Table of Contents

1 TCP/IP

El protocolo TCP/IP permitirá la comunicación usando la dirección IP de los equipos y, según el tipo de comunicación también puede que se tengan que definir los puertos, que es un número en el que los programas se pueden colocar a escuchar.

La pila TCP/IP se suele dividir en capas:

- Nivel aplicación: Es donde las aplicaciones se conectan para usar la red.
- Nivel Transporte (TCP): Se encarga de que los datos lleguen orden y no se pierdan por el camino.
- Nivel de Red (IP): Se encarga del enrutamiento (de que los datos lleguen a su destino)
- Nivel de enlace (esta capa no está bien definida en TCP/IP por lo que puede encontrar modelos de 5 capas al explicar TCP/IP): Se encarga de lidiar con el medio físico.

Los mensajes se van a dividir en paquetes.

El protocolo IP se encarga de hacer llegar los paquetes de un equipo a otro, aunque no va a garantizar su entrega (los paquetes se pueden perder). Entre otras cosas define la dirección IP de los equipos. Las funciones del protocolo IP son el direccionamiento (dar una dirección a cada equipo para identificarlo en la red) y el enrutamiento (hacer llegar la información de un punto a otro de la red).

La conexión puede ser orientada a conexión, se tendrá un par de streams en los que se colocará la información y llegará o se recibirá desde otro equipo. Es similar al concepto de tubería que ya se ha visto pero ahora la conexión será entre equipos.

Estas conexiones se hará usando sockets, que en Java pueden ser orientados o no orientados a conexión.

Lo que se hará es dividir el mensaje en pequeños paquetes que se enviarán desde el emisor al receptor. En el caso de que la conexión sea TCP, estos mensajes irán numerados de forma que si el receptor los recibe desordenados los podrá ordenar y si alguno falta lo podrá solicitar el emisor de nuevo o generar un error. Se dice que el protocolo TCP es un protocolo orientado a conexión. Es similar a una llamada telefónica en la que se llama a una persona, se establece una línea de comunicación entre ambos, la persona que recibe la llamada coge el teléfono y comienza la conversación. Cuando ambos terminan de hablar cuelgan y la llamada finaliza.

***En la conexión UDP,* no orientada a conexión o datagrama** los mensajes se envían del emisor al receptor en paquetes que no se numeran. El receptor no sabe el orden en el que se enviaron los paquetes y si algún paquete se pierde, no tiene constancia de dicha pérdida. Se dice que el protocolo

UDP no es orientado a conexión. Es similar a enviar una carta, en la que el emisor se despreocupa del mensaje una vez que pone la carta en el buzón. A partir de ese momento la carta puede que llegue, o no, el emisor no se va a enterar de dicha entrega.

Tanto el protocolo TCP como el UDP definen puertos. Los puertos son unos números que se usan para indicar el programa que debe recibir la comunicación. Si en un ordenador hay varios programas funcionando y quieren recibir mensajes de la red, cada programa se pondrá a escuchar en un puerto distinto (un número distinto). Por ejemplo, el servidor SSH suele escuchar en el puerto 22.

Hay 2^{16} = 65536 puertos en los que las aplicaciones se pueden escuchar.

- **Puertos bien conocidos:** Los puertos del 0 al 1023 son puertos del sistema y sólo el usuario root puede acceder a ellos. Son los puertos de la mayoría de aplicaciones que se encuentran en la red. Por ejemplo, servidores web en el puerto 80, FTP en el 21, SSH en el 22, DOOM en el 666,...
- **Puertos registrados:** Los puertos del 1024 al 49151 son los llamados puertos registrados y son usados por los servicios de los usuarios.
- **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).

La arquitectura habitual de comunicaciones es la **cliente-servidor.** El servidor es un equipo en el que se encuentran funcionando los servicios. Los clientes se conectan al servidor solicitando los servicios y el servidor se los proveé. Por ejemplo, piense en un servidor de correo.

1.1 Peer to peer

En la comunicación **P2P, peer to peer (de colega a colega)**, no hay servidores ni clientes y los equipos se conectan entre sí. Por ejemplo, el protocolo BitTorrent se usa para la descarga de archivos, un usuario comparte un archivo y el resto de usuarios se descargan una parte del archivo, cuando un nuevo usuario quiere descargarse el archivo, se descargará una parte de cada uno de los equipos que estén descargando el archivo en ese momento. Los equipos son a su vez servidores (de las partes del archivo que se hayan descargado) y clientes (de las partes del archivo que quedan por descargar).

Otro ejemplo son los juegos en red que suelen usar protocolos P2P para comunicar a unos jugadores con otros.

Para ver un ejemplo de P2P, se recomienda visitar:

https://joinpeertube.org/

Es un software que permite crear páginas con vídeos, tipo Youtube, pero en las que los clientes comparten con otros clientes los contenidos de los vídeos que se han descargado usando el protocolo WebTorrent.

Uno de los problemas a los que se enfrentan los protocolos P2P son los tipos de NAT que pueden tener los usuarios configurados en los routers de sus casas. El protocolo NAT permite que varios equipos salgan a través de una sola IP, la situación que se produce en los equipos de una casa que salen todos a través del mismo router. La solución es tan sencilla como que el router asocie una puerto a cada equipo de forma que todos los mensajes que van al puerto x, se reenvía al ordenador de la casa A. Los mensajes que van al puerto y, se reenvían al ordenador de la casa B.

Curiosidad:
Los tipos de NAT disponibles son:

- Full cone NAT: Todos los paquetes de la misma dirección y mismo puerto internos son mapeadas a la misma dirección y mismo puerto externo. Cualquier host externo puede mandar un paquete al host interno mandándolo a la dirección y el puerto externo que ha sido mapeado. Se conoce como también como "one-to-one NAT". (NAT uno a uno).
- Restricted cone NAT: Todos los paquetes de la misma dirección y mismo puerto internos son mapeadas a la misma dirección y mismo puerto externo. En este caso, en contraposición con full cone NAT, un host externo (con IP x.x.x.x) sólo puede mandar un paquete al host interno si previamente el host interno le había enviado un paquete a la dirección IP x.x.x.x.
- Port restricted cone NAT: es como restricted cone NAT, pero la restricción incluye también numeros de puerto. Un host externo (con IP x.x.x.x y puerto P) sólo puede mandar un paquete al host interno si previamente el host interno le había enviado un paquete a la dirección IP x.x.x.x y puerto P.
- Symmetric NAT: es NAT donde todas las peticiones de la misma IP y puerto interno con destino a otra IP y su correspondiente puerto son mapeadas en el router con la misma IP y puerto. Si el mismo host interno manda un paquete con la misma dirección interna y puerto a un destino diferente se usará un mapeo diferente. Sólo el host externo que recibe un paquete puede mandar un paquete UDP de vuelta al host interno. La diferencia del Port restricted cone NAT radica en que el mapeo de puertos será diferente según la IP con la que el equipo externo se comunique. Por ejemplo, cuando un cliente accede a Internet utilizando IP 192.168.0.1 y el puerto de origen es 56723 el NAT cambia la IP de origen para decir 56.35.67.35 pero mantiene el número de puerto igual; esto se conoce como la preservación del puerto, en cambio el NAT simétrico por cada requerimiento saliente se asigna un puerto aleatorio y éste varía para cada comunicación

La Symmetric NAT hace imposible la comunicación P2P, por lo que muchas veces será necesario usar un servidor externo que permita la comunicación entre clientes.

Para saber el tipo de NAT del que dispone un cliente, se puede usar un servidor STUN:

STUN (sigla en inglés de Session Traversal Utilities for NAT) es un protocolo de red del tipo cliente/servidor que permite a clientes NAT encontrar su dirección IP pública, el tipo de NAT en el que se encuentra y el puerto de Internet asociado con el puerto local a través de NAT.