**BT笔记**

## BT种子文件解析

announce:Tracker的主服务器  
  
announce-list(可选)：备用tracker服务器列表  
  
comment(可选):种子文件的注释  
  
comment.utf-8(可选)：种子文件注释的utf-8编码  
  
creation date：种子文件建立的时间，是从1970年1月1日00:00:00到现在的秒数。  
  
encoding:种子文件的默认编码，比如GB2312，Big5，utf-8等  
  
info：所有关于下载的文件的信息都在这个字段里，它包括多个子字段，而且根据下载的是单个文件还是多个文件，子字段的项目会不同。  
  
当种子里包含多个文件时，info字段包括如下子字段：  
  
files：表示文件的名字，大小，该字段包含如下三个子字段：  
  
 lenghth：文件的大小，用byte计算  
  
 path：文件的名字，在下载时不可更改  
  
 path.utf-8：文件名的UTF-8编码，同上  
  
以上的三个字段每个文件都有一组值。  
  
name：推荐的文件夹名，此项可于下载时更改。  
  
name.utf-8:推荐的文件夹名的utf-8编码，同上。  
  
piece length：每个文件块的大小，用Byte计算  
  
pieces：文件的特征信息，该字段比较大，实际上是种子内包含所有的文件段的SHA1的校验值的连接，即将所有文件按照piece length的字节大小分成块，每块计算一个SHA1值，然后将这些值连接起来就形成了pieces字段，由于SHA1的校验值为20Byte，所以该字段的大小始终为20的整数倍字节。该字段是Torrent文件中体积最大的部分，可见如果大文件分块很小，会造成Torrent文件体积庞大。  
  
publisher：文件发布者的名字  
  
publisher.utf-8：文件发布者的名字的utf-8编码  
  
publisher-url：文件发布者的网址  
  
publisher-url.utf-8：文件发布者网址的utf-8编码。  
  
另外，当发布单文件时，files字段是没有的，而  
  
  
lenghth:  
name：  
name.utf-8:  
  
这三个字段负责描述单文件的属性：大小，名字，名字的utf-8编码。其他项目和多文件相同。  
  
以上的项目即为info字段的全部。  
  
说到info就不得不说INFO\_HASH，这个值是info字段的HASH值，20个Byte，同样是使用SHA1作为HASH函数。由于info字段是发布的文件信息构成的，所以INFO\_HASH在BT协议中是用来识别不同的种子文件的。基本上每个种子文件的INFO\_HASH都是不同的(至少现在还没有人发现有SHA的冲突)，所以BT服务器以及客户端都是以这个值来识别不同的种子文件的。  
  
计算的具体范围是从info字段开始(不包含"info"这四个字节)，一直到nodes字段为止(不包含"nodes"这5个字节和nodes前边表示nodes字段长度的"5:"这两个字节)。另外，INFO\_HASH值是即时计算的，并不包含在Torrent文件中。  
  
nodes:最后的一个字段是nodes字段，这个字段包含一系列ip和相应端口的列表，是用于连接DHT初始node。  
  
综上，多文件Torrent的结构的树形图为：  
  
Multi-file Torrent  
├─announce  
├─announce-list  
├─comment  
├─comment.utf-8  
├─creation date  
├─encoding  
├─info  
│ ├─files  
│ │ ├─length  
│ │ ├─path  
│ │ └─path.utf-8  
│ ├─name  
│ ├─name.utf-8  
│ ├─piece length  
│ ├─pieces  
│ ├─publisher  
│ ├─publisher-url  
│ ├─publisher-url.utf-8  
│ └─publisher.utf-8  
└─nodes  
  
单文件Torrent的结构的树形图为：  
  
Single-File Torrent  
├─announce  
├─announce-list  
├─comment  
├─comment.utf-8  
├─creation date  
├─encoding  
├─info  
│ ├─length  
│ ├─name  
│ ├─name.utf-8  
│ ├─piece length  
│ ├─pieces  
│ ├─publisher  
│ ├─publisher-url  
│ ├─publisher-url.utf-8  
│ └─publisher.utf-8  
└─nodes  
  
下面是一个实际的例子来说明Torrent文件的实际结构：  
  
首先要了解在Torrent文件里数据的逻辑表示方法  
  
字符串： 字符串由 字符个数:字符串 的形式构成，比如 5:files 就表示"files"  
整数： Torrent用10进制来表示整数，由 i数字e 的形式构成，如 i50e 就表示50  
列表： 列表由 l 开头，e 结尾，中间是多个字符串，如 l5:files4:infoe 就表示 ["files","info"]  
字典： 由 d 开头，e 结尾，中间可以是任何结构， 如 d5:filesl4:info3:eggee 就表示 {files, ["info, "egg""]}  
  
  
示例的文件是一个由包含三个pdf文件的Torrent，在目录 MySampleTorrent 内，目录结构为：  
  
MySampleTorrent  
├─protocol.pdf  
├─Draft\_DHT\_protocol.pdf  
└─fast\_extensions.pdf  
  
通过BitComet制作出的Torrent文件名为 MySampleTorrent.torrent，大小为 976 Byte，分块大小为8M，以下是用WinHex打开该Torrent的显示：  
  
由上图可以得到的信息对应结构树：  
  
├─announce ->http://www.sample.org/announce  
├─announce-list  
├─comment ->This is a sample comment of thsi torrent file.  
├─comment.utf-8 ->This is a sample comment of thsi torrent file.  
├─creation date ->1182679174  
├─encoding ->Shift-JIS  
├─info  
│ ├─files  
│ │ ├─length ->59126  
│ │ ├─path ->Draft\_DHT\_protocol.pdf  
│ │ └─path.utf-8 ->Draft\_DHT\_protocol.pdf  
│ ├─name ->MySampleTorrent  
│ ├─name.utf-8 ->MySampleTorrent  
│ ├─piece length ->8388608  
│ ├─pieces ->751CE8FD85D69D5EDEB31F6B15F3C729587BDDD6  
│ ├─publisher ->Ci Bech  
│ ├─publisher-url ->http://www.cibech.cn  
│ ├─publisher-url.utf-8  
│ └─publisher.utf-8  
└─nodes  
  
由于只有一个Tracker，所以没有 announce-list ，其余不在图上的信息就不列举了。  
这和uTorrent读到的结果是一致的  
  
再来看看校验码的部分，由于设置为8M的文件分块，而总文件大小不到1M，所以三个文件混合成一个块来计算SHA1值。所以先将三个文件用2进制方式合并成一个，再计算SHA1值，如下：  
  
对比这里划白线的SHA1值，是否和上面文件中的一致。  
  
pieces ->751CE8FD85D69D5EDEB31F6B15F3C729587BDDD6  
  
另外看看INFO\_HASH的计算，先将INFO部分用WinHex拷贝出来，单独形成一个文件，命名为 info.part ，需要拷贝的部分如图中所选择部分  
  
红线标识出了开始和结尾，同样，计算该文件的SHA1值，得到结果如下：  
  
将这个校验值和BitComet计算的特征码比较  
  
结果是一致的。BT的服务器同样是根据这部分计算的特征值，确保了无相同的torrent被上传。而BT客户端也是根据这个INFO\_HASH来请求Peer的。

## BT是通过.torrent下载的原理

### BT协议之连接TrackerServer

TrackerServer有HTTP和UDP协议两种，这里简单介绍一下HTTP协议的TrackerServer，BTClient通过

HTTP Get请求完成Peers获取和自身的注册。请求的URL格式如下：

Trackerserver-url?info\_hash=xxxxxxxxxxxx,peer\_id=xxxxxxxxxxx,ip=x.x.x.x,port=xxxx,uploaded=xx,downloaded=xx,left=xx,event=x

URL中各参数需要经过urlencode处理，各个参数的意义如下：

|  |  |
| --- | --- |
| 参数名称 | 参数意义 |
| info\_hash | torrent文件中info属性的value部分(bencode格式)的SHA1 哈希值，这个哈希值是BT下载任务的唯一标识。 |
| peer\_id | 任务启动时BT Client为自己随机分配的20字节的ID |
| ip | BTClient的外网IP |
| port | BTClient监听的端口 |
| uploaded | BTClient对该任务已上传的字节数 |
| downloaded | BTClient对该任务已下载的字节数 |
| left | BTClient对该任务尚未下载的字节数 |
| event | 可选字段，表示BTClient对当前下载任务的状态，可选值：started,completed,stoped |

Hash 94b9354069a1347e0401de80411d6a3269cd12a5

分片长度 1048576

主要地http://tracker.trackerfix.com:80/announce

其他地址http://tracker.trackerfix.com:80/announce

其他地址 udp://9.rarbg.me:2710/announce

其他地址 udp://9.rarbg.to:2710/announce