

PRÁCTICA 3: P2P

1 Objetivos

En esta práctica se pretende conocer más en detalle el funcionamiento de una red P2P como la de BitTorrent para extraer la máxima información posible de esa red mediante implementación de los programas necesarios. Puede elegir cualquier lenguaje de programación: Java, PHP, python, etc.

2 Material

- PC Linux/Windows con conexión a Internet
- Entorno de programación en el lenguaje deseado

3 Protocolo BitTorrent

La especificación del protocolo de BitTorrent la puede encontrar en <https://wiki.theory.org/BitTorrentSpecification> y con otro enfoque en <http://jonas.nitro.dk/bittorrent/bittorrent-rfc.html> . Eche un vistazo a la especificación del protocolo.

3.1 Cree una carpeta bittorrent en su \$HOME donde guardará todo el material generado en la práctica.

4 Selección del lenguaje de programación

En esta práctica puede elegir el lenguaje de programación que desee para desarrollar la aplicación. Sin embargo, intente elegir el lenguaje con cierto criterio de utilidad:

- Según su familiarización con el mismo
- Según la “simplicidad” del lenguaje
- Según la existencia de librerías relacionadas con BitTorrent disponibles o funcionalidades concretas necesarias que haya podido ver en la especificación del protocolo como la decodificación Beconding.

5 Obtención y decodificación del fichero .torrent

El fichero metafile .torrent es el que caracteriza el funcionamiento de un determinado torrente de descarga de cierto contenido.

5.1 Usar algún buscador de torrents (por ejemplo, <http://thepiratebay.se/> , <http://extratorrent.com> , <http://torrentz.eu> , etc.) para encontrar un fichero .torrent que caracterice el torrente. Intentar que sea un torrente activo (por ejemplo de un contenido actual) para que tenga el mayor número de nodos conectados a cualquier hora. También puede acudir a listado de distribuciones Linux distribuidas por BitTorrent como <http://torrent.ubuntu.com:6969/> .

Contemplará que en algunos sitios no puede descargar el .torrent directamente

sino que se le da un enlace magnético (<http://es.wikipedia.org/wiki/Magnet>) . ¿Por qué se utilizan en lugar de los .torrent? En su caso, el obtener el .torrent del enlace magnético lo haremos con herramientas online como <http://magnet2torrent.com/> (Si al final de la práctica tiene tiempo se podrá volver a aquí a intentar automatizar esta conversión de enlace magnético a .torrent).

- 5.2 Una vez tenga descargado el fichero .torrent y lo abra como un fichero de texto observará que está codificado de forma especial. Usa la codificación BEncode (<https://wiki.theory.org/BitTorrentSpecification#Bencoding>). Revise los principios de la codificación e implemente un programa para su decodificación usando alguna librería que realice el trabajo. En esa misma URL tiene referencias a librerías o puede usar estas otras, no implemente la decodificación usted mismo:
- Java: <http://www.ohloh.net/p/jbencode>
 - PHP: https://wiki.theory.org/Decoding_encoding_bencoded_data_with_PHP
 - Python: <https://pypi.python.org/pypi/bencode>
- 5.3 Puede comparar su decodificación con la obtenida por herramientas online como http://www.tools4noobs.com/online_tools/torrent_decode/ o <http://pauw.student.utwente.nl/bencode.html> .
- 5.4 Interprete los campos de la decodificación. En particular obtenga la URL del tracker (campo announce, de aquellas alcanzables por HTTP) y el hash del fichero. Si su .torrent apunta a un tracker UDP busque otro que .torrent que tenga un tracker HTTP.

6 Comunicación con el tracker

Una vez conocida la URL del tracker, se puede conectar con él y hacer una petición HTTP GET con una serie de parámetros 'announce-url?param=value¶m=value&...'. Tenga cuidado de codificar aquellos caracteres que sean especiales (<http://en.wikipedia.org/wiki/Percent-encoding>). El detalle de estos parámetros los tiene en la especificación del protocolo ([https://wiki.theory.org/BitTorrent Tracker Protocol](https://wiki.theory.org/BitTorrent_Tracker_Protocol)). Se trata de implementar un pequeño programa que realice la primera consulta al tracker con los parámetros adecuados y en particular el parámetro event=started para indicar el comienzo de nuestro peer. En la respuesta obtendrá la lista de peers con sus direcciones IP y puertos conectados al torrente. Para ello tenga en cuenta:

- 6.1 Según el tracker utilizado cambian los parámetros obligatorios. Por ejemplo, uno que funciona es

```
http://torrent.ubuntu.com:6969/announce?info_hash=%da%acp%08%e2%e3%a6%e42%19P%c11i%0a%ca%20%c5%a0%8a&peer_id=-UT3300-%b9s%0a%60%24%d4Q%81j%9b%9dV&port=61552&uploaded=0&downloaded=0&left=831717376&key=0&event=started&numwant=50&compact=0
```

- 6.2 El info_hash del fichero .torrent es una cadena hexadecimal de un hash SHA-1 que debemos convertir a URLEncoded. Su programa tiene que hacer la conversión. Puede verificar la conversión en <http://www.asciitohex.com/> rellenando el info_hash donde dice Hexadecimal, pulsar Convert y obteniendo el

resultado que aparece en el cuadro URL Encoded.

- 6.3 Existe una versión del protocolo de comunicación con el tracker sobre UDP más compleja (http://xbtt.sourceforge.net/udp_tracker_protocol.html) que no vamos a abordar. Elegir siempre un tracker http.
- 6.4 La respuesta viene codificada BEncode, con la lista de direcciones IP y puertos del número de peers que hayamos solicitado. Cada vez que se repita la consulta se obtendrá una lista de peers aleatoria de los conectados al torrente. Fíjese que existe dos formas de codificar la lista de peers.
- 6.5 ¿Qué otra información obtiene de la respuesta del tracker?

Checkpoint P03.1: Avisar al responsable de prácticas cuando haya completado las prácticas hasta este apartado. No se quede bloqueado, mientras tanto avance con las siguientes secciones.

7 Comunicación con los peers

Una vez que conoce la lista de peers, implemente un programa que pasada una lista de peers, se conecte con ellos y obtenga toda la información posible de los mismos. Será una conexión TCP a la dirección IP y puerto de cada peer de forma correlativa entre todos los peers, que se puede repetir continuamente. Tenga en cuenta:

- 7.1 Su programa es el encargado de establecer la conexión y comenzar el handshake (<https://wiki.theory.org/BitTorrentSpecification#Handshake>). La codificación del info_hash es la misma que la utilizada con el tracker.
- 7.2 Por cada peer, extraiga la máxima información posible (¿fichero que se está descargando?, piezas que posee, velocidad de descarga, etc)
- 7.3 Fíjese que usted no va a participar en la descarga del torrente, solamente va a conectarse a los peers para obtener información de los mismos.

8 Integración (opcional)

Este apartado es opcional: únicamente puede realizarlo si no ha superado el número de sesiones de prácticas pensadas para esta práctica.

- 8.1 Integre la operativa de los programas anteriores en un solo programa que partiendo de un fichero .torrent vuelque en pantalla o fichero la información de todos los peers que participan en ese torrente, actualizada mientras el programa siga en funcionamiento.
- 8.2 Valore y en su caso implemente que la entrada a su programa pueda ser un enlace al .torrent en lugar del propio fichero y también que pueda ser un enlace magnético.

Checkpoint P03.2: Avisar al responsable de prácticas cuando haya completado las prácticas hasta este apartado. No se quede bloqueado, mientras tanto avance con las siguientes secciones.