

# Protocolo P2P para compartir archivos

---

Pedro Ricci

## Introducción

---

Este protocolo tiene como objetivo permitir el intercambio de archivos a través de internet (IPv4) por medio de un modelo descentralizado, es decir, no existe una estructura Servidor-Cliente estricta.

## Hosts

---

Todos los hosts que utilizan el protocolo son llamados *peers*. Estos no poseen un rol específico de cliente o servidor, sino que actúa como uno u otro dependiendo la circunstancia. Por ejemplo, si un peer *A* solicita un archivo que se encuentra bajo posesión de un peer *B*, *A* actúa como cliente y *B* como servidor. Estos roles se intercambiarían en caso de que *B* llegase a solicitar un archivo que reside en el nodo *A*.

Para que el protocolo sea lo más descentralizado posible, no existe un host específico que actúa como *tracker*, conectando los peers entre sí, sino que todos los peers actúan como trackers. Cada peer guarda un listado de los peers que ha descubierto en la red, el cual puede compartir a un nuevo peer, recién llegado, que no conoce a nadie más que a él. El protocolo bittorrent utiliza algo parecido llamado DHT<sup>1</sup>.

## Interacciones entre peers

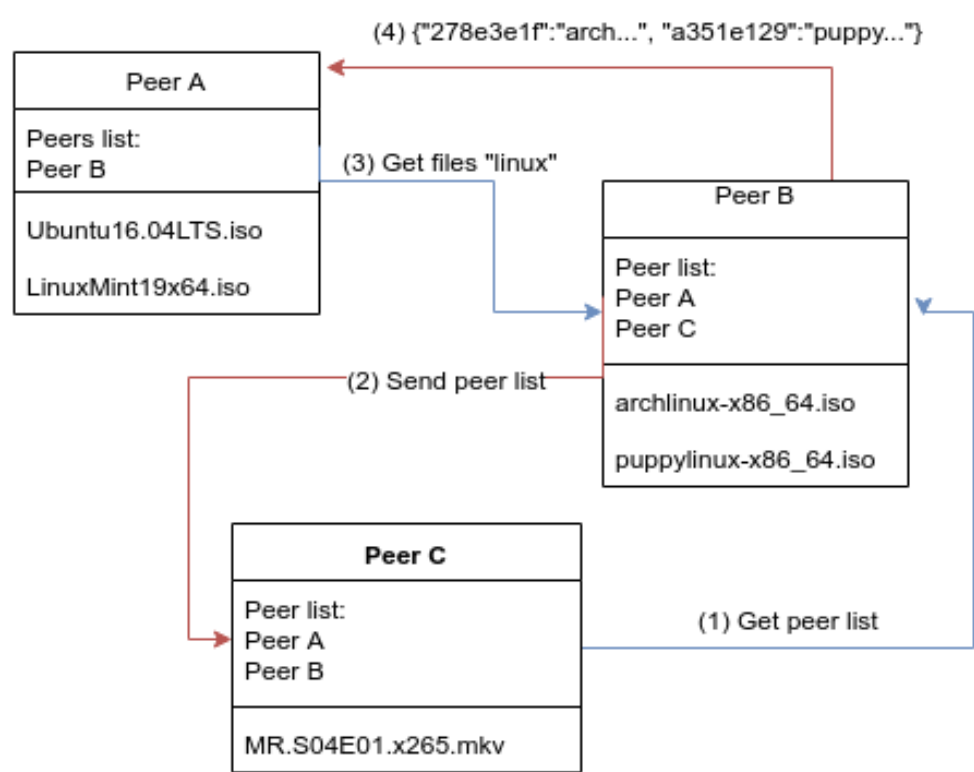
---

Cada peer posee un identificador único.

Los peers pueden:

- Hacer ping para verificar el estado de determinado peer
- Solicitar el listado de peers de otro peer, para agregar a su lista

- Preguntar a otro peer si posee archivos cuyos nombres concuerden con un string
- Responder a la consulta de archivos (afirmativa o negativa), junto a un arreglo de los archivos que hacen match con el string solicitado y sus respectivos hash
- Solicitar el envío de un archivo en base a su hash
- 



## Mensajes

Se utilizará TCP.

4 Bytes para el identificador de mensaje

Los mensajes posibles son:

- RQLT: (**Re**quest **Li**st) Solicitar lista de peers a otro peer
- PING: Obtener respuesta para verificar que un peer aún está vivo
- RQFS <string> : (**Re**quest **f**iles matching **s**tring) Solicitar una lista de los archivos que posee el peer destino que hacen match con el string
- RPFL <# of files> <dict(filename:hash)> : (**Re**sponse **F**ile **l**ist)  
Respuesta a RQFS. Numero de matches y dichos matches
- RQFH <hash of wanted file> : (**Re**quest **f**ile of **h**ash) Solicitar a peer el archivo con hash determinado

- TPHF <peer id> <string> : (**This peer has files matching string**): Un peer puede consultar a los peers en su lista si alguno tiene un archivo con la string e informar al peer que le consultó por dicho string (hay que ver a que profundidad llegar)
- ERRR <error id> : (**Error**) Informar que ocurrió un error (id's por definir)
- ACKN <message> : Informar la recepción de los diversos mensajes.

## Otros

---

Se puede “encadenar” búsquedas. Si se envía RQFS a un peer, este puede solicitar el listado de peers del emisor y reenviar el mensaje RQFS a los peers que posee él pero no el emisor. Esto con una profundidad y mecanismo por determinar (para evitar loops infinitos)

## Conclusión

---

Solo existen peers. Estos son clientes y servidor.

No hay peer central, pero si queremos conectarnos a una red de peers debemos conocer por lo menos uno y conectarnos directamente. Luego copiamos su lista de peers conectados.

- 
1. Descripción oficial del protocolo DHT, implementado sobre UDP  
[http://www.bittorrent.org/beps/bep\\_0005.html](http://www.bittorrent.org/beps/bep_0005.html) ↩