Trabajo Práctico Final Redes

**Catedra:** Redes.

**Profesor:** García Vandenbosch, Leandro.

**Autores:** Díaz, Sergio Ricardo.

Demaria, Felipe.

Murrie Battistoni, Felipe.

1. **¿Qué es un puerto?**

Un puerto es un identificador de virtualización que define un punto final de servicio (a diferencia de un punto final de instancia de servicio también conocido como identificador de sesión).

Los puertos son identificadores de virtualización que definen un punto final de servicio. A modo más coloquial, se puede decir que son compuertas de datos que permiten al software de aplicación dentro de la red usar recursos sin interferencia. Estos puertos son usados por protocolos y normalmente son asociados con una dirección IP para establecer comunicaciones.

1. **¿Cómo están formados los endpoints?**

Un endpoint está compuesto por una dirección de IP y un puerto, que se utilizan para ingresar dentro de una red de comunicación.

1. **¿Qué es un socket?**

Es una interfaz de entrada­salida de datos que permite la intercomunicación entre procesos.

Los procesos pueden estar ejecutándose en el mismo o en distintos sistemas, unidos mediante una red.

A su vez sirve como método para la comunicación entre un programa del cliente y un programa del servidor en una red, se define, por tanto, como el punto final en una conexión.

1. **¿A qué capa del modelo TPC/IP pertenecen los sockets? ¿Porque?**

Los sockets pertenecen a la capa de transporte del modelo TCP/IP debido a que son componentes que permiten justamente efectuar el transporte de los datos independientemente del tipo de red física que se esté utilizando.

Esto se debe a que el modelo TPC/IP es orientado a la conexión y por lo tanto trabaja puertos lógicos y junto con la capa red dan forma a los ya mencionados Sockets.

1. **¿Cómo funciona el modelo cliente-servidor con TCP/IP Sockets?**

Los sockets son una especie de puente virtual que va a comunicar a él o los clientes con el servidor con el fin de poder trasladar a través y a lo largo de ese puente virtual los paquetes de información que irán desde un cliente al servidor o viceversa. Para ellos utilizan flujos de entrada y salida, y hilos que se encargan de estar a la escucha para la recepción de paquetes de información utilizando el puerto.

1. **¿Cuáles son las causas comunes por la que la conexión entre cliente/servidor falle?**

-El servidor no se encuentra habilitado y por esto el cliente no puede conectarse al mismo.

-El puerto de recepción del servidor está bloqueado o cerrado.

-El cliente intenta utilizar un endpoint que se encuentra ya en uso por otro.

-El servidor intenta enviar datos a un cliente que ya se ha desconectado.

-El cliente utiliza una dirección de host y/o un puerto inválido para conectarse al servidor.

1. **Diferencias entre sockets UDP y TCP:**

Ambos tipos tienen sus ventajas y desventajas,UDP es más rápido, simple y eficiente y, por lo tanto, generalmente se utiliza para el envío de archivos de audio, vídeo y TCP, por otro lado, es robusto, fiable y garantiza la entrega de paquetes en el mismo orden.

Sin embargo, aquí detallamos las principales diferencias:

-TCP está orientado a la conexión, mientras que UDP es un protocolo sin conexión.

-TCP es altamente confiable para transferir datos útiles ya que toma el acuse de recibo de la información enviada. Y vuelve a enviar los paquetes perdidos si los hay. Mientras que en el caso de UDP, si el paquete se pierde, no solicitará su retransmisión y el ordenador de destino recibirá un dato corrupto. Por lo tanto, UDP es un protocolo poco fiable.

-TCP es más lento en comparación con UDP, ya que TCP establece la conexión antes de transmitir los datos y garantiza la entrega adecuada de los paquetes. Por otro lado, UDP no reconoce si los datos transmitidos son recibidos o no.

-El tamaño de cabecera de UDP es de 8 bytes, y el de TCP es más del doble. El tamaño de la cabecera TCP es de 20 bytes desde entonces, y la cabecera TCP contiene opciones, relleno, suma de comprobación, banderas, desplazamiento de datos, número de confirmación, número de -secuencia, puertos de origen y destino, etc.

-Tanto TCP como UDP pueden comprobar si hay errores, pero sólo TCP puede corregir el error ya que tiene control de congestión y de flujo.

1. **Diferencia entre sync & async sockets.**

La diferencia primordial entre los sockets síncronos y asíncronos, es que los primeros entablan una comunicación bloqueante debido a que realizan un envió, esperan una respuesta, y recién luego de eso, pueden realizar otro envió de datos. Sin embargo, los sockets asíncronos pueden establecer una comunicación donde no hay ningún factor bloqueante y se desenvuelve de manera desestructurada la emisión y recepción de datos.