



UNIVERSIDAD DE LOS ANDES  
Facultad de Ingeniería y Ciencias Aplicadas  
Sistemas Operativos y Redes 2018-20

## Tarea 3: Protocolo de red en capa de aplicación

Integrantes:

Hugo de la Fuente, María Pía Sánchez

## 1. Introducción

Para esta parte de la tarea, se pide describir una posible implementación de una aplicación de red que permita buscar y compartir archivos en internet, incorporando arquitecturas cliente-servidor y *peer-to-peer*. Para ello, se detallan a continuación los siguientes puntos:

1. Tipos de *Hosts*.
2. Interacciones posibles en el protocolo.
3. Tipos y formato de mensaje intercambiado en cada interacción.

La segunda parte de esta tarea se realizará utilizando *python3* utilizando la librería *pyp2p* para manejar las conexiones de cada cliente con el *tracker*, cabe mencionar que en esta red habrán *hosts* permanentes que servirán de *trackers*, la funcionalidad de estos se detalla más adelante.

## 2. Tipos de *Hosts*

Habrían varios *hosts* terminales, los cuales serían los dispositivos que quieren descargar un archivo y los que lo contienen para descargarlo. También, estaría el *host* servidor o *tracker*, cabe mencionar que estos pueden almacenar archivos pero su principal función es manejar un diccionario con el nombre del archivo que se quiere descargar que actúa como llave, y como valor el o los *hosts* terminales que poseen el archivo y además están conectados a Internet. A medida que un *host* terminal sube un archivo o lo descarga en su carpeta compartida, se actualizan los parámetros pertinentes en el diccionario.

### 3. Interacciones

1. Consulta al Servidor El cliente manda una solicitud al servidor para descargar el archivo deseado.
2. Respuesta del Servidor El servidor le responde al cliente que solicita un archivo, con la dirección IP pública de algún *host* que lo contenga y que, además, esté conectado a internet. Si no hay dispositivos conectados que contengan el archivo, el servidor enviará un mensaje de error.
3. Compartir IP el cliente que quiera compartir archivos debe conectarse con el servidor para que éste *sepa* que dicho *host* lo tiene.
4. Verificar Conectados El servidor se tiene que comunicar con los clientes que contienen el archivo para asegurarse de que estén conectados, y así enviarle la IP pública del *host* que esté en línea al cliente que solicitó dicho archivo.
5. Descarga del Archivo: una vez que el cliente recibe la dirección IP, éste se comunica con el *host* para realizar la descarga del archivo. Esta interacción se realizará a través de una transmisión UDP.

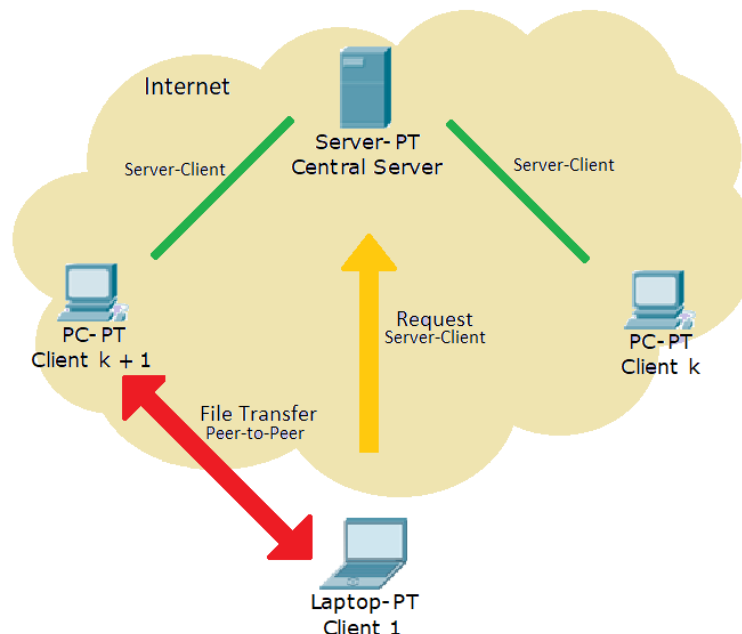


Figura 1: Interacciones.

## 4. Tipos y formatos de mensaje

Cada mensaje tiene una estructura definida por un comando luego se especifica el nombre del archivo seguido de un *EOM* para indicar el final del mensaje. El formato de mensaje para la descarga de archivos será el siguiente:

GET <Nombre del Archivo>EOM

En respuesta el servidor buscará dentro del diccionario el nombre del archivo y si este se encuentra en más de un *host* se le entregará el archivo usando cualquiera de esos *hosts*. Cabe destacar que en cada descarga se actualiza el diccionario con el fin de llevar un registro de la ubicación de cada archivo. En caso de no existir el archivo, el servidor notificará al usuario de este hecho.

Como funcionalidad extra, se implementará un mensaje para verificar si un archivo existe previo a descargarlo, esto con el fin de no realizar descargas innecesarias y hacer un poco más eficiente la actualización del diccionario.

SEARCH <Nombre del Archivo>EOM

En respuesta el servidor le indicará si el archivo existe o no en la red y le dará la opción de descargarlo.



## 5. Bibliografía

1. Jeff Tyson. (30 de Octubre del 2000). Peer-to-Peer File Sharing - How Napster worked. howstuffworks.com/ Sitio web: <https://computer.howstuffworks.com/napster2.htm>
2. Short Bytes. (6 de Enero ed 2016). What Is P2P File Sharing And How It Works?. Short Bytes Sitio web: <https://fossbytes.com/what-is-p2p-file-sharing-and-how-it-works>