**TCP y UDP**

**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.

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.

**Confirmación de recepción de segmentos.**

Una de las funciones de TCP es garantizar que cada segmento llegue a destino.

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.

La cantidad de datos que un origen puede transmitir antes de recibir un acuse de recibo se denomina “tamaño de la ventana” (es un campo de 16 bits), que es un campo en el encabezado TCP que permite administrar datos perdidos y controlar el flujo.

**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.

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.

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.

La retransmisión de un segmento de origen a destino lleva el último numero de ACK recibido y el numero de solicitud es el mismo al de acuse de recibo ACK.

**Control de 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.

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.

Durante el retraso en la recepción del acuse de recibo, el emisor no envía ningún otro segmento.

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.

El tamaño de la ventana determina la cantidad de bytes enviados antes de que se espere recibir un acuse de recibo. El numero de acuse de recibo es el numero del siguiente byte previsto.

**Reducción del tamaño de la ventana.**

Otra forma de controlar el flujo de datos es utilizar tamaños de ventana dinámicos. Se usa cuando los recursos de la red son limitados.

Esto reduce de forma efectiva la velocidad de transmisión porque el origen espera que se dé 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.

Cuanto mayor es el campo de la ventana, menos acuses de recibo hay, y se reduce también la sobrecarga en la red.

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.

**Alternativas actuales al protocolo TCP.**

Giga es un sustituto para TCP, un protocolo para la transferencia de datos.

Giga incluye varias mejoras respecto a TCP que pueden exprimir un mayor rendimiento de las conexiones existentes de internet. Una de ellas detecta mejor si la ruta hasta una persona que intenta acceder a unos datos ya está funcionando a máxima capacidad o no.

Otro truco que incorpora Giga para impulsar la velocidad codifica los datos en paquetes electrónicos de una forma nueva que reduce la tasa de enlaces caídos en caso de flaquear la conexión.

La empresa la quiere combinar con QUIC, un software desarrollado por Google, que puede lograr que algunos tipos de tráfico de internet, como el streaming de vídeo, sean más rápidos.

**PROTOCOLO DE DATAGRAMA DE USUARIO (UDP)**

UDP se considera un protocolo de transporte de máximo esfuerzo. 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. Sus características:

* Sin conexión.
* Entrega no confiable.
* Reconstrucción de datos no ordenada.
* Sin control de flujo.

Aunque UDP no incluye la confiabilidad y los mecanismos de control del flujo de TCP, 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.

Algunas aplicaciones que utilizan UDP son el Sistema de nombres de dominios (DNS), el streaming de video y la voz sobre IP (VoIP).

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.

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, la debe administrar la aplicación.

UDP es un protocolo sin conexión.

**COMUNICACIÓN UDP**

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. Quiere decir que estas funciones no las proporciona el protocolo de la capa de transporte, y se deben implementar aparte, si fuera necesario.

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.

La baja sobrecarga del UDP es deseada por dichas aplicaciones.

Se dice que UDP está basado en las transacciones; es decir, cuando una aplicación tiene datos para enviar, simplemente los envía.

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.

Todo lo relacionado a los protocolos TCP y UDP me ha parecido importante.

Me ha parecido interesante la alternativa actual al protocolo TCP, llamado Giga.

Me ha quedado menos claro:

El proceso del acuse de recibo de los segmentos TCP.

La gestión de la conexión TCP.

He comprendido el funcionamiento de la capa TCP (a la hora de ordenar los segmentos, manejar los segmentos perdidos, ...) y UDP.