

Los protocolos TCP y UDP utilizan el concepto de sockets para proporcionar los puntos extremos de la comunicación entre aplicaciones o procesos.

La comunicación entre procesos consiste en la transmisión de un mensaje entre un conector de un proceso y un conector de otro proceso, a este conector es a lo que llamamos socket.

Para los procesos receptores de mensajes, su conector debe tener asociado dos campos:

- La dirección IP del host en el que la aplicación está corriendo.
- El puerto local a través del cual la aplicación se comunica y que identifica el proceso.

Así, todos los mensajes enviados a esa dirección IP y a ese puerto concreto llegarán al proceso receptor.

Un puerto es un punto de destino que identifica hacia qué aplicación o proceso deben dirigirse los datos.

Normalmente es una aplicación cliente-servidor, el programa servidor se ejecuta en una máquina específica y tiene un socket que está unido a un número de puerto específico.

El servidor queda a la espera "escuchando" las solicitudes de conexión de los clientes sobre ese puerto.

El programa cliente conoce el nombre de la máquina en la que se ejecuta el servidor y el número de puerto por el que escucha las peticiones.

Para realizar una solicitud de conexión, el cliente realiza la petición a la máquina a través del puerto, y el cliente debe identificarse ante el servidor por lo que durante la conexión se utilizará un puerto local asignado por el sistema.

Si todo va bien, el servidor acepta la conexión. Una vez aceptada, el servidor obtiene un nuevo socket sobre un puerto diferente. Esto se debe a que por un lado debe seguir atendiendo las peticiones de conexión mediante el socket original y por otro debe atender las necesidades del cliente que se conectó.

En el lado del cliente, si se acepta la conexión, se crea un socket y el cliente puede utilizarlo para comunicarse con el servidor. Este socket utiliza un número de puerto diferente al usado para conectarse al servidor. El cliente y el servidor pueden ahora comunicarse escribiendo y leyendo por sus respectivos sockets.