Práctica 5

1. ¿ Cuál es la función de la capa de transporte?

La función principal de la capa de transporte es proporcionar una comunicación lógica (desde la perspectiva de la aplicación, es como si los hosts que ejecutan los procesos estuvieran conectados directamente) entre procesos de aplicación que se ejecutan en dispositivos diferentes dentro de una red. Esta comunicación lógica permite que los procesos de aplicación se envíen mensajes entre sí sin preocuparse por los detalles de la infraestructura física subyacente.

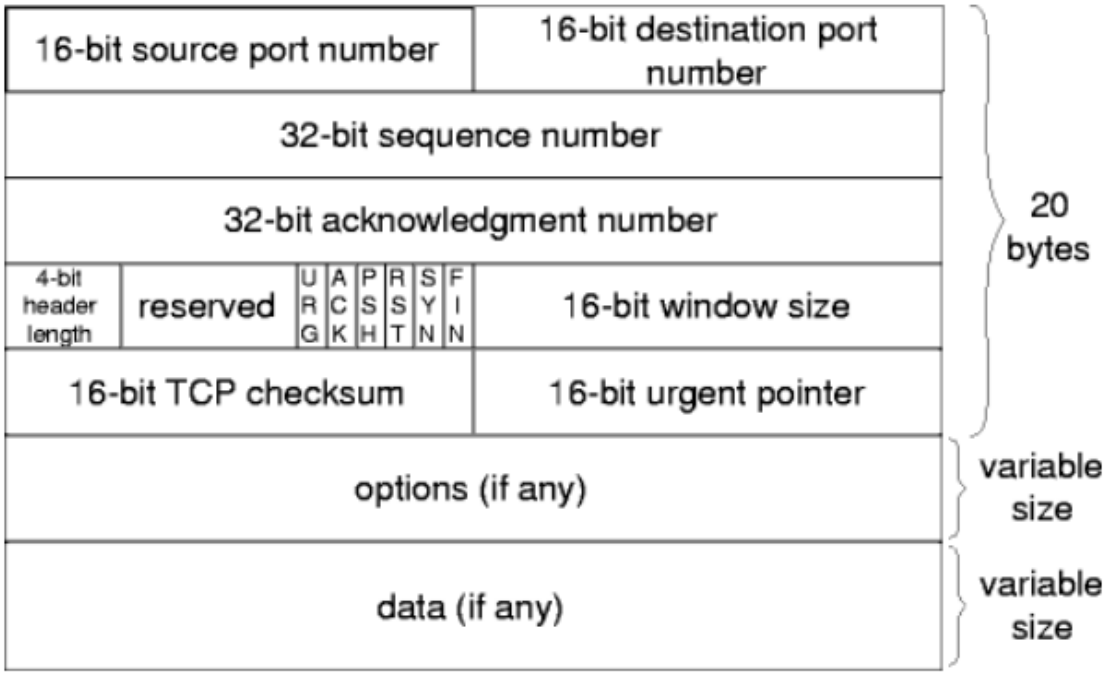
Los protocolos de la capa de transporte se implementan en los sistemas terminales (hosts), y no en los routers de la red. En el lado emisor, la capa de transporte convierte los mensajes de la aplicación en segmentos de la capa de transporte, que luego se encapsulan en paquetes de la capa de red y se envían al destino. Los routers de la red solo actúan sobre los campos correspondientes a la capa de red del paquete, sin examinar los campos del segmento de la capa de transporte encapsulado.

En el lado receptor, la capa de transporte extrae el segmento de la capa de transporte del paquete y lo entrega a la aplicación receptora. Para las aplicaciones de red, existen varios protocolos de la capa de transporte disponibles, como TCP y UDP, cada uno ofreciendo un conjunto diferente de servicios a las aplicaciones que los utilizan.

2. Describa la estructura del segmento TCP y UDP.

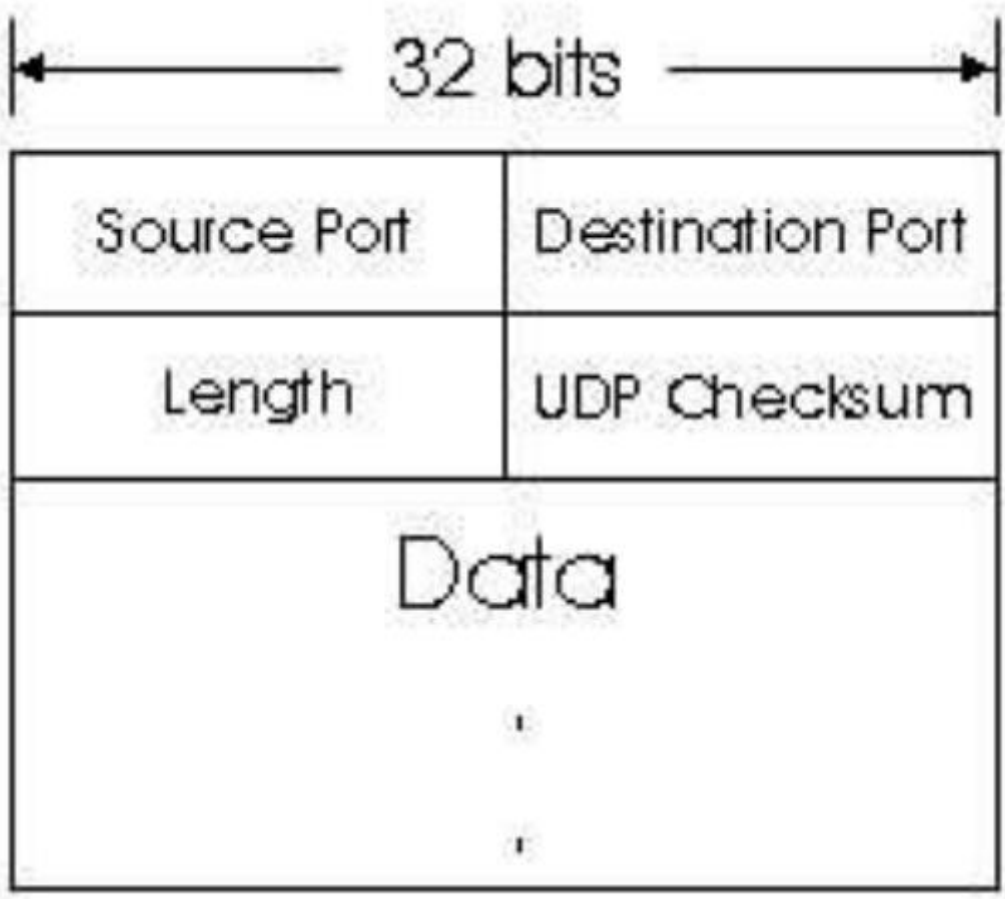
TCP	
16-bit source port number	Indica el número de puerto de origen de la aplicación que está enviando los datos.
16-bit destination port number	Especifica el número de puerto de destino de la aplicación receptora.
32-bit sequence number	Este campo se utiliza para mantener un seguimiento del orden de los segmentos TCP en una comunicación. Cada segmento TCP se etiqueta con un número de secuencia único.
32-bit acknowledgment number	Indica el número de secuencia que espera recibir el emisor del siguiente segmento. Ayuda a establecer que los datos se han recibido de manera confiable.
4-bit header length	Este campo especifica la longitud del encabezado TCP en palabras de 32 bits. Se utiliza para identificar dónde comienza la carga útil de datos en el segmento.

reserved	Este campo se reserva para uso futuro y debe establecerse en cero.
16-bit widow size	Indica el tamaño de la ventana de recepción que el receptor tiene disponible para aceptar datos. Ayuda a controlar el flujo de datos en la conexión.
16-bit TCP checksum	Proporciona una suma de verificación para verificar la integridad de los datos en el segmento TCP y detectar errores de transmisión.
16-bit urgent pointer	Se utiliza en la comunicación para indicar la posición de datos urgentes dentro del segmento, si es necesario.
options	Este campo opcional permite la inclusión de información adicional en el encabezado TCP, como máximas segment size (MSS), ventana de escala, timestamp, entre otros. Las opciones se utilizan para ajustar y optimizar la comunicación según las necesidades de la aplicación.



UDP	
Source Port	Especifica el número de puerto del proceso que envía el datagrama UDP
Destination Port	Este campo indica el número de puerto del proceso de destino al que se debe entregar el datagrama UDP
Length	Este campo especifica la longitud total del datagrama UDP, incluyendo tanto el encabezado como los datos. La longitud se mide en bytes y permite al receptor conocer

	la cantidad de información que debe procesar en el datagrama.
UDP Checksum	La suma de verificación es un valor calculado que se utiliza para detectar errores en el datagrama UDP durante la transmisión. Se calcula en función del contenido del datagrama, incluyendo el encabezado y los datos. El receptor verifica esta suma de verificación para determinar si el datagrama UDP ha llegado intacto o si ha sufrido algún tipo de corrupción durante la transmisión.



3. **¿Cuál es el objetivo del uso de puertos en el modelo TCP/IP?**

   Se utilizan para distinguir las aplicaciones (y, por lo tanto, protocolos) que están enviando/recibiendo datos. Los puertos actúan como puntos finales en una comunicación y permiten que múltiples aplicaciones en una misma computadora o dispositivo se comuniquen simultáneamente a través de la red.

4. **Compare TCP y UDP en cuanto a:**

   a. **Confiabilidad.**

TCP	UDP
Es un modelo confiable ya que garantiza que los datos se entreguen en el orden correcto y sin errores a través de técnicas como la retransmisión de datos perdidos y la detección y corrección de errores utilizando sumas de verificación.	UDP es menos confiable en comparación con TCP. No garantiza la entrega de datos ni el orden de entrega. Los segmentos UDP pueden perderse o llegar desordenados sin corrección automática.

b. Multiplexación.

Un proceso puede tener uno o varios sockets, por tanto la capa de transporte entrega los datos al socket (y no directamente a la aplicación), para identificar los sockets, éstos tienen un identificador único. Cada segmento de la capa de transporte contiene un campo para poder entregar los datos al socket adecuado. En el receptor, la capa de transporte examina estos campos para identificar el socket receptor y lo envía (demultiplexación).

Denominamos multiplexación al trabajo de reunir los datos en el host origen desde diferentes sockets, encapsulando los fragmentos de datos con la información de cabecera (que se usará en la demultiplexación).

TCP	UDP
Multiplexación y demultiplexación orientada a la conexion	Multiplexación y demultiplexación sin conexión
El socket TCP queda identificado por una tupla de cuatro elementos: dirección IP de origen, número de puerto de origen, dirección IP de destino, número de puerto de destino. Por lo tanto, cuando un segmento TCP llega a un host procedente de la red, el host emplea los cuatro valores para dirigir (demultiplexar) el segmento al socket apropiado.	El socket UDP queda completamente identificado por una tupla que consta de una dirección IP de destino y un número de puerto de destino. En consecuencia, si dos segmentos UDP tienen diferentes direcciones IP y/o números de puerto de origen, pero la misma dirección IP de destino y el mismo número puerto de destino, entonces los dos segmentos se enviarán al mismo proceso de destino a través del mismo socket de destino.

c. Orientado a la conexión.

TCP	UDP
Establece una conexión antes de la transmisión de datos y asegura que ambas partes estén sincronizadas en términos de secuencia de datos y control de flujo.	No se necesita conexión para iniciar y finalizar una transferencia de datos

d. Controles de congestión.

TCP	UDP
Proporciona mecanismos de control de congestión. Los mecanismos de control de congestión de TCP evitan que cualquier conexión TCP inunde con una cantidad de tráfico excesiva los enlaces y routers existentes entre los hosts que están comunicándose. Esto se consigue regulando la velocidad a la que los lados emisores de las conexiones TCP pueden enviar tráfico a la red.	El tráfico UDP no está regulado. Una aplicación que emplee el protocolo de transporte UDP puede enviar los datos a la velocidad que le parezca, durante todo el tiempo que quiera.
Posee un mecanismo que indica al emisor cuánto espacio libre hay en el búfer de almacenamiento del receptor (ventana de recepción). Ayuda a controlar el flujo de datos para evitar la congestión y garantizar una comunicación eficiente, permitiendo que el emisor ajuste la cantidad de datos enviados en función de la capacidad disponible en el receptor.	

e. Utilización de puertos.

TCP	UDP
Como está orientado a la conexión, establece una conexión punto a punto entre dos dispositivos, por lo que cada conexión está limitada a dos procesos que intercambian datos.	Permite que muchos clientes o procesos envíen datos por el mismo socket
Utiliza números de puerto para identificar aplicaciones específicas.	Utiliza números de puerto para identificar aplicaciones específicas.

5. La PDU de la capa de transporte es el segmento. Sin embargo, en algunos contextos suele utilizarse el término datagrama. Indique cuando

Cuando se trata del protocolo UDP, el termino datagrama se utiliza para su PDU.

6. Describa el saludo de tres vías de TCP. ¿Se utiliza algo similar en UDP?

También conocido como protocolo de enlace de TCP, es un método utilizado por TCP para establecer una conexión confiable entre dos dispositivos en una red. Es un método de tres pasos que requiere que tanto el cliente como el servidor intercambien segmentos SYN y ACK antes de que comience la comunicación de datos real.

Paso 1 (SYN) – El cliente inicia el proceso enviando un segmento al servidor con el bit SYN establecido y un número de secuencia inicial (ISN) generado de manera pseudoaleatoria. Este es importante para identificar y ordenar los datos en la conexión.

Paso 2 (SYN/ACK) – El servidor recibe el segmento del cliente, reconoce el bit SYN y responde enviando un segmento de respuesta con los bits SYN y ACK establecidos. En este, el servidor incluye su propio número de secuencia inicial (ISN), que también es elegido de manera pseudoaleatoria.
El servidor también reconoce el ISN del cliente, lo que indica que ha recibido correctamente el paquete de solicitud de conexión.

Paso 3 (ACK) – El cliente recibe la respuesta del servidor, reconociendo el ISN del servidor. El cliente responde enviando un segmento de confirmación con el bit ACK establecido, confirmando que ha recibido correctamente la respuesta del servidor. Ambos establecen una conexión confiable con la cual iniciarán la transferencia de datos real.

En UPD no se utiliza nada similar ya no se establece ninguna conexion

7. **Investigue qué es el ISN (Initial Sequence Number). Relaciónelo con el saludo de tres vías**

Initial Sequence Number (ISN) se refieren al número de secuencia único de 32 bits asignado a cada nueva conexión en una comunicación TCP. Ayuda con la asignación de un número de secuencia que no entre en conflicto con otros bytes de datos transmitidos a través de una conexión TCP. Un ISN es único para cada conexión y está separado por cada dispositivo. Para el ISN se utiliza un contador que se incrementa cada 4 mseg.

El ISN ayudar a identificar, controlar el origen y mantener el orden de los segmentos de datos transmitidos entre el cliente y el servidor.

8. **Investigue qué es el MSS. ¿Cuándo y cómo se negocia?**

El tamaño de la ventana de recepción TCP es la cantidad de datos de recepción (en bytes) que se pueden almacenar en búfer durante una conexión.

En lugar de usar un tamaño de ventana de recepción predeterminado codificado de forma rígida, TCP se ajusta a incrementos pares del tamaño máximo de segmento (MSS).

Maximum Segment Size es un campo de los encabezados que indica el tamaño más grande de datos que puede tener un segmento sin ser fragmentado. El MSS mide la parte de un paquete que no tiene encabezado, lo

que se conoce como carga útil. El MSS está determinado por otra métrica que tiene que ver con el tamaño de los paquetes: MTU, o la unidad máxima de transmisión, que sí incluye los encabezados TCP e IP (Protocolo de Internet).

El MSS es igual a la MTU menos el tamaño de un encabezado TCP y un encabezado IP:

MTU - (encabezado TCP + encabezado IP) = MSS

Una de las principales diferencias entre la MTU y el MSS es que si un paquete supera la MTU de un dispositivo, se divide en trozos más pequeños, o "se fragmenta." En cambio, si un paquete supera el MSS, se descarta y no se entrega.

El MSS se negocia durante la configuración de la conexión, es decir, durante el saludo de tres vías.