Herrada, Emilio Álvaro – Valverde, Gustavo Martín

REDES

1) El modelo OSI es quien se preocupa de la administración de los puertos y los establece en el encabezado de los segmentos es la capa de transporte o capa 4, administrando así el envío y reensamblaje de cada segmento enviado a la red haciendo uso del puerto especificado. Un puerto suele estar numerado para de esta forma poder identificar la aplicación que lo usa. Decidir a qué programa entregará los datos recibidos. Esta asignación de puertos permite a una máquina establecer simultáneamente diversas conexiones con máquinas distintas, ya que todos los segmentos que se reciben tienen la misma dirección, pero van dirigidos a puertos diferentes.

   Puertos

   Los números de puerto se indican mediante una palabra de un procesador de 16 bits (2 bytes), por lo que existen 65536 puertos, numerados del 0 al 65535. Aunque podemos usar cualquiera de ellos para cualquier protocolo, existe una entidad, la IANA, encargada de su asignación, la cual creó tres categorías:

   - Puertos bien conocidos: Los puertos inferiores al 1024 son puertos reservados para el sistema operativo y usados por "protocolos bien conocidos" como por ejemplo HTTP (servidor Web), POP3/SMTP (servidor de e-mail) y Telnet. Si queremos usar uno de estos puertos tendremos que arrancar el servicio que los use teniendo permisos de administrador.
   - Puertos registrados: Los comprendidos entre 1024 (0400 en hexadecimal) y 49151 (BFFF en hexadecimal) son denominados "registrados" y pueden ser usados por cualquier aplicación. Existe una lista pública en la web del IANA donde se puede ver qué protocolo que usa cada uno de ellos.
   - Puertos dinámicos o privados: Los comprendidos entre los números 49152 (C000 en hexadecimal) y 65535 (FFFF en hexadecimal) son denominados dinámicos o privados, normalmente se asignan en forma dinámica a las aplicaciones de clientes al iniciarse la conexión. Se usan en conexiones peer to peer (P2P).

2) Un “Endpoint” es el conjunto entre la dirección IP y el Puerto. A través de un endpoint, un cliente puede acceder a un servidor o hacer la petición de conexión.

3) Socket designa un concepto abstracto por el cual dos procesos (posiblemente situados en computadoras distintas) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.
   El término socket es también usado como el nombre de una interfaz de programación de aplicaciones (API) para la familia de protocolos de Internet TCP/IP, provista usualmente por el sistema operativo. Los sockets de Internet constituyen el mecanismo para la entrega de paquetes de datos provenientes de la tarjeta de red a los procesos o hilos apropiados. Un socket queda definido por un par de direcciones IP local y remota, un protocolo de transporte y un par de números de puerto local y remoto.

4) Los sockets pertenecen a la capa de transporte.
   Son los que se encargan de establecer las reglas del transporte de la información. Estas reglas pueden ser tanto bajo el protocolo TCP como UDP.

5) Los sockets pueden ser uno de dos tipos: Servidor (y como tal escuchan) o Cliente (y como tal, se conectan a un socket que escucha) . Nos referimos entonces a los distintos tipos de socket como Server o Client.
   Los servidores, utilizan sockets tipo listener, para generar y administrar las conexiones de los clientes que a ellos se conectan, a través de una IP y un puerto. Un socket tipo listener, es aquel que está "escuchando" mediante un puerto TCP, peticiones de clientes que desean conectarse al servidor. Es decir, un socket listener ES un server socket.

6) Las causas más comunes pueden ser:

   a. Dirección IP o puerto incorrecto al conectar.
   b. Que el servidor tenga el puerto bloqueado
   c. Que el servidor se apague y se cambie la IP del mismo.

7) Existen dos tipos de sockets, los que utilizan el protocolo de datagramas de usuario o UDP (User Datagram Protocol) y los que utilizan el protocolo de control de la transmisión o TCP (Transmission Control Protocol). La principal diferencia entre ambos es que el UDP necesita que le entregemos paquetes de datos que el usuario debe construir, mientras el TCP admite bloques de datos (cuyo tamaño puede ir desde 1 bytes hasta muchos K bytes, dependiendo de la implementación) que serán empaquetados de forma transparente antes de ser transmitidos.
   Existe además otra diferencia importante. Tanto los paquetes de datos UDP como los segmentos TCP (este es el nombre que reciben los paquetes TCP) pueden perderse (muy rara vez llegan al destino correcto con errores). Si un paquete se pierde el UDP no hace nada. Por el contrario, si un segmento se pierde el TCP lo retransmitirá, y este proceso durará hasta que el segmento ha sido correctamente entregado al host receptor, o se produzca un número máximo de retransmisiones.
   Finalmente, en aplicaciones en tiempo real es necesario también tener en cuenta una cosa. En el UDP controlamos qué datos viajan en cada paquete. En el TCP esto no es posible porque el empaquetamiento es automático. De hecho, el TCP espera un tiempo prudencial a tener bastantes datos que transmitir antes de enviar un segmento ya que esto ahorra ancho de banda. Si es importante que los datos tarden el mínimo tiempo posible en llegar al receptor el UDP es la mejor opción. En este sentido se dice que el UDP tiene una menor latencia que el TCP.

8)

a. Desde el lado del cliente:

    i. Sincrónico:
        1. Un socket de cliente sincrónico suspende el programa de aplicación mientras se completa la operación de red
    ii. Asincrónico:
        1. Se procesa la conexión en un hilo, mientras la aplicación continúa ejecutándose en el hilo original.

b. Desde el lado del Servidor
    i. Sincrónico:
        1. Los socket del servidor suspenden la ejecución hasta que se recibe una solicitud de conexión.
    ii. Asincrónico:
        1. Utilizan subprocesos de sistema para procesar las conexiones entrantes. Un hilo se encarga de aceptar las conexiones, otro maneja la conexión y otro recibe los datos.