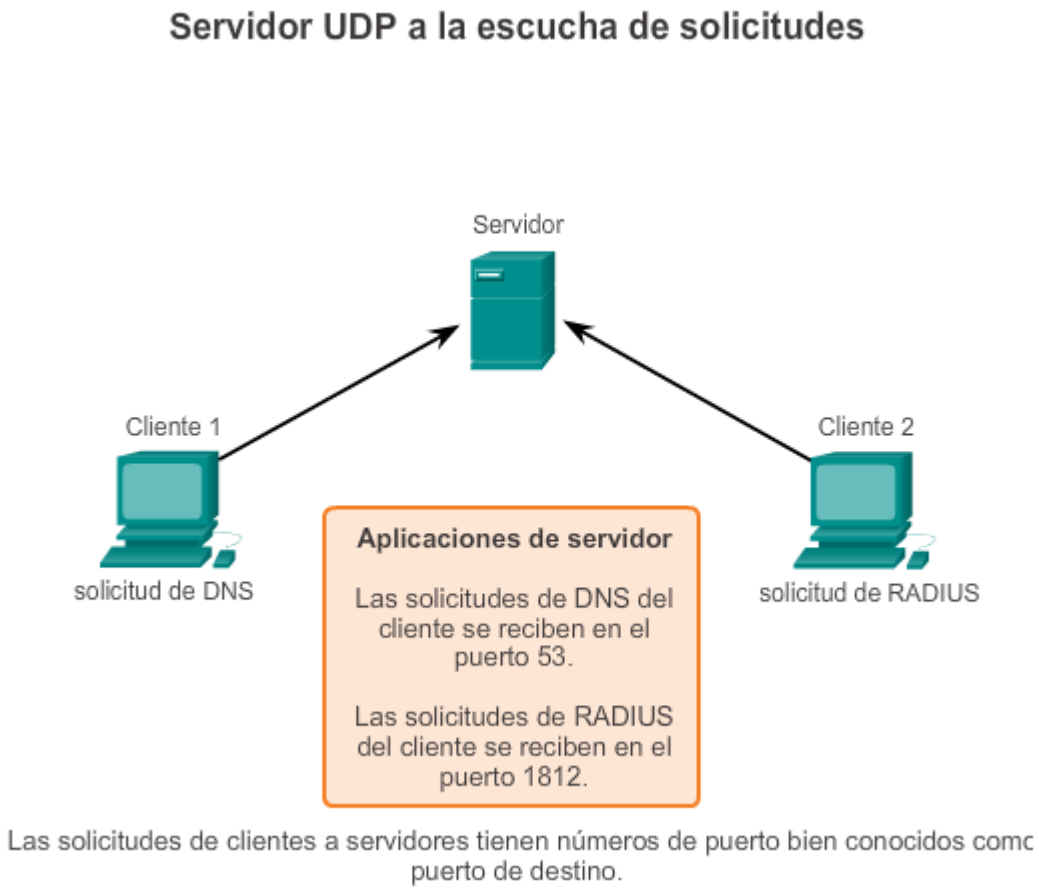
Ya que UDP opera sin conexión, las sesiones no se establecen antes de que se lleve a cabo la comunicación, como sucede con TCP. Se dice que UDP está basado en las transacciones; es decir, cuando una aplicación tiene datos para enviar, simplemente los envía.

Muchas aplicaciones que utilizan UDP envían pequeñas cantidades de datos que pueden ajustarse en un segmento. Sin embargo, algunas aplicaciones envían cantidades de datos más grandes que deben dividirse en varios segmentos. La PDU del UDP se conoce como un “datagrama”, aunque los términos “segmento” y “datagrama” se utilizan algunas veces de forma intercambiable para describir una PDU de la capa de transporte.

Cuando se envían datagramas múltiples a un destino, pueden tomar diferentes rutas y llegar en el orden equivocado. UDP no realiza un seguimiento de los números de secuencia de la manera en

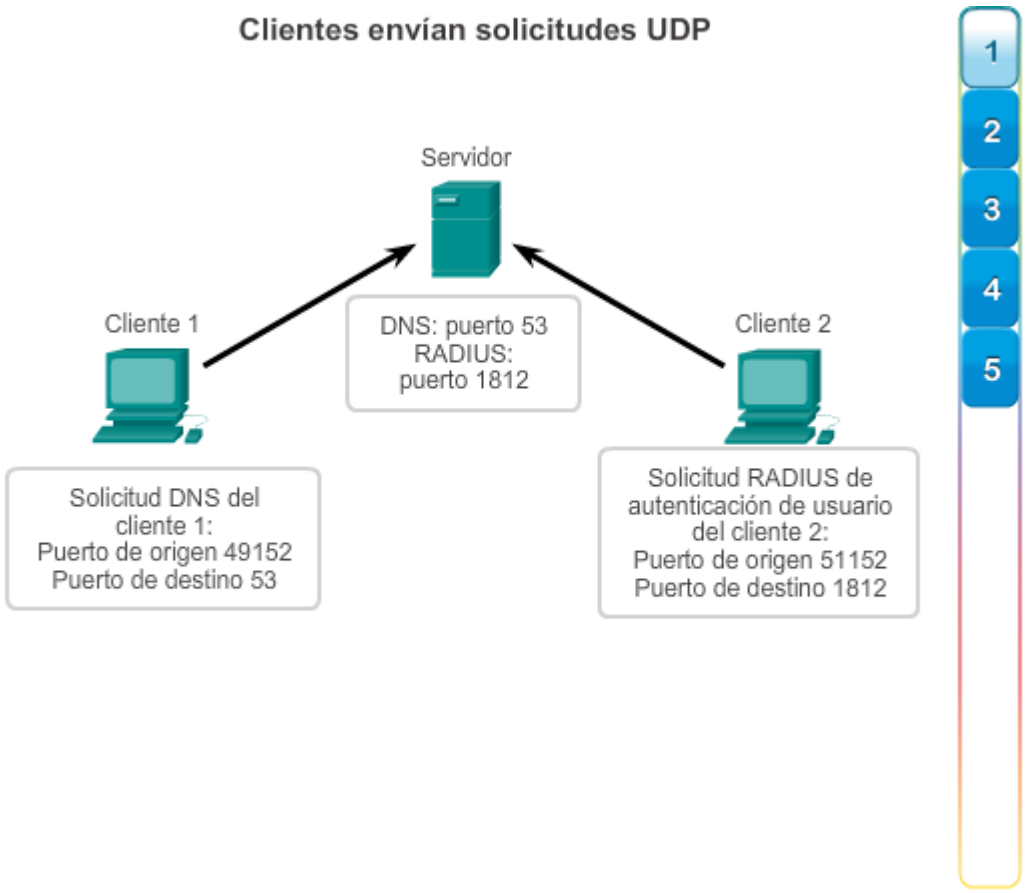
que lo hace TCP. UDP no tiene forma de reordenar datagramas en el orden en que se transmiten, como se muestra en la ilustración.

Por lo tanto, UDP simplemente reensambla los datos en el orden en que se recibieron y los envía a la aplicación. Si la secuencia de datos es importante para la aplicación, esta debe identificar la secuencia adecuada y determinar cómo se deben procesar los datos.



T4. Ilustración 33

Al igual que las aplicaciones basadas en TCP, a las aplicaciones de servidor basadas en UDP se les asignan números de puerto bien conocidos o registrados. Cuando estas aplicaciones o estos procesos se ejecutan en un servidor, aceptan los datos que coinciden con el número de puerto asignado. Cuando UDP recibe un datagrama destinado a uno de esos puertos, envía los datos de aplicación a la aplicación adecuada en base a su número de puerto.



T4. Ilustración 34. **Para consultar toda la información, pincha en la imagen.**

Como en TCP, la comunicación cliente/servidor la inicia una aplicación cliente que solicita datos de un proceso de servidor. El proceso de cliente UDP selecciona al azar un número de puerto del rango de números de puerto dinámicos y lo utiliza como puerto de origen para la conversación. Por lo general, el puerto de destino es el número de puerto bien conocido o registrado que se asigna al proceso de servidor.

Los números de puerto de origen seleccionados al azar colaboran con la seguridad. Si existe un patrón predecible para la selección del puerto de destino, un intruso puede simular el acceso a un cliente de manera más sencilla intentando conectarse al número de puerto que tenga mayor posibilidad de estar abierto.

Dado que no se crean sesiones con UDP, no bien los datos están listos para enviarse y los puertos están identificados, UDP puede formar los datagramas y pasarlos a la capa de red para direccionarlos y enviarlos a la red.

Una vez que el cliente selecciona los puertos de origen y de destino, este mismo par de puertos se utiliza en el encabezado de todos los datagramas que se utilizan en la transacción. Para la devolución de datos del servidor al cliente, se invierten los números de puerto de origen y destino en el encabezado del datagrama.

Desplácese por las ilustraciones a la derecha para ver los detalles de los procesos de cliente UDP.