Planificación y Administración de Redes

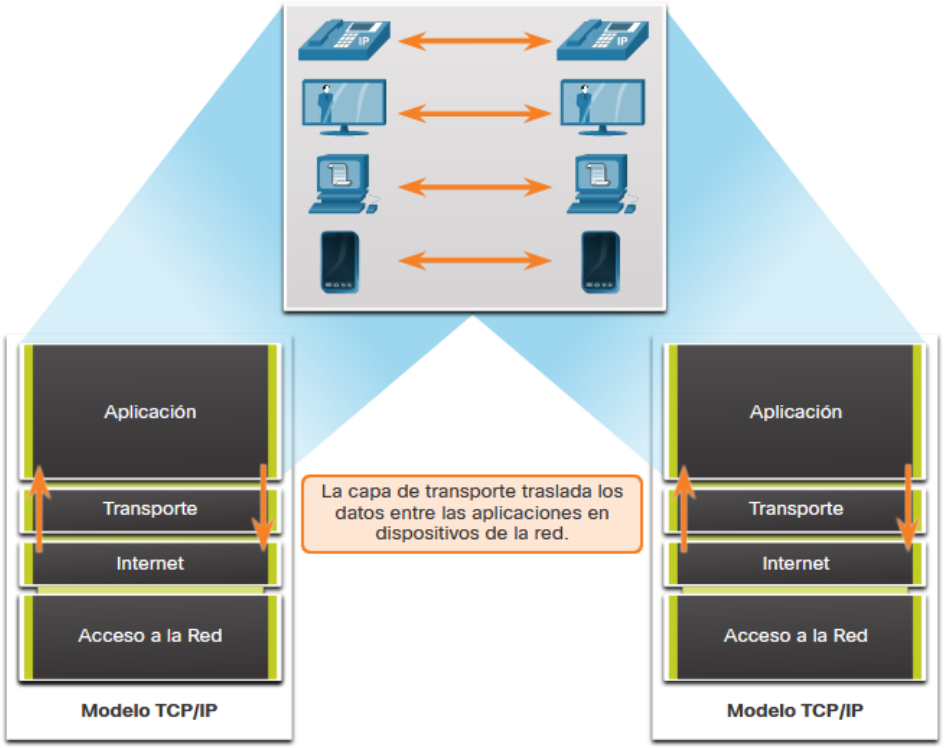
T.6 La capa de transporte

Índice

1 Transporte de datos

Función de la capa de transporte

Los programas de capa de aplicación generan datos que deben intercambiarse entre los hosts de origen y de destino. **La capa de transporte es responsable de las comunicaciones lógicas entre aplicaciones que se ejecutan en diferentes hosts.**



Función de la capa de transporte

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 no tiene conocimiento del tipo de host de destino, el tipo de medio por el que deben viajar los datos, la ruta tomada por los datos, la congestión en un enlace o el tamaño de la red.

Responsabilidades de la capa de Transporte

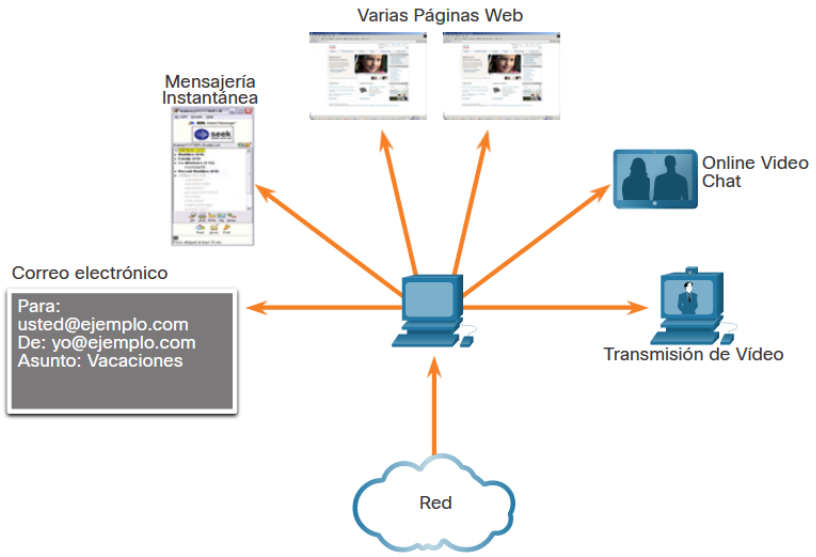
La capa de transporte tiene muchas responsabilidades:

- **Seguimiento de conversaciones individuales**
- **Segmentación de Datos y Rearmado de Segmentos**
- **Agregar Información de Encabezado**
- **Identificación de las Aplicaciones**
- **Multiplexión de Conversaciones**

Responsabilidades de la capa de Transporte

Seguimiento de conversaciones individuales

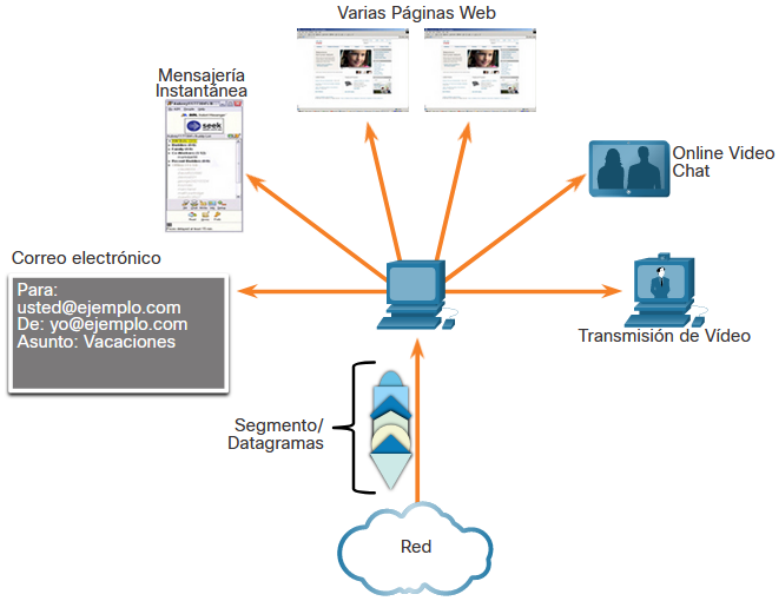
En la capa de transporte, **cada conjunto de datos que fluye entre una aplicación de origen y una aplicación de destino se conoce como una conversación y se rastrea por separado**. Es responsabilidad de la capa de transporte mantener y hacer un seguimiento de todas estas conversaciones.



Responsabilidades de la capa de Transporte

Segmentación de Datos y Rearmado de Segmentos

Es responsabilidad de la capa de transporte **dividir los datos de la aplicación en bloques de tamaño adecuado.** Dependiendo del protocolo de capa de transporte utilizado, los bloques de capa de transporte se denominan segmentos o datagramas.



Responsabilidades de la capa de Transporte

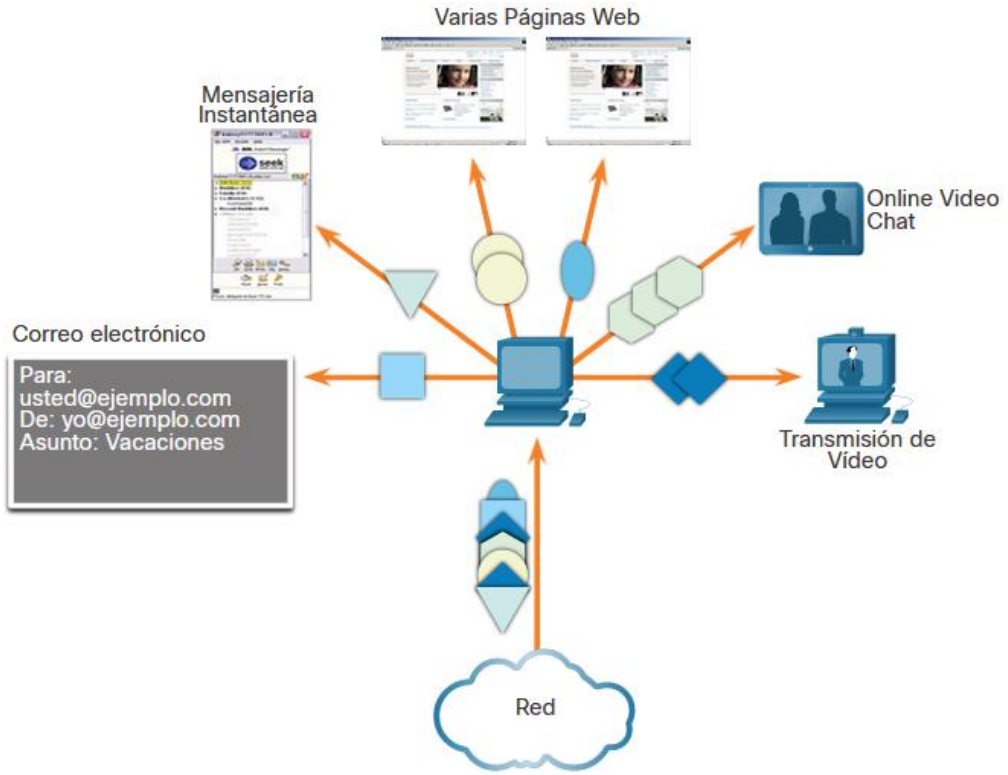
Agregar Información de Encabezado

El protocolo de capa de transporte también agrega información de encabezado que contiene datos binarios organizados en varios campos a cada bloque de datos. Los valores de estos campos permiten que los distintos protocolos de la capa de transporte lleven a cabo variadas funciones de administración de la comunicación de datos.

Por ejemplo, el host receptor utiliza la información de encabezado para volver a ensamblar los bloques de datos en un flujo de datos completo para el programa de capa de aplicación de recepción.

Responsabilidades de la capa de Transporte

La capa de transporte garantiza que incluso con múltiples aplicaciones que se ejecutan en un dispositivo, todas las aplicaciones reciben los datos correctos.



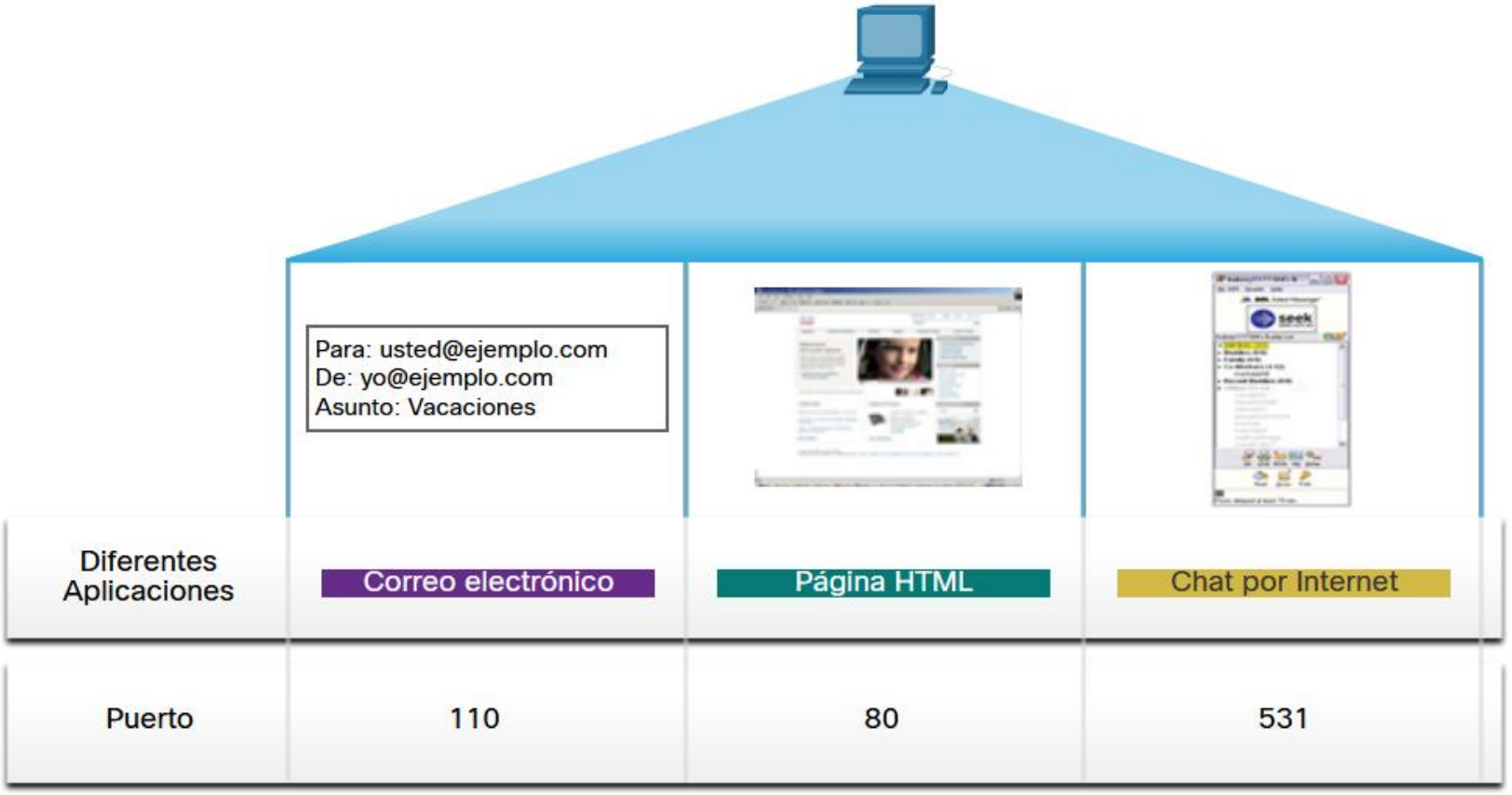
Responsabilidades de la capa de Transporte

Identificación de las Aplicaciones

La capa de transporte debe poder separar y administrar varias comunicaciones con diferentes necesidades de requisitos de transporte. Para pasar flujos de datos a las aplicaciones adecuadas, **la capa de transporte identifica la aplicación de destino utilizando un identificador llamado número de puerto.**

Responsabilidades de la capa de Transporte

A cada proceso de software que necesita acceder a la red se le asigna un número de puerto único para ese host.



Responsabilidades de la capa de Transporte

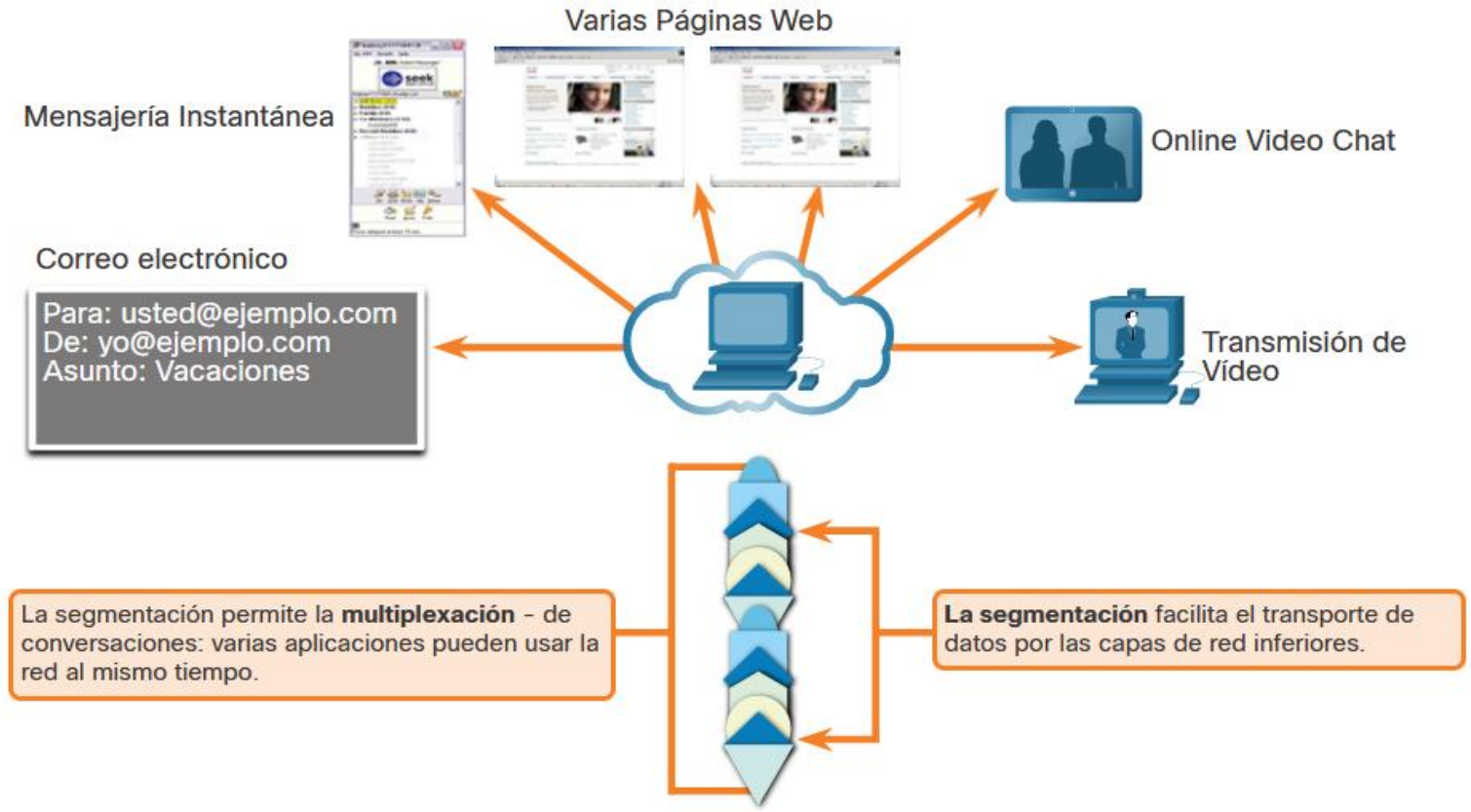
Multiplexión de Conversaciones

El envío de algunos tipos de datos (por ejemplo, una transmisión de video) a través de una red, como una transmisión de comunicación completa, puede consumir todo el ancho de banda disponible. Esto evitaría que se produzcan otras conversaciones de comunicación al mismo tiempo. También podría dificultar la recuperación de errores y la retransmisión de datos dañados.

La capa de transporte utiliza segmentación y multiplexación para permitir que diferentes conversaciones de comunicación se intercalen en la misma red.

Responsabilidades de la capa de Transporte

La verificación de errores se puede realizar en los datos del segmento, para determinar si el segmento se modificó durante la transmisión.



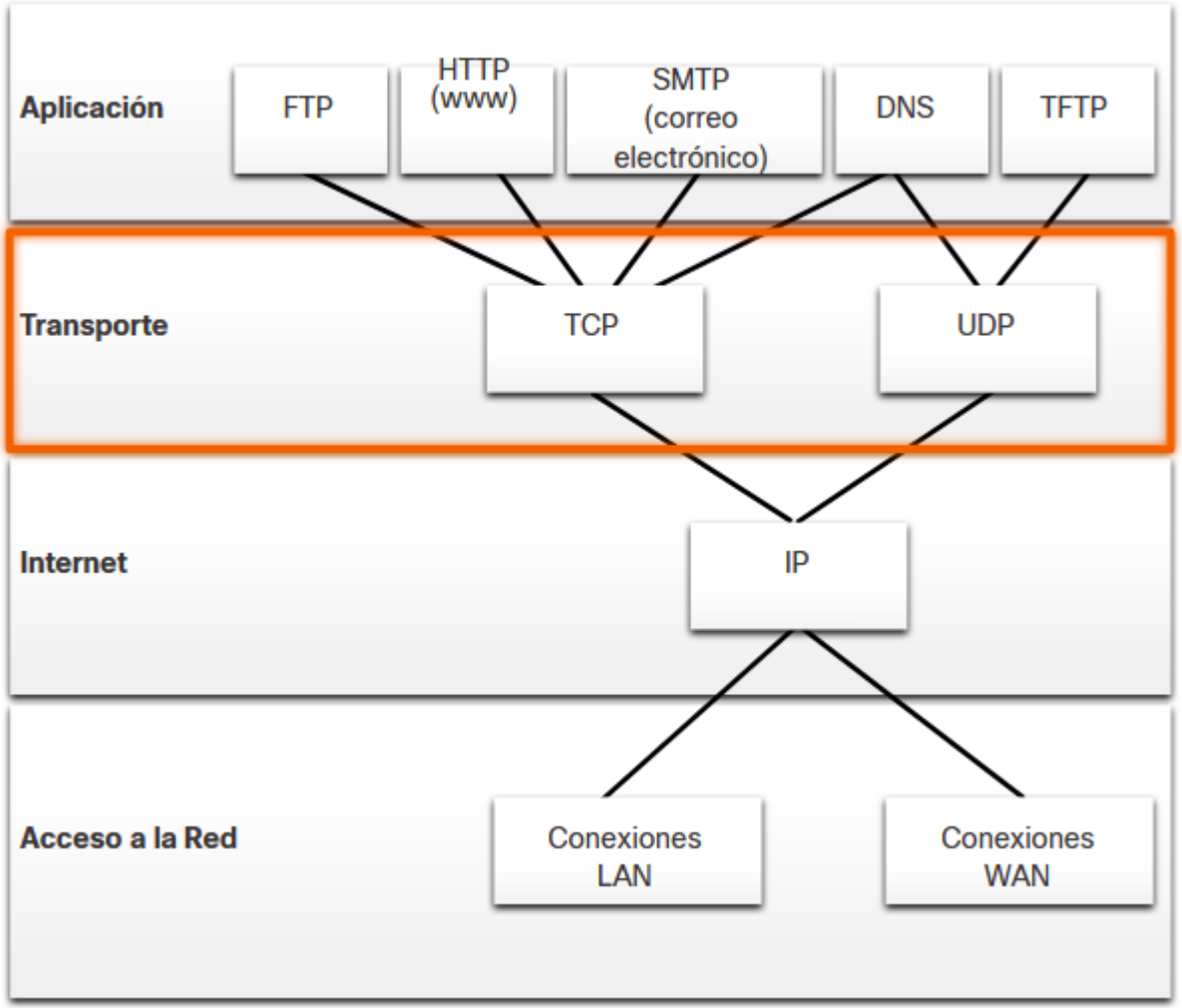
Protocolos de capa de transporte

IP se ocupa solo de la estructura, el direccionamiento y el routing de paquetes. IP no especifica la manera en que se lleva a cabo la entrega o el transporte de los paquetes.

Los protocolos de capa de transporte especifican cómo transferir mensajes entre hosts y son responsables de administrar los requisitos de fiabilidad de una conversación. La capa de transporte incluye los protocolos **TCP y UDP**.

Las diferentes aplicaciones tienen diferentes requisitos de confiabilidad de transporte. Por lo tanto, TCP/IP proporciona dos protocolos de capa de transporte.

Protocolos de capa de transporte



2 Protocolo TCP

Protocolo de Control de Transmisión (TCP)

IP solo se refiere a la estructura, direccionamiento y enrutamiento de paquetes, desde el remitente original hasta el destino final. IP no es responsable de garantizar la entrega o determinar si es necesario establecer una conexión entre el remitente y el receptor.

TCP se considera un protocolo de la capa de transporte confiable y completo, que garantiza que todos los datos lleguen al destino. TCP incluye campos que **garantizan** la entrega de los datos de la aplicación. Estos campos requieren un **procesamiento adicional** por parte de los hosts de envío y recepción.

Protocolo de Control de Transmisión (TCP)

TCP proporciona confiabilidad y control de flujo mediante estas operaciones básicas:

- **Enumerar y rastrear segmentos de datos transmitidos a un host específico desde una aplicación específica**
- **Confirmar datos recibidos**
- **Retransmitir cualquier información no reconocida después de un cierto período de tiempo**
- **Secuenciar datos que pueden llegar en un orden incorrecto**
- **Enviar datos a una velocidad eficiente que sea aceptable por el receptor**

Para mantener el estado de una conversación y realizar un seguimiento de la información, TCP debe establecer primero una conexión entre el remitente y el receptor. Es por eso que **TCP** se conoce como un **protocolo orientado a la conexión.**

Características de TCP

Para comprender las diferencias entre TCP y UDP, es importante comprender cómo cada protocolo implementa funciones de confiabilidad específicas y cómo rastrea las conversaciones.

Además de admitir las funciones básicas de segmentación y reensamblado de datos, TCP también proporciona los siguientes servicios:

Establece una sesión - **TCP es un protocolo orientado a la conexión que negocia y establece una sesión entre los dispositivos de origen y destino antes de reenviar cualquier tráfico.** 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.

Características de TCP

TCP también proporciona los siguientes servicios:

Garantiza una entrega confiable - por muchas razones, es posible que un segmento se corrompa o se pierda por completo, ya que se transmite a través de la red. **TCP asegura que cada segmento que envía la fuente llega al destino.**

Proporciona orden de entrega- Debido a que las redes pueden proporcionar múltiples rutas que pueden tener diferentes velocidades de transmisión, los datos pueden llegar en el orden incorrecto. Al numerar y secuenciar los segmentos, **TCP garantiza que los segmentos se vuelvan a ensamblar en el orden correcto**.

Características de TCP

TCP también proporciona los siguientes servicios:

Admite control de flujo - los hosts de red tienen recursos limitados (es decir, memoria y potencia de procesamiento). 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 un cabo TCP, que regula la cantidad de datos que transmite el origen. **El control de flujo puede evitar la necesidad de retransmitir los datos cuando los recursos del host receptor se ven desbordados**.

Encabezado TCP

TCP es un protocolo con estado, lo que significa que realiza un seguimiento del estado de la sesión de comunicación. Para hacer un seguimiento del estado de una sesión, TCP registra qué información se envió y qué información se reconoció. La sesión con estado comienza con el establecimiento de la sesión y termina con la finalización de la sesión. Un segmento TCP agrega 20 bytes de sobrecarga al encapsular los datos de la capa de aplicación.

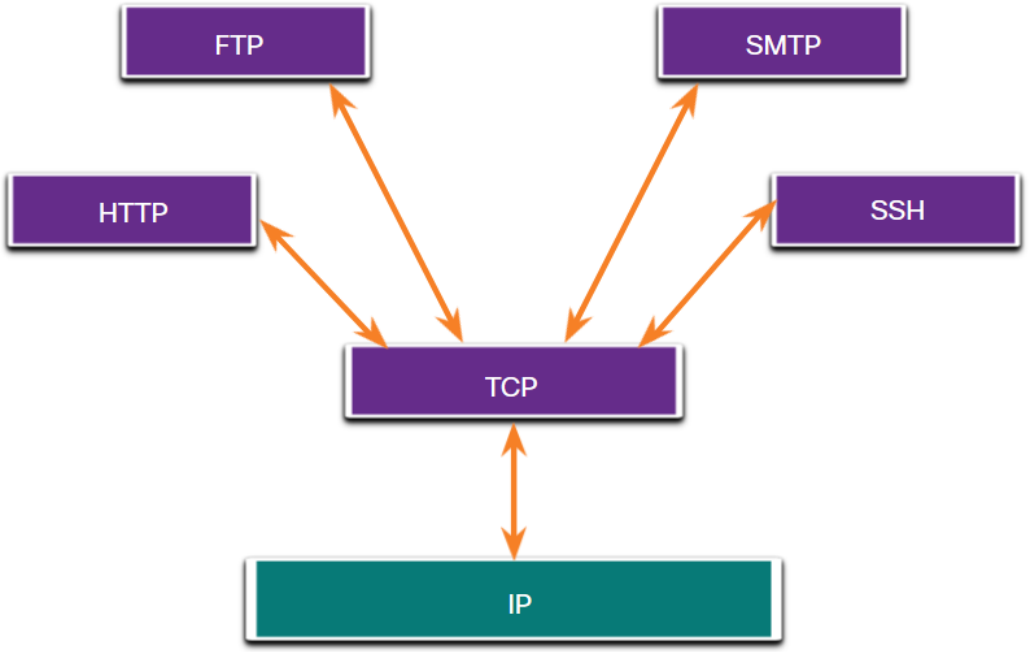
Encabezado TCP

Encabezado TCP

Campo de Encabezado TCP	Descripción
Puerto de Origen	Campo de 16 bits utilizado para identificar la aplicación de origen por número de puerto.
Puerto de Destino	Un campo de 16 bits utilizado para identificar la aplicación de destino por puerto número.
Secuencia de Números	Campo de 32 bits utilizado para reensamblar datos.
Número de Acuse de Recibo	Un campo de 32 bits utilizado para indicar que se han recibido datos y el siguiente byte esperado de la fuente.
Longitud del Encabezado	Un campo de 4 bits conocido como «desplazamiento de datos» que indica la propiedad longitud del encabezado del segmento TCP.
Reservado	Un campo de 6 bits que está reservado para uso futuro.
Bits de Control	Un campo de 6 bits utilizado que incluye códigos de bits, o indicadores, que indican el propósito y función del segmento TCP.
Tamaño de la ventana	Un campo de 16 bits utilizado para indicar el número de bytes que se pueden aceptar a la vez.
Suma de Comprobación	A 16-bit field used for error checking of the segment header and data.
Urgente	Campo de 16 bits utilizado para indicar si los datos contenidos son urgentes.

Aplicaciones que utilizan TCP

TCP es un buen ejemplo de cómo las diferentes capas del conjunto de protocolos TCP / IP tienen roles específicos. TCP maneja todas las tareas asociadas con la división del flujo de datos en segmentos, proporcionando confiabilidad, controlando el flujo de datos y reordenando segmentos. TCP libera la aplicación de tener que administrar estas tareas. Las aplicaciones, simplemente pueden enviar el flujo de datos a la capa de transporte y utilizar los servicios de TCP.



3 Protocolo UDP

Protocolo de Datagramas de Usuario (UDP)

UDP es un protocolo de capa de transporte más simple que TCP. No proporciona confiabilidad y control de flujo, lo que significa que requiere menos campos de encabezado.

Debido a que los procesos UDP remitente y receptor no tienen que administrar la confiabilidad y el control de flujo, esto significa que **los datagramas UDP se pueden procesar más rápido que los segmentos TCP.**

UDP proporciona las funciones básicas para entregar segmentos de datos entre las aplicaciones adecuadas, con **muy poca sobrecarga y revisión de datos.**

Protocolo de Datagramas de Usuario (UDP)

UDP es un protocolo sin conexión. Debido a que UDP no proporciona fiabilidad ni control de flujo, no requiere una conexión establecida. Debido a que UDP no realiza un seguimiento de la información enviada o recibida entre el cliente y el servidor, UDP también se conoce como protocolo sin estado.

UDP también se conoce como un protocolo de entrega de mejor esfuerzo porque no hay reconocimiento de que los datos se reciben en el destino. Con UDP, no existen procesos de capa de transporte que informen al emisor si la entrega se realizó correctamente.

Características de UDP

UDP es un protocolo de transporte del mejor esfuerzo. **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.

Las características UDP incluyen lo siguiente:

- **Los datos se reconstruyen en el orden en que se recibieron.**
- **Los segmentos perdidos no se vuelven a enviar.**
- **No hay establecimiento de sesión.**
- **El emisor no está informado sobre la disponibilidad de recursos.**

Encabezado UDP

UDP es un protocolo sin estado, lo que significa que ni el cliente ni el servidor rastrean el estado de la sesión de comunicación. Si se requiere confiabilidad al utilizar UDP como protocolo de transporte, a esta la debe administrar la aplicación.

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 en vivo pueden tolerar cierta pérdida de datos con un efecto mínimo o imperceptible, y se adaptan perfectamente a UDP.**

Encabezado UDP

Los bloques de comunicación en UDP se denominan datagramas o segmentos. Estos datagramas se envían como el mejor esfuerzo por el protocolo de la capa de transporte.

Encabezado UDP

Campo de Encabezado UDP	Descripción
Puerto de Origen	Campo de 16 bits utilizado para identificar la aplicación de origen por número de puerto.
Puerto de Destino	Un campo de 16 bits utilizado para identificar la aplicación de destino por puerto número.
Longitud	Campo de 16 bits que indica la longitud del encabezado del datagrama UDP.
Suma de comprobación	Campo de 16 bits utilizado para la comprobación de errores del encabezado y los datos del datagrama.

Aplicaciones que utilizan UDP

Existen tres tipos de aplicaciones que son las más adecuadas para UDP:

Aplicaciones de video y multimedia en vivo: - estas aplicaciones pueden tolerar cierta pérdida de datos, pero requieren poco o ningún retraso. Los ejemplos incluyen **VoIP** y la transmisión de video en vivo.

Solicitudes simples de solicitud y respuesta: - aplicaciones con transacciones simples en las que un host envía una solicitud y puede o no recibir una respuesta. Los ejemplos incluyen **DNS** y **DHCP**.

Aplicaciones que manejan la confiabilidad por sí mismas: - comunicaciones unidireccionales donde el control de flujo, la detección de errores, los reconocimientos y la recuperación de errores no son necesarios o la aplicación puede manejarlos. Los ejemplos incluyen **SNMP** y **TFTP**.

4 Números de puerto

Comunicaciones múltiples separadas

Los protocolos de capa de transporte TCP y UDP utilizan números de puerto para administrar múltiples conversaciones simultáneas.

Los campos de encabezado TCP y UDP identifican un número de puerto de aplicación de origen y destino.

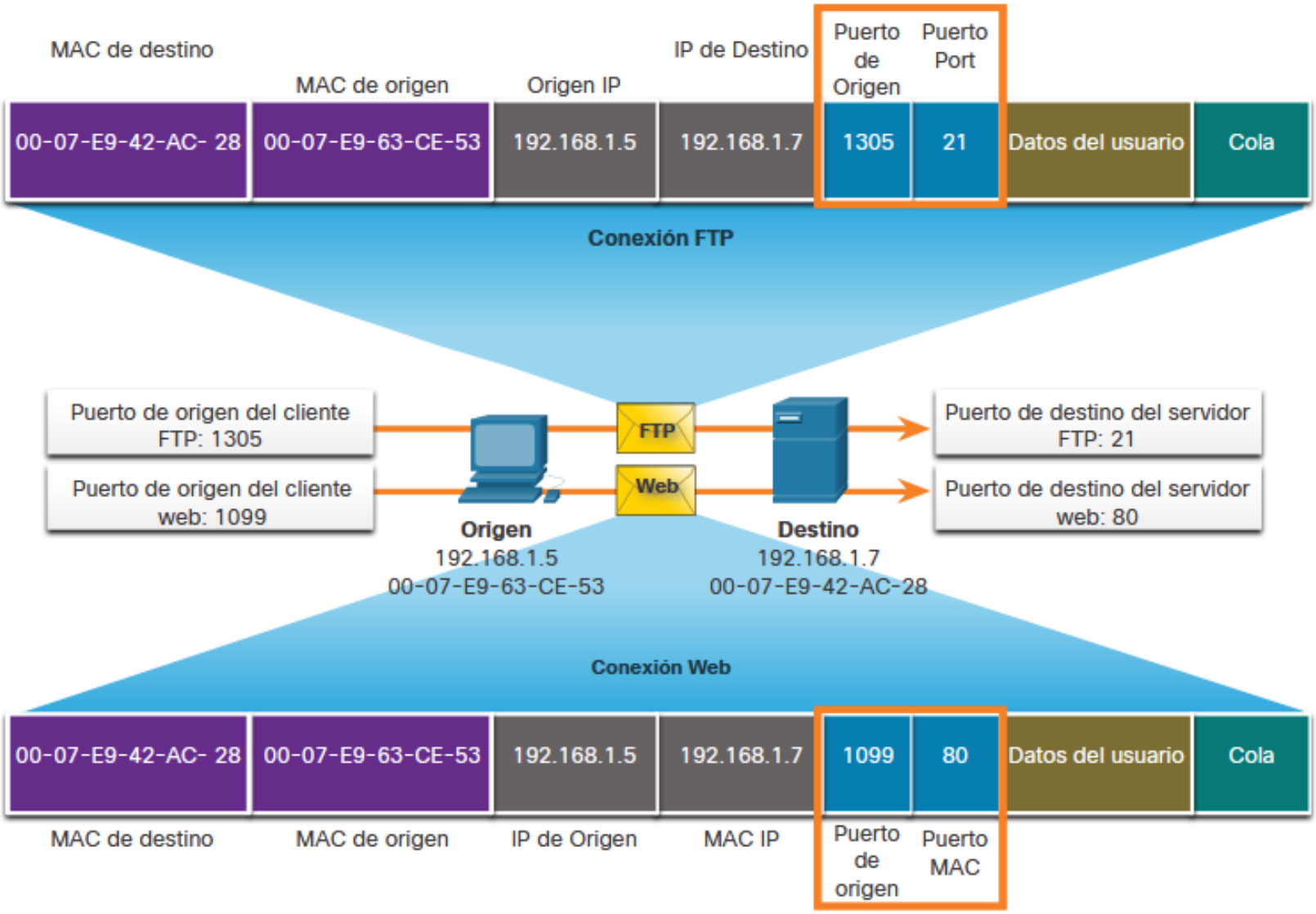
El número de puerto de origen está asociado con la aplicación de origen en el host local, mientras que el número de puerto de destino está asociado con la aplicación de destino en el host remoto.

Pares de sockets

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. **Se conoce como socket a la combinación de la dirección IP de origen y el número de puerto de origen, o de la dirección IP de destino y el número de puerto de destino.**

Por ejemplo-> 192.168.1.7:1234

Pares de sockets



Pares de sockets

Los sockets permiten que los diversos procesos que se ejecutan en un cliente se distingan entre sí. También permiten la diferenciación de diferentes conexiones a un proceso de servidor.

El número de puerto de origen 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.

Grupos de números de puerto

La Autoridad de Números Asignados de Internet (IANA) es la organización de estándares responsable de asignar varios estándares de direccionamiento, incluidos los números de puerto de 16 bits. Los 16 bits utilizados para identificar los números de puerto de origen y destino proporcionan un rango de puertos entre 0 y 65535.

La IANA ha dividido el rango de números en los siguientes tres grupos de puertos.

Grupos de números de puerto

Grupo de puertos	Rango de números	Descripción
Puertos bien conocidos	**0 to 1,023**	• Estos números de puerto están reservados para servicios comunes o populares y aplicaciones como navegadores web, clientes de correo electrónico y acceso remoto clientes. • Los puertos conocidos definidos para aplicaciones de servidor comunes permiten para identificar fácilmente el servicio asociado requerido.
Puertos registrados	**1,024 to 49,151**	• Estos números de puerto son asignados por IANA a una entidad solicitante a utilizar con procesos o aplicaciones específicos. • Estos procesos son principalmente aplicaciones individuales que un usuario ha elegido instalar, en lugar de aplicaciones comunes que recibir un número de puerto conocido. • Por ejemplo, Cisco ha registrado el puerto 1812 para su servidor RADIUS proceso de autenticación
Privada and/or **puertos dinámicos**	**49,152 to 65,535**	• Estos puertos también se conocen como puertos *efímeros*. • El sistema operativo del cliente generalmente asigna números de puerto dinámicamente cuando se inicia una conexión a un servicio. • El puerto dinámico se utiliza para identificar la aplicación del cliente. durante la comunicación

Números de puertos bien conocidos

Número de puerto	Protocolo	Aplicación
20	TCP	Protocolo de transferencia de archivos (FTP) - Datos
21	TCP	Protocolo de transferencia de archivos (FTP) - Control
22	TCP	Secure Shell (SSH)
23	TCP	Telnet
25	TCP	Protocolo simple de transferencia de correo (SMTP)
53	UDP, TCP	Servicio de nombres de dominio (DNS, Domain Name System)
67	UDP	Protocolo de configuración dinámica de host (DHCP): servidor
68	UDP	Protocolo de configuración dinámica de host: cliente
69	UDP	Protocolo trivial de transferencia de archivos (TFTP)
80	TCP	Protocolo de transferencia de hipertexto (HTTP)
110	TCP	Protocolo de oficina de correos, versión 3 (POP3)
143	TCP	Protocolo de acceso a mensajes de Internet (IMAP)
161	UDP	Protocolo simple de administración de redes (SNMP)
443	TCP	Protocolo seguro de transferencia de hipertexto (HTTPS)

El comando *netstat*

Las conexiones TCP no descritas pueden representar una importante amenaza a la seguridad. Pueden indicar que algo o alguien está conectado al host local. **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.**

El comando **netstat** enumera los protocolos en uso, la dirección local y los números de puerto, la dirección externa y los números de puerto y el estado de la conexión.

```
C:\> netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.1.124:3126     192.168.0.2:netbios-ssn    ESTABLISHED
  TCP    192.168.1.124:3158     207.138.126.152:http       ESTABLISHED
  TCP    192.168.1.124:3159     207.138.126.169:http       ESTABLISHED
  TCP    192.168.1.124:3160     207.138.126.169:http       ESTABLISHED
  TCP    192.168.1.124:3161     sc.msn.com:http            ESTABLISHED
  TCP    192.168.1.124:3166     www.cisco.com:http         ESTABLISHED
(output omitted)
C:\>
```

5 Proceso de comunicación TCP

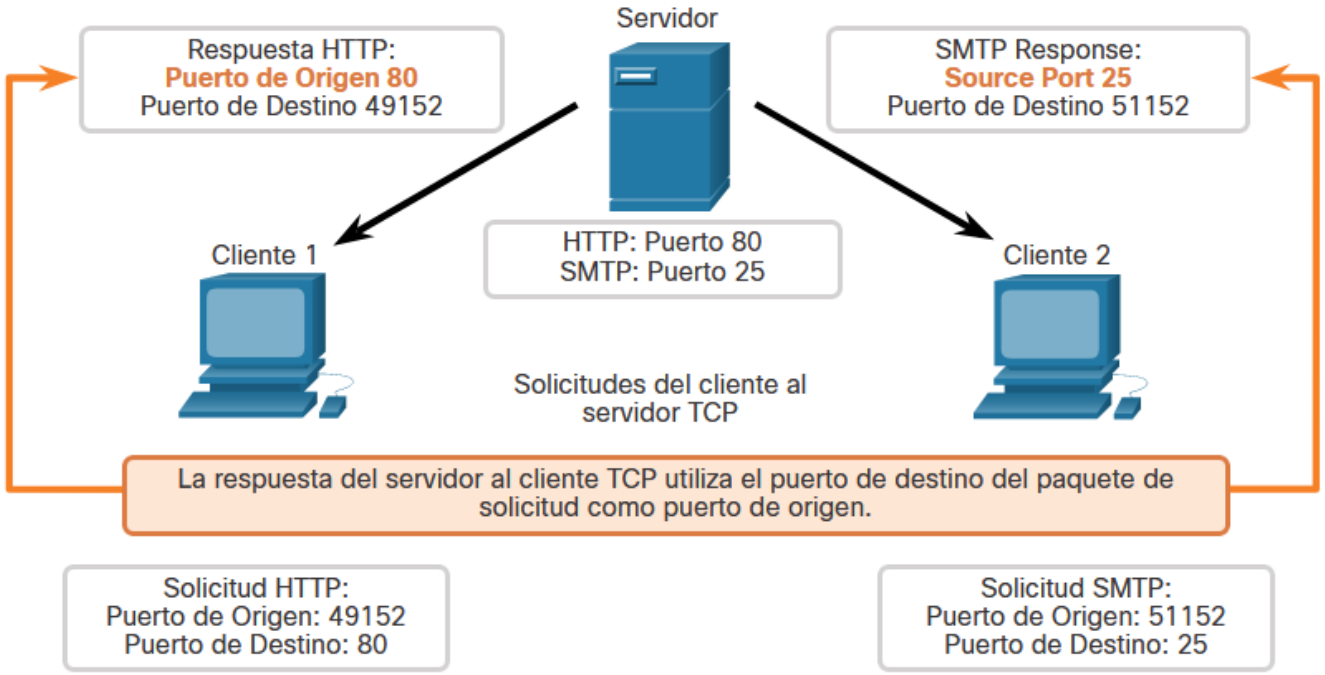
Procesos del Servidor TCP

Cada proceso de aplicación que se ejecuta en el servidor utiliza un número de puerto. El número de puerto es asignado automáticamente o configurado manualmente por un 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. Por ejemplo, un host que ejecuta una aplicación de servidor web y una aplicación de transferencia de archivos no puede tener ambos configurados para usar el mismo puerto, como el puerto TCP 80.

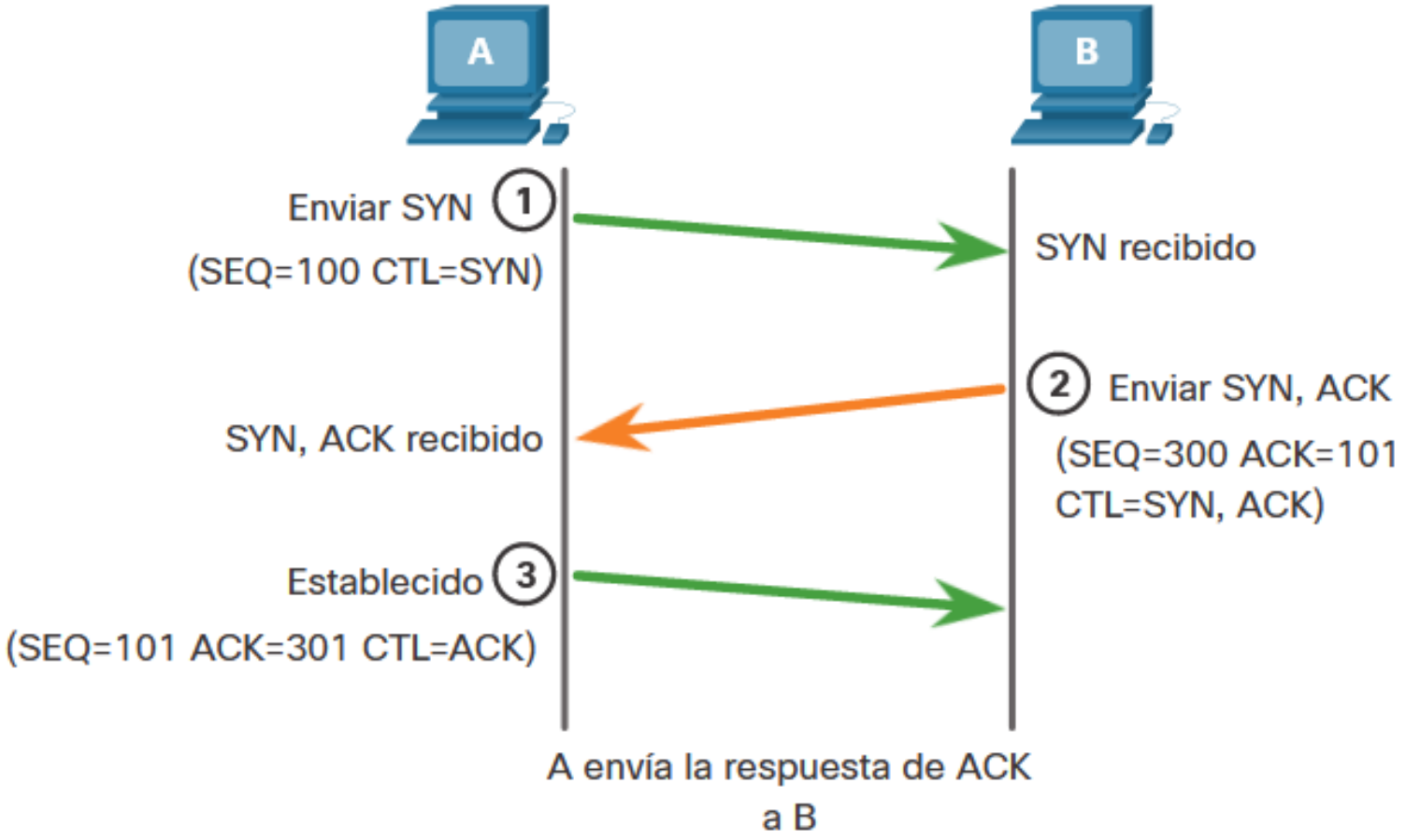
Procesos del Servidor TCP

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 abiertos** simultáneamente en un servidor, uno **para cada aplicación de servidor activa.**



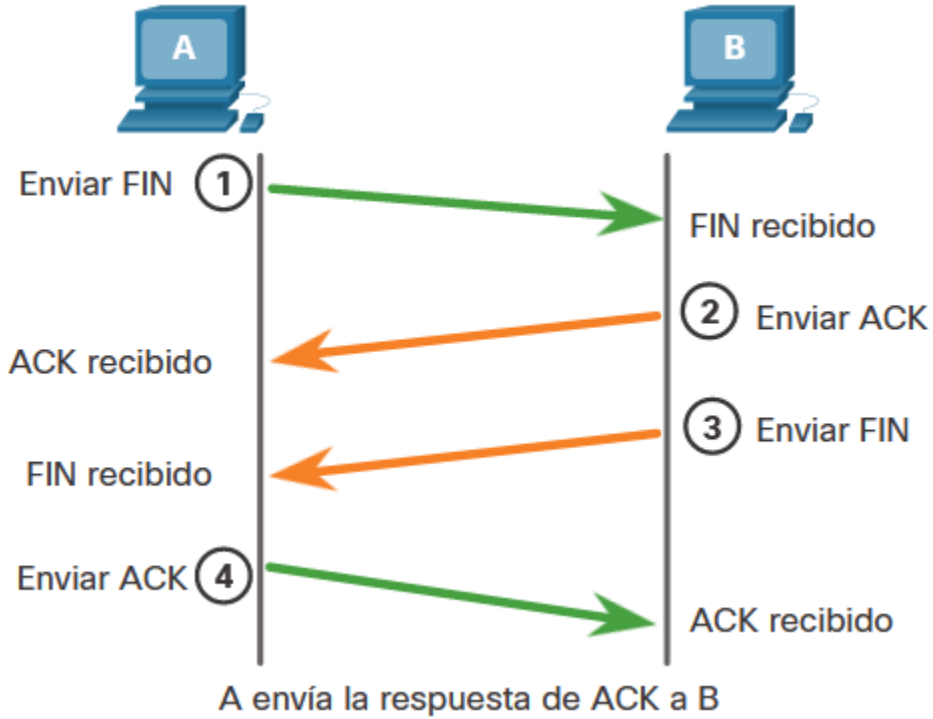
Establecimiento de Conexiones TCP

En las conexiones TCP, el cliente host establece la conexión con el servidor mediante el proceso de enlace de tres vías.



Terminación de sesión

Para cerrar una conexión, se debe establecer el marcador de control de finalización (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 de reconocimiento (ACK).** Por lo tanto, para terminar una conversación simple admitida por TCP, se requieren cuatro Intercambios para finalizar ambas sesiones. **El cliente o el servidor pueden iniciar la terminación**.

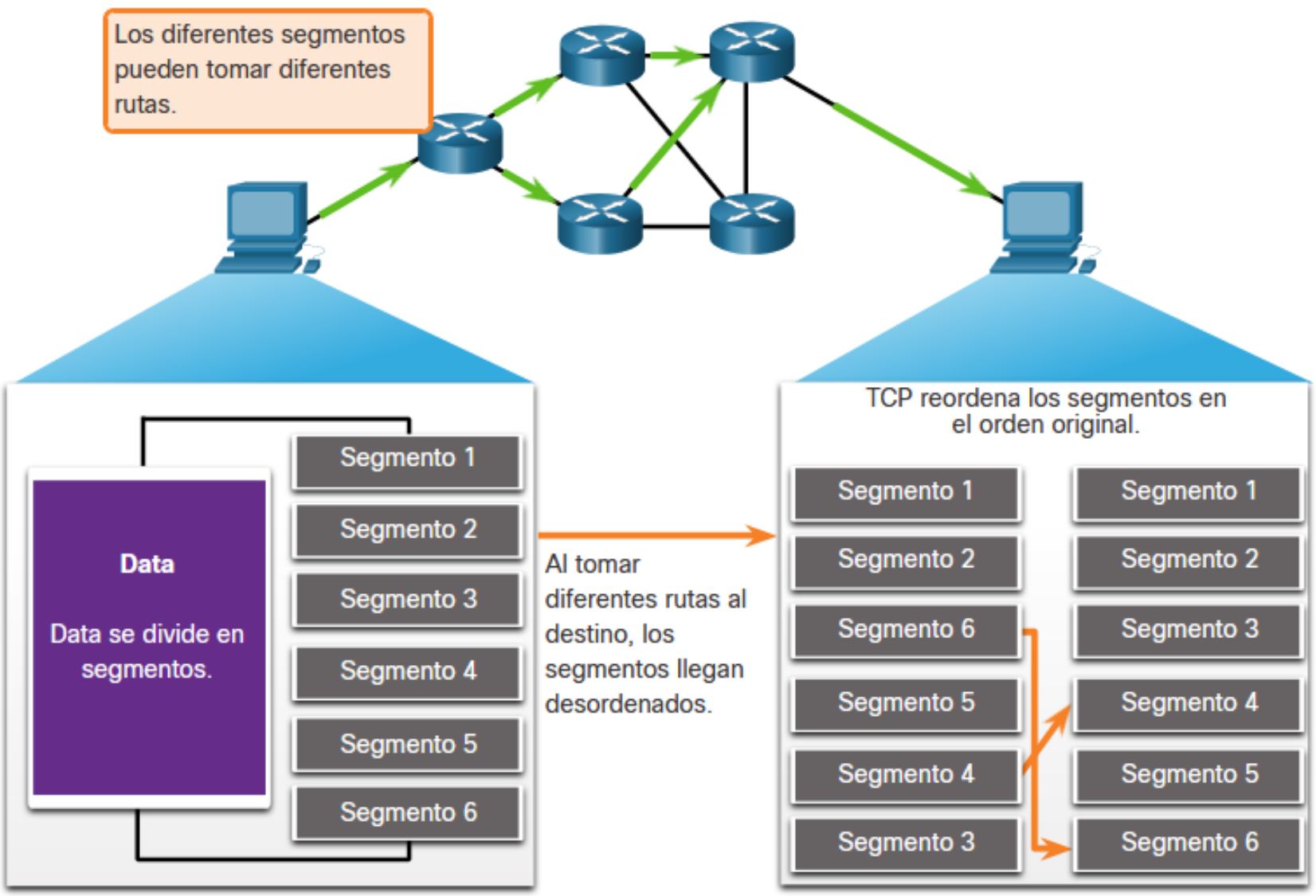


Fiabilidad de TCP: Entrega garantizada y ordenada

TCP reenvía paquetes descartados y paquetes numerados para indicar su **orden correcto** antes de la entrega. TCP **también puede ayudar a mantener el flujo de paquetes para que los dispositivos no se sobrecarguen.**

Durante la configuración de la sesión, se establece un número de secuencia inicial (ISN). Este ISN representa el valor inicial de los bytes que se transmiten a la aplicación receptora. A medida que se transmiten los datos durante la sesión, el **número de secuencia** se incrementa según el número de bytes que se han transmitido. Este **seguimiento de bytes de datos** permite identificar y reconocer cada segmento de manera exclusiva. A partir de esto, **se pueden identificar segmentos perdidos.**

Fiabilidad de TCP: Entrega garantizada y ordenada



Fiabilidad de TCP: Entrega garantizada y ordenada

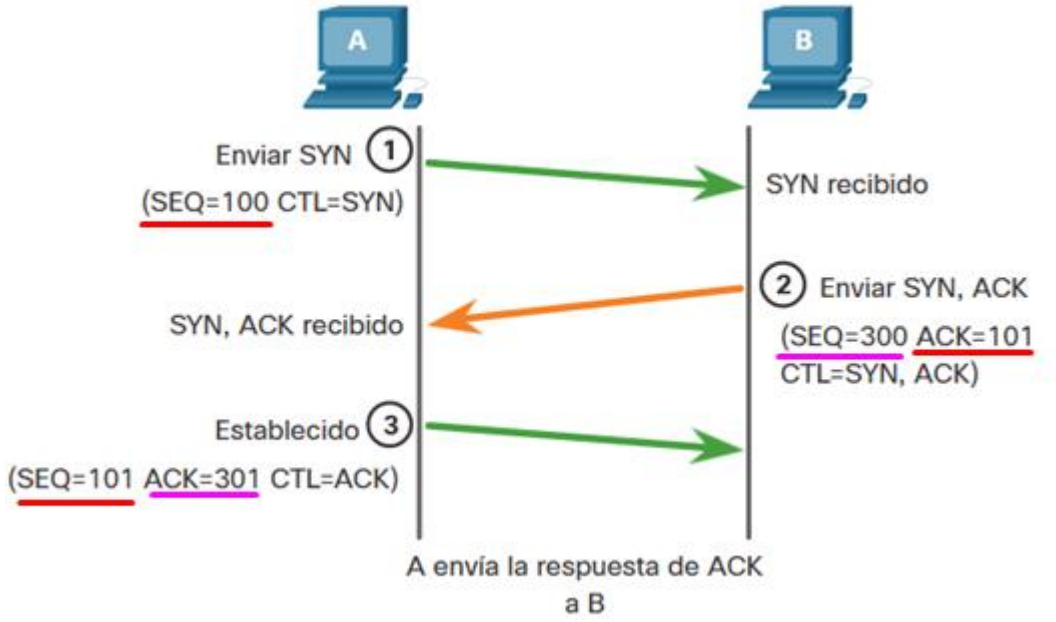
El ISN no comienza en uno, pero es efectivamente un número aleatorio. Esto permite evitar ciertos tipos de ataques maliciosos.

Los números de secuencia de segmento indican cómo reensamblar y reordenar los segmentos recibidos.

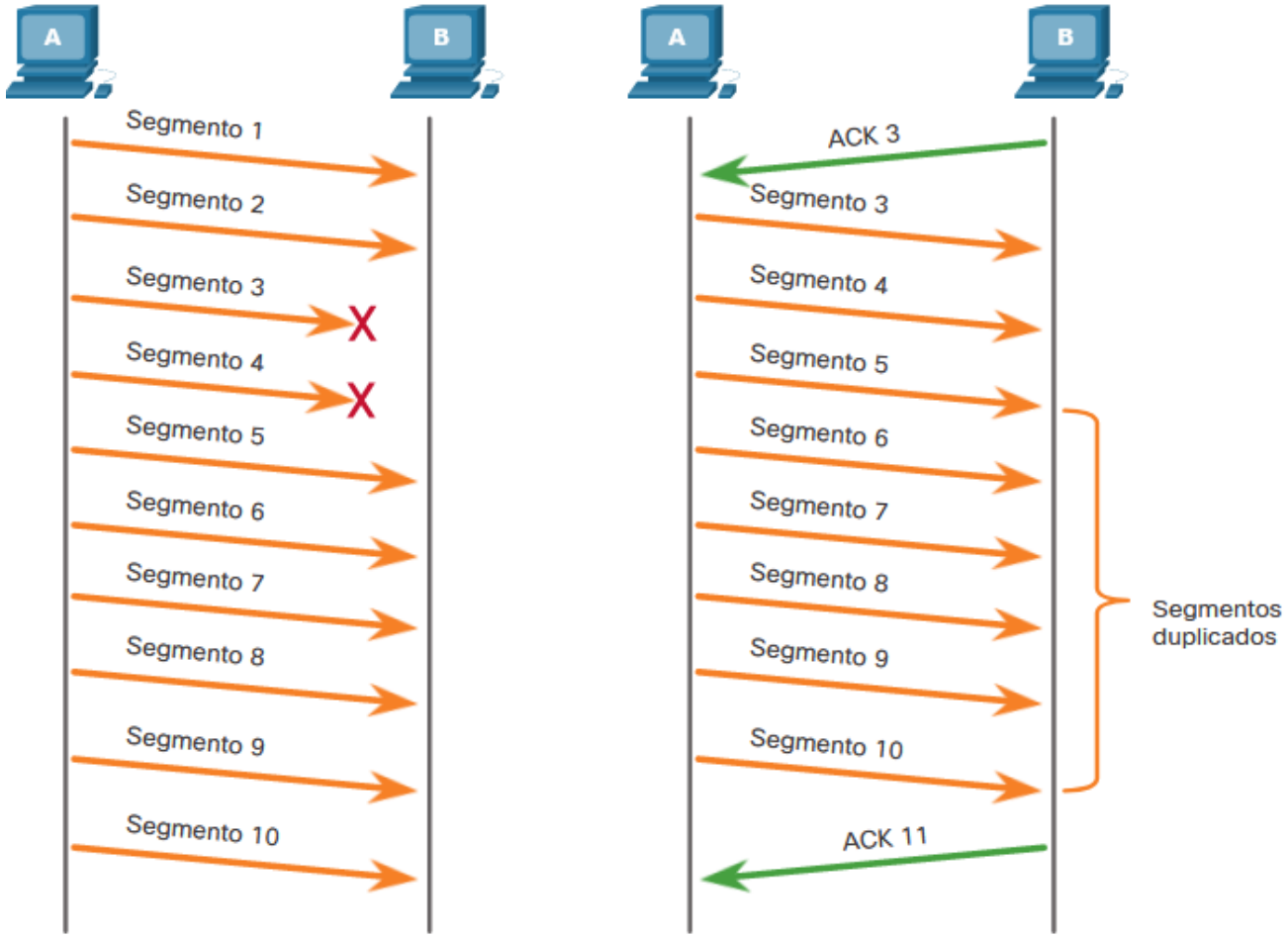
El proceso TCP receptor coloca los datos del segmento en un búfer de recepción. Los segmentos se colocan en el orden de secuencia adecuado y se pasan a la capa de aplicación cuando se vuelven a montar. Todos los segmentos que lleguen con números de secuencia desordenados se retienen para su posterior procesamiento. A continuación, cuando llegan los segmentos con bytes faltantes, tales segmentos se procesan en orden.

Fiabilidad de TCP: Pérdida y retransmisión de datos

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 SEQ identifica el primer byte de datos en el segmento que se transmite. TCP utiliza el número de ACK reenviado al origen para indicar el próximo byte que el receptor espera recibir.



Fiabilidad de TCP: Pérdida y retransmisión de datos

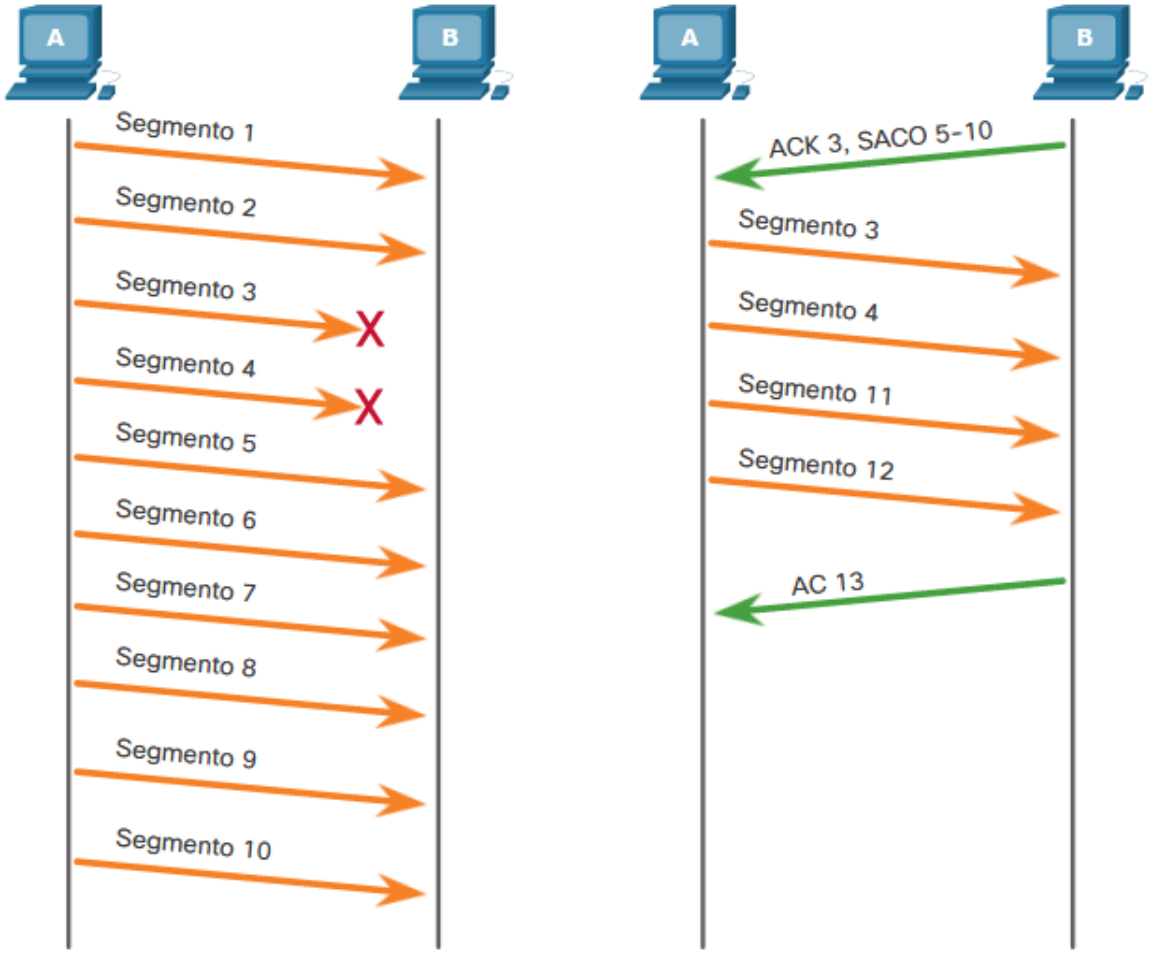


Nota: Los números de acuse de recibo corresponden al siguiente byte esperado y no a un segmento. Los números de segmentos utilizados se simplifican con fines ilustrativos.

Fiabilidad de TCP: Pérdida y retransmisión de datos

Los sistemas operativos actualmente suelen emplear una característica TCP opcional llamada reconocimiento selectivo (SACK), negociada durante el protocolo de enlace de tres vías. Si ambos hosts admiten SACK, el receptor puede reconocer explícitamente qué segmentos (bytes) se recibieron, incluidos los segmentos discontinuos. Por lo tanto, **el host emisor solo necesitaría retransmitir los datos faltantes.**

Fiabilidad de TCP: Pérdida y retransmisión de datos

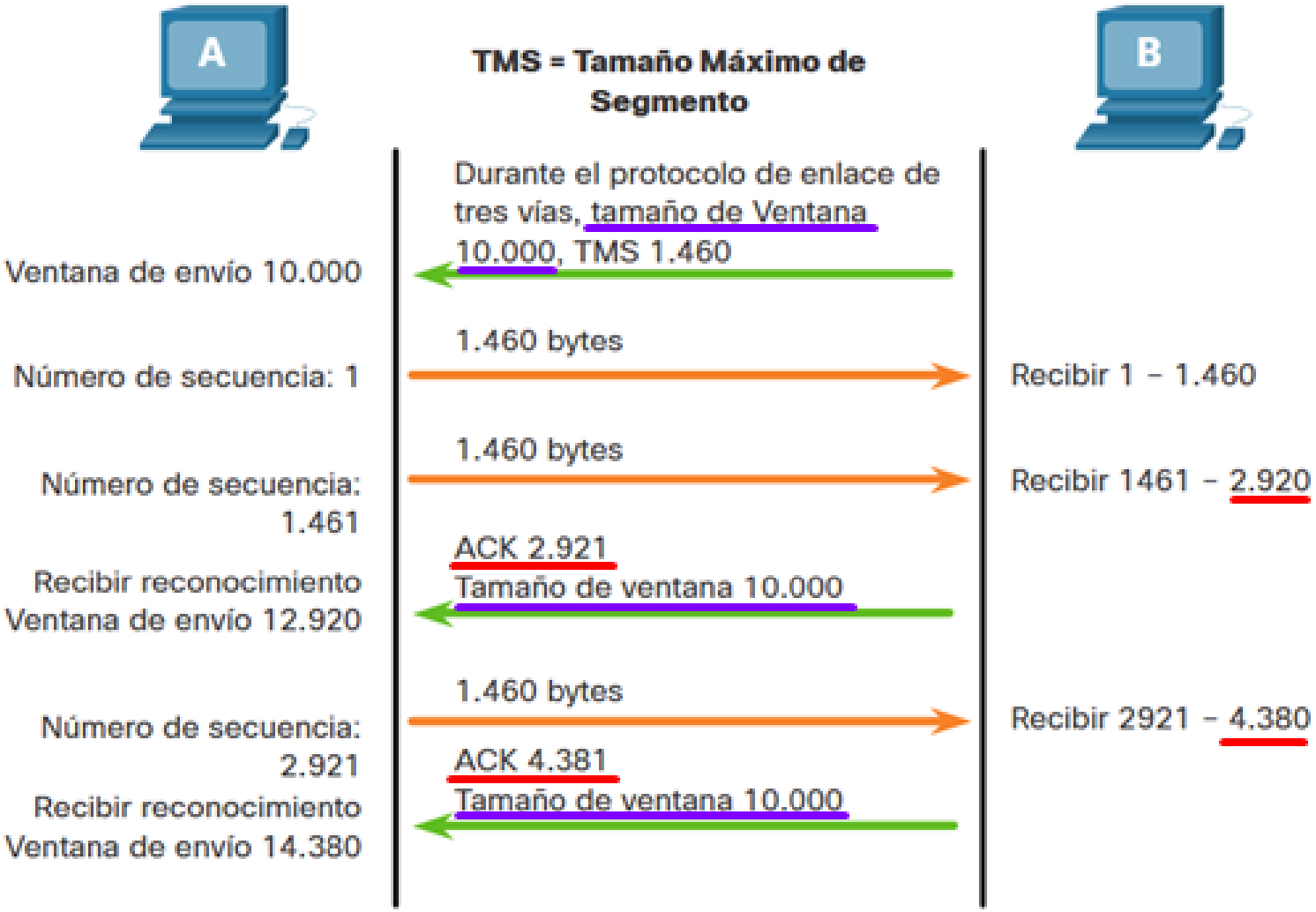


Nota: Los números de acuse de recibo corresponden al siguiente byte esperado y no a un segmento. Los números de segmentos utilizados se simplifican con fines ilustrativos.

Control de Flujo de TCP: Tamaño de la Ventana y Reconocimientos

TCP también proporciona mecanismos para el control de flujo. El control de flujo es la cantidad de datos que el destino puede recibir y procesar de manera confiable. El control de 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.** Para lograr esto, el encabezado TCP incluye un campo de 16 bits llamado **"tamaño de la ventana".**

Control de Flujo de TCP: Tamaño de la Ventana y Reconocimientos



Control de Flujo de TCP: Tamaño de la Ventana y Reconocimientos

El tamaño de la ventana determina la cantidad de bytes que se pueden enviar para recibir un reconocimiento (ACK). El número de ACK es el número del siguiente byte esperado.

El tamaño de ventana es la cantidad de bytes que el dispositivo de destino de una sesión TCP puede aceptar y procesar al mismo tiempo. En este ejemplo, el tamaño de la ventana inicial de la PC B para la sesión TCP es de 10,000 bytes. A partir del primer byte, el byte1, el último byte que la PC A puede enviar sin recibir un reconocimiento es el byte 10000. Esto se conoce como la ventana de envío de la PC A. El tamaño de la ventana se incluye en cada segmento TCP para que el destino pueda modificar el tamaño de la ventana en cualquier momento **dependiendo de la disponibilidad del búfer.**

Control de Flujo de TCP: Tamaño de la Ventana y Reconocimientos

El tamaño inicial de la ventana se acuerda cuando se establece la sesión TCP durante la realización del enlace de tres vías. **El dispositivo de origen debe limitar el número de bytes enviados al dispositivo de destino en función del tamaño de la ventana del destino.** El dispositivo de origen puede continuar enviando más datos para la sesión solo cuando obtiene un reconocimiento de los bytes recibidos. Por lo general, el destino no esperará que se reciban todos los bytes de su tamaño de ventana antes de contestar con un acuse de recibo. A medida que se reciben y procesan los bytes, el destino envía reconocimientos para informar al origen que puede continuar enviando bytes adicionales.

Control de Flujo de TCP: Tamaño de la Ventana y Reconocimientos

Un destino que envía confirmaciones a medida que procesa los bytes recibidos, y el ajuste continuo de la ventana de envío de origen, se conoce como ventanas deslizantes. En el ejemplo anterior, la ventana de envío del PC A aumenta o se desliza sobre otros 2.921 bytes de 10.000 a 12.920.

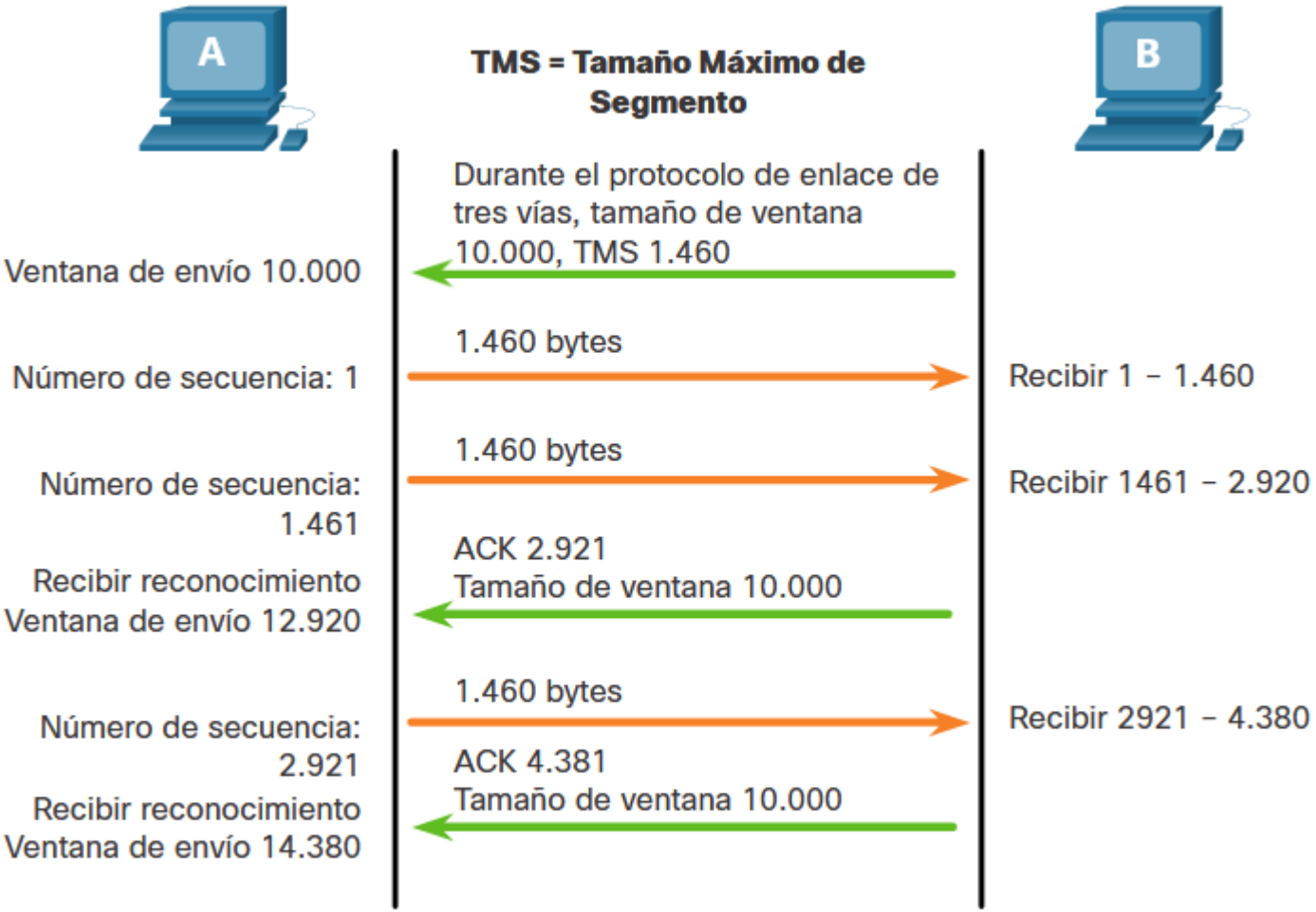
Si disminuye la disponibilidad de espacio de búfer del destino, puede reducir su tamaño de ventana para informar al origen que reduzca el número de bytes que debe enviar sin recibir un reconocimiento.

La ventaja de las ventanas deslizantes es que permiten que el emisor transmita continuamente segmentos mientras el receptor está acusando recibo de los segmentos anteriores.

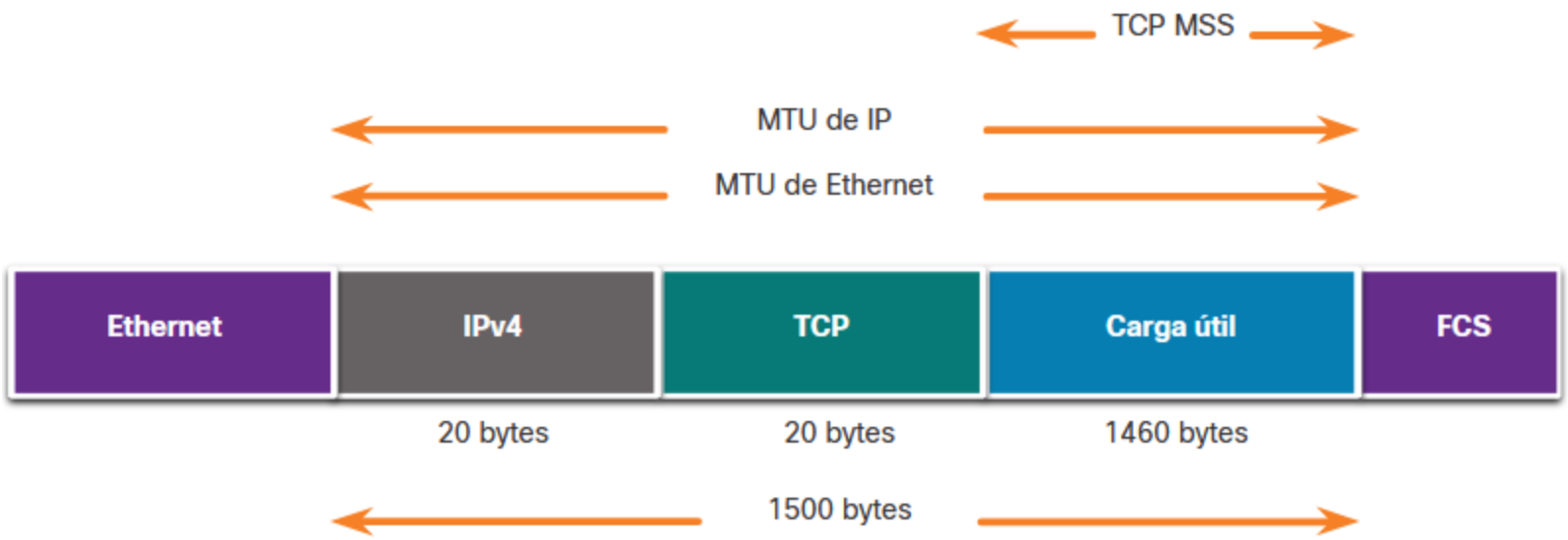
Control de Flujo TCP - Tamaño Máximo de Segmento (MSS)

En la figura, la fuente está transmitiendo **1.460 bytes** de datos dentro de cada segmento TCP. Normalmente, es el Tamaño Máximo de Segmento (**MSS**) que puede recibir el dispositivo de destino. **El MSS forma parte del campo de opciones del encabezado TCP que especifica la mayor cantidad de datos, en bytes, que un dispositivo puede recibir en un único segmento TCP. El tamaño MSS no incluye el encabezado TCP. El MSS se incluye normalmente durante el apretón de manos de tres vías.**

Control de Flujo TCP - Tamaño Máximo de Segmento (MSS)



Control de Flujo TCP - Tamaño Máximo de Segmento (MSS)



Control de flujo de TCP: Prevención de Congestiones

Cuando se produce congestión en una red, el router sobrecargado comienza a **descartar paquetes**. Cuando los paquetes que contienen **segmentos TCP** no llegan a su destino, **se dejan sin confirmar.** Mediante la determinación de la tasa a la que se envían pero no se reconocen los segmentos TCP, **el origen puede asumir un cierto nivel de congestión de la red.**

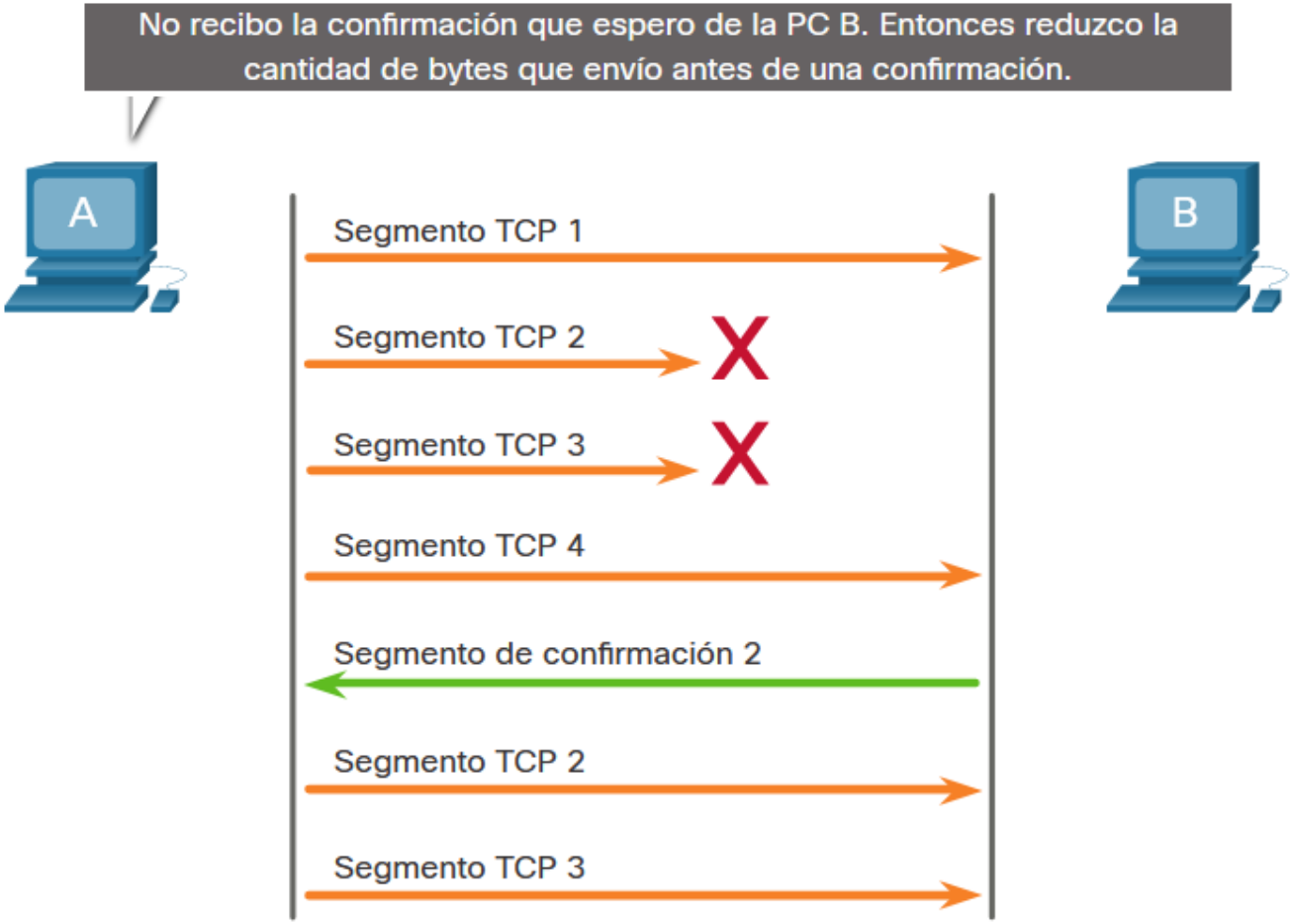
Control de flujo de TCP: Prevención de Congestiones

Siempre que haya congestión, se producirá la retransmisión de los segmentos TCP perdidos del origen. Si la retransmisión no se controla adecuadamente, la retransmisión adicional de los segmentos TCP puede empeorar aún más la congestión. No sólo se introducen en la red los nuevos paquetes con segmentos TCP, sino que **el efecto de retroalimentación de los segmentos TCP retransmitidos que se perdieron también se sumará a la congestión. Para evitar y controlar la congestión, TCP emplea varios mecanismos, temporizadores y algoritmos de manejo de la congestión.**

Control de flujo de TCP: Prevención de Congestiones

Si el origen determina que los segmentos TCP no están siendo reconocidos o que sí son reconocidos pero no de una manera oportuna, entonces puede reducir el número de bytes que envía antes de recibir un reconocimiento.

Control de flujo de TCP: Prevención de Congestiones

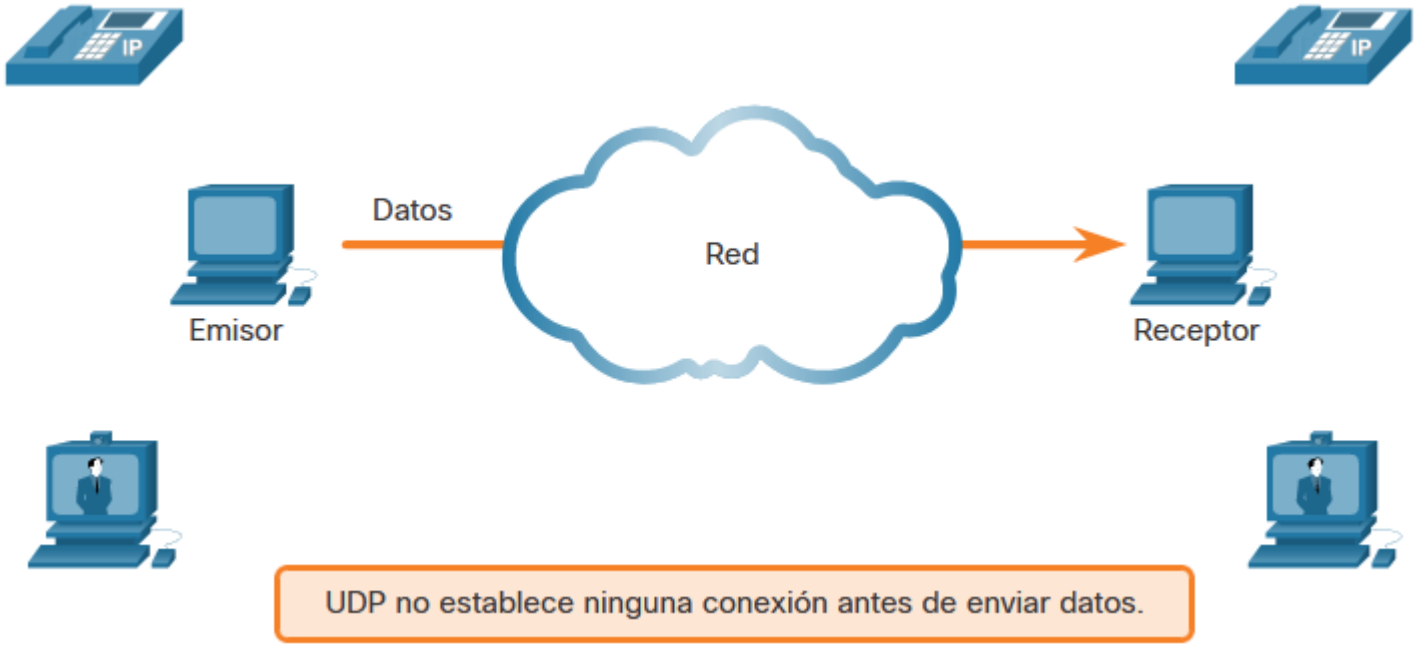


Nota: Los números de acuse de recibo corresponden al siguiente byte esperado y no a un segmento. Los números de segmentos utilizados se simplifican con fines ilustrativos.

6 Comunicación UDP

Comunicación UDP

UDP es perfecto para comunicaciones que necesitan ser rápidas, como VoIP. UDP no establece una conexión. UDP suministra transporte de datos con baja sobrecarga debido a que posee un encabezado de datagrama pequeño sin tráfico de administración de red.

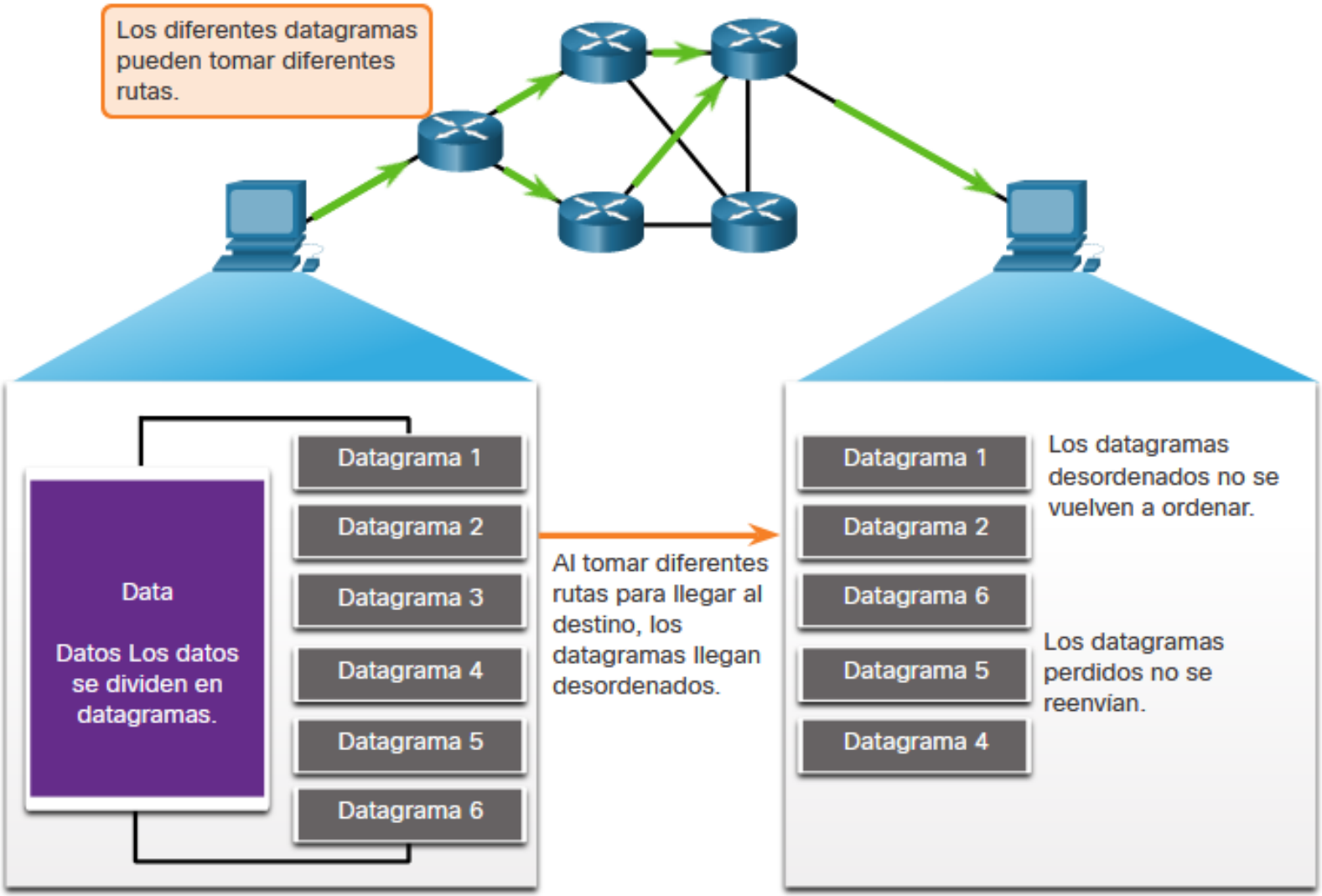


Reensamblaje de datagaramas UDP

Tal como los segmentos con TCP, cuando se envían **datagramas UDP** a un destino, **a menudo toman diferentes rutas y llegan 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.**

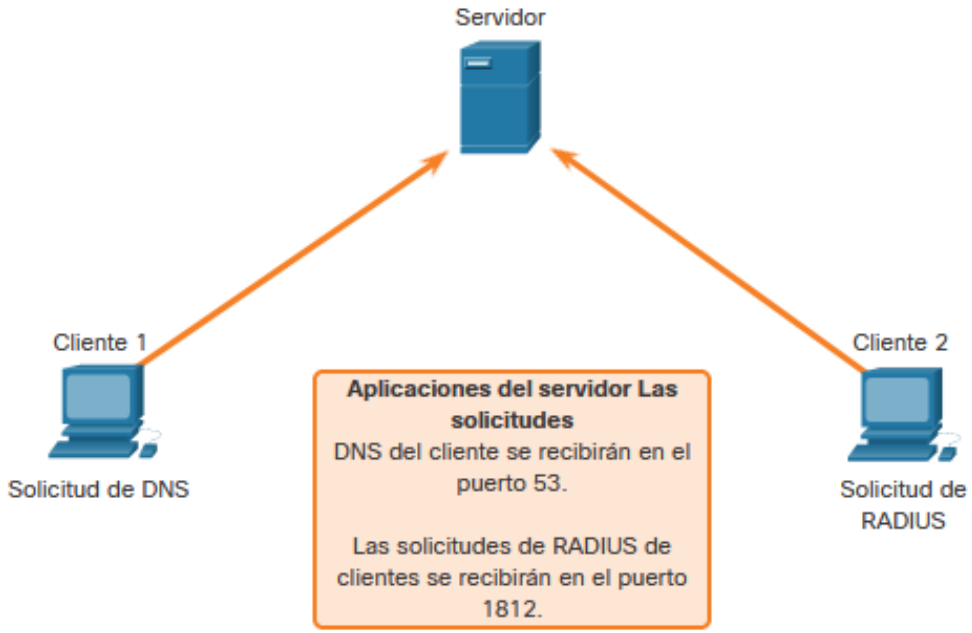
Por lo tanto, UDP simplemente reensambla los datos en el orden en que se recibieron y los envía a la aplicación. Si la secuencia de datos es importante para la aplicación, esta debe identificar la secuencia adecuada y determinar cómo se deben procesar los datos.

Reensamblaje de datagaramas UDP



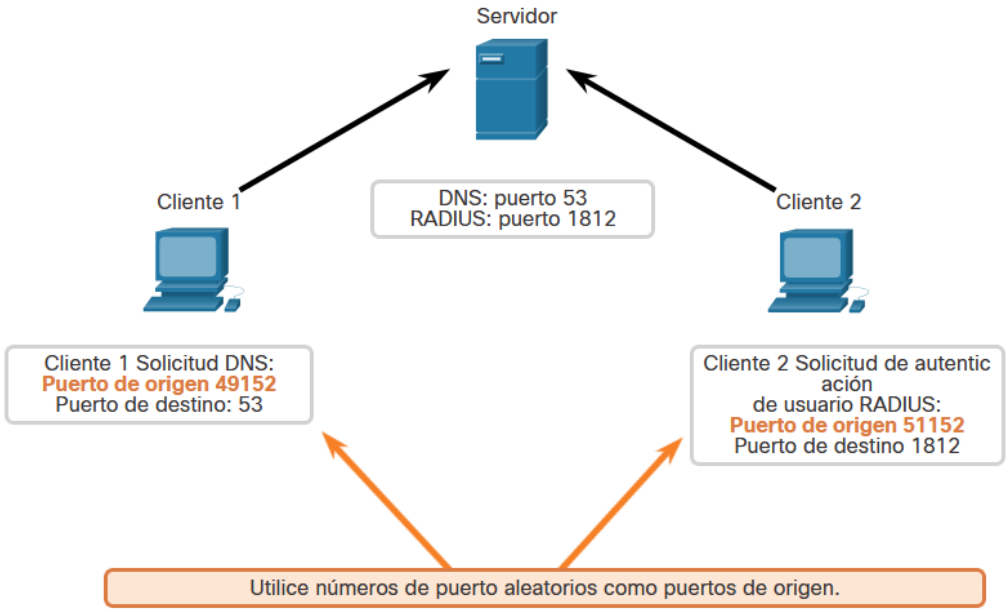
Procesos y solicitudes del servidor UDP

Al igual que las aplicaciones basadas en TCP, a las aplicaciones de servidor basadas en UDP se les asignan números de puerto 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.



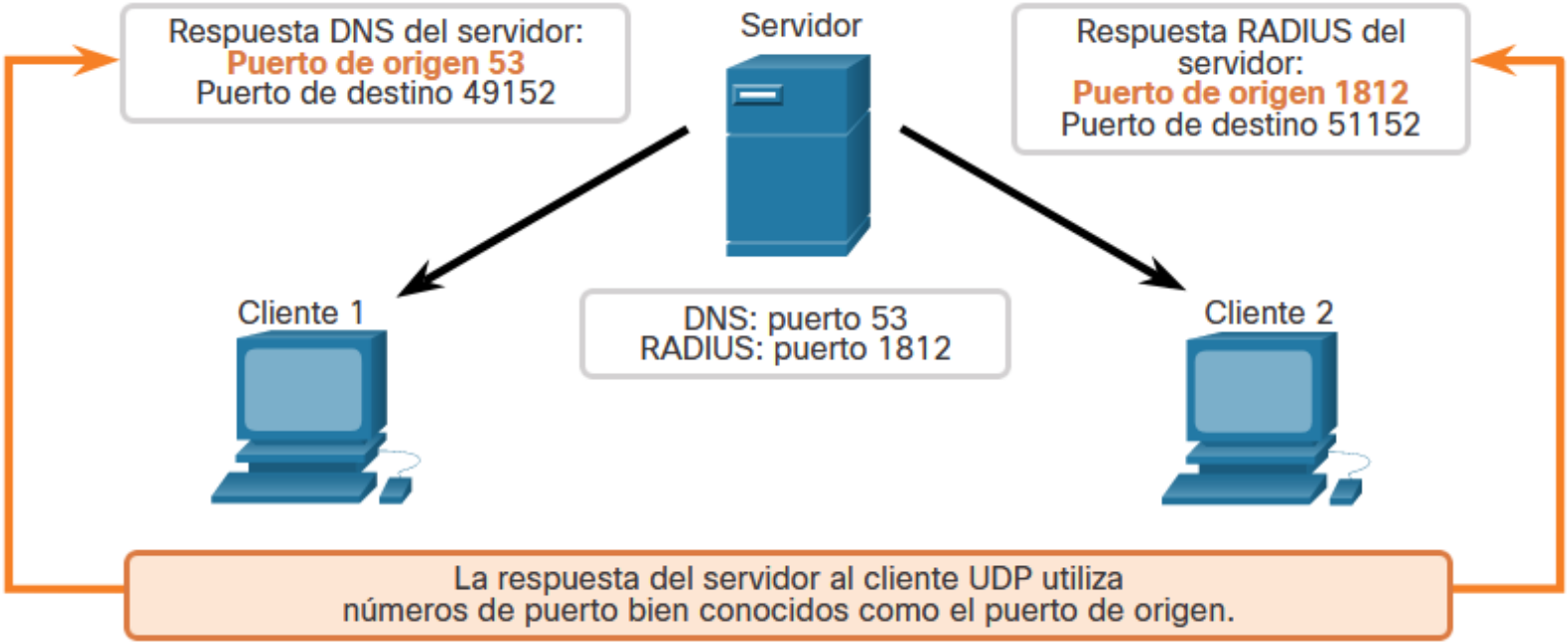
Procesos de cliente UDP

Como en TCP, la comunicación cliente-servidor es iniciada por una aplicación cliente que solicita datos de un proceso de servidor. El proceso de cliente UDP selecciona dinámicamente un número de puerto del intervalo de números de puerto 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.



Procesos de cliente UDP

Después de que un cliente ha seleccionado los puertos de origen y destino, se utiliza el mismo par de puertos en el encabezado de todos los datagramas 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.



7 TCP vs UDP

TCP vs UDP

Algunas **aplicaciones pueden tolerar cierta pérdida de datos durante la transmisión a través de la red**, pero los retrasos en la transmisión son inaceptables. Para estas aplicaciones, **UDP** es la mejor opción porque requiere menos sobrecarga de red. UDP es preferible para aplicaciones como Voz sobre IP (VoIP). Los reconocimientos y la retransmisión retrasarían la entrega y harían inaceptable la conversación de voz.

UDP también es utilizado por las aplicaciones de solicitud y respuesta donde los datos son mínimos, y la retransmisión se puede hacer rápidamente. Por ejemplo, el servicio de nombres de dominio (DNS) utiliza UDP para este tipo de transacción. El cliente solicita direcciones IPv4 e IPv6 para obtener un nombre de dominio conocido desde un servidor DNS. Si el cliente no recibe una respuesta en un período de tiempo predeterminado, simplemente envía la solicitud de nuevo.

TCP vs UDP

Por ejemplo, si uno o dos segmentos de una transmisión de vídeo en vivo no llegan al destino, se interrumpe momentáneamente la transmisión. Esto puede manifestarse como distorsión en la imagen o el sonido, pero puede no ser perceptible para el usuario. Si el dispositivo de destino tuviera que dar cuenta de los datos perdidos, la transmisión se podría demorar mientras espera las retransmisiones, lo que ocasionaría que la imagen o el sonido se degraden considerablemente. En este caso, es mejor producir el mejor vídeo o audio posible con los segmentos recibidos y prescindir de la confiabilidad.

TCP vs UDP

Para otras aplicaciones es importante que todos los datos lleguen y que puedan ser procesados en su secuencia adecuada. Para estos tipos de aplicaciones, TCP se utiliza como protocolo de transporte.
Por ejemplo, las aplicaciones como las bases de datos, los navegadores web y los clientes de correo electrónico, requieren que todos los datos que se envían lleguen a destino en su formato original. Cualquier dato faltante podría corromper una comunicación, haciéndola incompleta o ilegible. Por ejemplo, cuando se accede a la información bancaria a través de la web, es importante asegurarse de que toda la información se envía y recibe correctamente.

TCP vs UDP

UDP

VoIP
(telefonía IP)

DNS
(Domain Nameresolución de nombre de dominio Resolution)

Propiedades de protocolo requeridas:

- Rápido
- Baja sobrecarga
- No requiere reconocimiento
- No reenvía los datos perdidos
- Entrega los datos a medida que llegan

TCP

SMTP/IMAP
(Correo electrónico)

HTTP/HTTPS
(World Wide Web)

Propiedades de protocolo requeridas:

- Confiable
- Reconoce los datos
- Reenvía los datos perdidos
- Entrega los datos en orden secuencial

Gracias por la atenshion!