

Informe Tarea 3.1

Emilio Vélez

Cliente: El programa en cada PC ofrecerá al usuario una barra de búsqueda en la cual insertará un texto para que sea buscado en una base de datos del servidor. El servidor compara este parámetro de búsqueda y le devuelve al cliente una lista de los archivos relacionados, cada uno con su respectiva dirección del peer que lo posee, para comunicarse directamente con él y comenzar la descarga.

Servidor: tendrá una función de buscador simplemente, este le otorga a cada solicitante de archivos que similares podrían tener algo de interés con ese string parámetro. Una analogía es cuando alguien pide LiquidPaper en prueba y el profesor/Ayudante pregunta a todo el curso quien tiene. Si algún compañero tiene se lo pasará.

Otros clientes: aparte de poder buscar archivos, son utilizados como fuente de archivos para compartir con sus similares, así el servidor no colapsa con tanto archivo en descarga ni requiere memoria para guardar archivos en su mayoría basura que la gente si la guarda en su propio PC la borrara.

(ClienteB=Cliente que realiza la búsqueda):

#se abre el programa y automáticamente se conecta con servidor, así si nosotros tenemos programas para compartir el servidor puede avisarles a otros. (la idea también es compartir y ser root a veces)

-ClienteB a Servidor: Solicita comunicación con servidor.

-Servidor a ClienteB: Comunicación iniciada (confirma) / SI NO HAY RESPUESTA REINTENTAR

-ClienteB a Servidor: Buscar archivos con el texto [INPUT]

#implementar tracker

-Servidor a todos los Clientes: Pregunta a todos los clientes DISPONIBLES si tiene archivo relacionado a [INPUT]

-Todos los Clientes a servidor: SI, tengo el [archivo.x]/NO, no tengo nada relacionado.

-Servidor a ClienteB: Envía lista de archivos relacionados con respectiva dirección del cliente/SI NO HAY RESPUESTA REINTENTAR O VOLVER A CONECTAR EN CASO DE QUE LA CONEXIÓN HAYA TERMINADO POR X MOTIVO

#ClienteB elige el archivo que quiere descargar, llamaremos al poseedor de archivo.x que quiere descargar como Cliente Proveedor, ClienteP

-ClienteB a ClienteP: solicita conexión con ClienteP

-ClienteP a ClienteB: confirma conexión / Si no hay respuesta reintentar, si luego de reiteradas veces no hay respuesta buscar otro clienteP o esperar que vuelva a estar disponible. (en programas como ARES, LIMEWIRE, EMULE solía pasar que descargas quedaban a la mitad o nunca comenzaban a descargarse)

-ClienteB a ClienteP: Iniciar descarga de [archivo.x]

-ClienteP a ClienteB: Envía paquetes de datos

#si la descarga se detiene, o tiene cualquier problema se deberá volver a conectar y continuar la descarga de paquetes desde donde se quedó

-ClienteP a ClienteB: finalizada la descarga de [archivo.x]

-ClienteB a ClienteP: Confirmado, cerrar conexión

-ClienteP a ClienteB: Confirmado, conexión cerrada

#Al cerrar programa nos desconectamos del servidor

-ClienteB a Servidor: desconectarse

-Servidor a ClienteB: desconectado

Obviamente todo lo anterior tendrá un manejo de errores y se basará en un protocolo confiable de TCP, el cual nos asegure integridad de paquete de datos y confirmación de mensajes, por que nadie quiere descargar música con archivos corruptos ni ningún otro archivo que no sea streaming directo.

General:



Solicitud y Respuesta búsqueda:



Búsqueda de archivos:



Descarga:

