



UNIVERSIDAD TÉCNICA
FEDERICO SANTA MARÍA

¿Cómo funciona el software μ Torrent



(Cliente BitTorrent)?

Integrantes: Manuel Cruces

Tomás Vega

Luciano Ortiz de Zárate

Fecha: 24-08-18

Resumen

Se estudiará el funcionamiento de este software, pero principalmente, el medio por el cual este mismo logra realizar las descargas, hablando también de su relación con las redes p2p. Se explicarán conceptos como seeders y leechers. También se verá la actualidad del programa y su relevancia actual en el campo de transportes de archivos grandes en el internet.

Introducción

Con la masificación de internet se ha hecho más accesible el acceso de información y contenidos para las personas, dicha información o contenido puede llegar en cuestión de minutos al acceso de las personas, este cambio ha producido grandes desafíos para la compartición de archivos multimedia, que un archivo sea accesible es necesario que exista alguien para suministrarlo. Pues el paradigma cliente-servidor existen fallas como la congestión sobre el servidor y un deterioro en la velocidad de descargar a medida que la gente quiera más acceso a este y más si este archivo es de gran tamaño. Tras esto una de las motivaciones al realizar este trabajo fue, ¿Cuál es una solución para la descarga de archivos pesados y optimizar el tiempo de descarga? Tras investigar nos encontramos con uTorrent que es un cliente Bittorrent, que es la solución para lo dicho anteriormente pero ¿Qué es bittorrent?

El bitTorrent es un protocolo de red de pares, el cual fue creado por Bram Cohen en el año 2001 y programado en python, se volvió la solución para el problema de descargas con un contenido de gran tamaño. Ya que en vez de descargar el archivo deseado desde un servidor, el protocolo de torrent permite conectarse entre usuarios como un tipo de enjambre para descargar y subir un archivo. Si un usuario quiere compartir un archivo, primero crea un archivo torrent y los distribuye, la típica forma de distribución de este es a través de páginas web, una de las famosas es The Pirate Bay. Al hacer esto genera que el archivo esté disponible en la red y a través de un nodo torrent que trabaja como semilla. Pero cuando quieres descargar un archivo, uno obtiene el archivo desde cualquier usuario que tenga el archivo (semilla), cuando quieres descargar un archivo actúas como leecher (sanguijuela).

Una solución para este problema es la red de pares, más conocidas como P2P, pero que son la red de pares, la red de pares es una red de ordenadores que como nodos que se comportan igual entre sí. Estos nodos actúan como si fueran cliente-servidor. Por lo tanto la red de pares es un sistema descentralizado de descarga, al contrario con el sistema convencional, este mientras sea más popular la descarga, son menores los tiempos de esta, debido a que existen más fuentes disponibles donde realizar la descarga. Un sistema con red de pares es Torrent, un sistema de redes independientes el cual vamos a profundizar su esquema

Ventajas y desventajas de un paradigma P2P en Utorrent

El paradigma arquitectónico que usa Utorrent es “red de pares” o mejor conocido como peer-to-peer, donde los ordenadores se comportan igual entre sí, es decir, como cliente y servidor.

Se escoge este en lugar del sistema convencional, ya que, es descentralizado, además de ser eficiente mientras el archivo sea más demandado, logrando con ello un menor tiempo de descarga debido a la existencia de los peers, y además debido a la poca o casi nula presencia de infraestructura abarata los costos.

A pesar de lo anterior, algunos problemas generados por esta elección son la seguridad, debido a la naturaleza abierta y distribuidora. Otro inconveniente se genera cuando el archivo no es tan popular, retrasando los tiempos de descarga debido a la falta de servidores.

Protocolo BitTorrent

El protocolo BitTorrent, fue creado para el intercambio de archivos basados en una red peer-to-peer, dentro de internet. El fin de la creación de este protocolo, fue la eliminación de la centralización de archivos demasiado grandes en los servidores. De esta forma, el protocolo BitTorrent permite a sus usuarios conectarse entre ellos, para así descargar y subir el archivo, simultáneamente.

Peers, Seeders, Leechers, Trackers y Enjambre.

Peers (punto): Se le conoce como peer a cualquier usuario (host) que esté en la red.

Seeders (semillas): Los seeders o semillas, son usuarios (peer) los cuales han descargado el 100% del archivo, y se encuentran compartiéndolo, para que así, otros usuarios de la red puedan conectarse (a la semilla) y descargar el archivo. Es decir, si descargamos un archivo entero, y lo compartimos, nos convertiremos en semillas. Para poder descargar el 100% del archivo, es necesario por lo menos tener una semilla.

Leechers (sanguijuelas): Se les llama leechers a todos los usuarios conectados a la red, que se encuentran descargando el archivo, es decir, no tienen el 100% del archivo descargado. Un leecher, se encarga de descargar fragmentos que aún no están en el ordenador, mientras comparte a los demás leecher partes del archivo que ya están en el ordenador. También son conocidos como leechers, usuarios los cuales descargaron el 100% del archivo, pero no lo comparten a la red.

Trackers(rastreadores): Los trackers, son servidores especiales que contienen información necesaria para que los usuarios de la red(peers) se conecten con

otros usuarios, asistiendo la comunicación entre ambos peers, usando el protocolo BitTorrent. En pocas palabras, es el ducto que hay entre dos usuarios en la red, el cual se encarga de la comunicación y distribución de datos entre ellos. En caso de no tener un rastreador, el usuario cliente(que quiere realizar la descarga), no podrá conectarse con el enjambre, y con ello, no podrá realizar la descarga.

Enjambre: Cuando se habla del enjambre, se refiere a todos los usuarios los cuales el tracker se encarga de buscar. Se le llama enjambre, ya que el tracker actúa como un panal de abejas, el enjambre de abejas son los usuarios y la miel es el torrent.

Archivo Torrent

Definido en el protocolo BitTorrent, y con extensión *.torrent*, un archivo de este tipo se encarga de almacenar URL(s) de tracker(s), y la integridad de los metadatos de cada fragmento, los cuales serán usados por un cliente BitTorrent. Un archivo Torrent, contiene información acerca de la localización de diferentes piezas un archivo de destino, pero no información sobre el contenido del archivo. El trabajo de los Torrents, es dividir el archivo de destino en pequeños fragmentos de información, los cuales puede estar localizados en ilimitados números de host. Una vez descargados los fragmentos (y sólo cuando haya terminado), el cliente puede ensamblar los fragmentos, dejando un archivo utilizable.

Estructura del archivo Torrent

Un archivo torrent, es un diccionario *Bencode* con las siguientes claves:

- **announce* : URL(s) de(los) tracker(s)
- **info* : crea un diccionario con claves independientes de si uno o más archivos son compartidos:
 - **name* : directorio donde el(los) archivo(s) serán guardados.
 - **piece length*: número de bytes por pieza (comúnmente 256 KiB)
 - **pieces* : lista de hash, la cual tendrá una longitud múltiplo de 160 bits
 - **length* : tamaño del archivo destino, en bytes.
 - **files* : lista de diccionarios, que corresponden cada uno a un

archivo(cuando varios archivos son compartidos). Cada uno de estos diccionarios tiene las siguientes claves:

**path* : lista de cadenas correspondientes a los nombres de los subdirectorios, el último de los cuales será el verdadero nombre.

**length*: tamaño del archivo en bytes.

Extensiones del archivo Torrent

- Tablas de hash distribuido: Un torrent sin trackers, no tiene la clave *announce*, por lo tanto se le asigna la clave *nodes* de esta forma:

```
{  
...  
'nodes':[[<host>,<port>], [<host>,<port>], ...]  
...  
}
```

- Múltiples trackers: En caso de que hayan múltiples trackers, se añade una nueva clave, *announce-list*, en la parte superior del Torrent (donde se ubica *info* o *announce*):

```
{  
...  
'announce-list'=[[tracker1],[tracker2], ...]  
...  
}
```

- HTTP seed: Si se desea añadir direcciones webs, donde el torrent pueda ser recuperado, se añade la clave *httpseeds* (parte superior de la lista), en donde su valor es una lista con las direcciones webs:

```
{  
...  
'httpseeds'=['http://www.site1.com/source1.php',  
             'http://www.site2.com/source2.php',...]  
...  
}
```

}

- Torrents privados: En caso de ser un Torrent privado, se añade la clave *private* en el diccionario de *info*. El valor de *private* es 1 si el torrent es privado o 0 en caso que sea público.

Actualidad

En la actualidad utorrent se encuentra escrito en c++, además es conocido por su alta tasa de uso para piratear, lo cual no es el uso que sus creadores buscaban. En el año 2012 la página oficial de Utorrent se declararon contrarios a la piratería y que sus aplicaciones son más amplias y legales. Un ejemplo de lo anterior es su uso para la descarga de actualizaciones en juegos masivos en línea..

Referencias:

- https://es.wikipedia.org/wiki/Archivo_Torrent#cite_note-4

-<http://petercasillas1.blogspot.com/2014/01/archivo-torrent-y-estructura-de-una-red.html>

<https://www.genbeta.com/herramientas/los-responsables-de-utorrent-se-declaran-completamente-contrarios-a-la-pirateria>

<https://www.youtube.com/watch?v=EkkFT1bRCT0&t=221s>

<https://www.howtogeek.com/141257/htg-explains-how-does-bittorrent-work/>

Quinta edición de “redes de computadoras un enfoque descendente”

<https://sites.google.com/site/pratikgetsitall/tutorials-1/utorrent-1>

<https://es.ccm.net/faq/3149-que-son-los-seeders-y-leechers>