

## La terminologie

**Peer:** un pair est un nœud dans le réseau qui participe au partage de fichier. Il peut agir simultanément en tant que serveur ou en tant que client avec les autres nœuds du réseau.

**Neighboring peers:** Pairs à qui un client a un point actif à point TCP.

**Client:** Un client est un agent utilisateur qui agit comme un pair pour le compte d'un utilisateur.

**Torrent:** Un torrent est le terme pour le fichier (un seul fichier torrent) ou un groupe de fichiers (multi-fichiers torrent) que le client télécharge.

**Swarm:** Un réseau de pairs qui opèrent activement sur un torrent donné.

**Seeder:** Un pair qui a une copie complète d'un torrent.

**Tracker:** Un tracker est un serveur centralisé qui contient des informations sur un ou plusieurs torrents et des réseaux de pairs associés. Il fonctionne comme une passerelle pour les pairs dans un réseau de pairs.

**Metainfo file:** Un fichier texte qui contient des informations sur le torrent, par exemple, l'URL du tracker. Il a généralement le .torrent d'extension.

**Peer ID:**

Une chaîne de 20 octets qui identifie le poste. Comment l'ID des pairs est obtenu en dehors de la portée de ce document, mais un poste doit faire en sorte que l'ID de pairs qu'il utilise a une très forte probabilité d'être unique dans le réseau de pairs.

**Info hash:** Un hash SHA1 qui identifie de manière unique le torrent. Elle est calculée à partir de données dans le fichier de metainfo.

## L'encodage

Dans BTP / 1.0 le fichier metainfo et toutes les réponses du Tracker sont encodés dans le format de bencoding. Le format spécifie deux types scalaires (entiers et chaînes) et deux types de composés (listes et dictionnaires).

```
dictionary = "d" 1*(string anytype) "e" ; non-empty dictionary
list       = "l" 1*anytype "e"          ; non-empty list
integer    = "i" signumber "e"
string     = number ":" <number long sequence of any CHAR>
anytype    = dictionary / list / integer / string
signumber  = "-" number / number
number     = 1*DIGIT
CHAR       = %x00-FF                      ; any 8-bit character
DIGIT      = "0" / "1" / "2" / "3" / "4" /
            "5" / "6" / "7" / "8" / "9"
```

Les entiers sont codés en préfixant une chaîne contenant la représentation en base dix du nombre entier avec la lettre "i" et postfixant avec la lettre "e". Par exemple l'entier 123 est codé comme i123e.

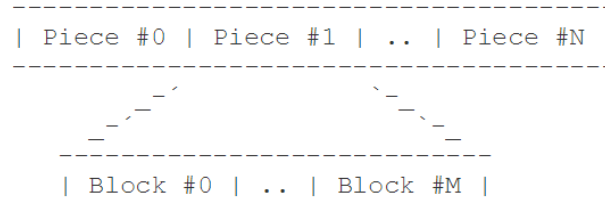
Les chaînes sont encodées en faisant précéder le contenu de la chaîne avec la longueur de la chaîne suivi de deux points. Par exemple la chaîne "announce" est codé comme "8: announce".

Les listes sont un nombre arbitraire d'éléments qui sont « bencodé » précédés de la lettre "l" et postfixé avec la lettre "e". Il en résulte que les listes peuvent contenir des listes imbriquées et des dictionnaires. Par exemple "li2e3: foo" définit une liste contenant le nombre entier "2" et la chaîne "foo".

Les dictionnaires sont un nombre arbitraire de paires clé / valeur délimités par la lettre "d" au début et à la lettre "e" à la fin. Toutes les touches sont bencodées cordes tandis que la valeur associée peut être tout élément bencodé. Par exemple " d5:monthi4e4:name5:april" définit un dictionnaire contenant les associations: "month" => "4" et "name" => "april". Toutes les clés de dictionnaire doivent être triées.

## Les pièces et les blocks

Chaque pièce représente une série de données dont il est possible de vérifier en utilisant un hachage SHA1 pièce. Lors de la distribution des données sur pièces PWP (Peer Wire Protocol) sont divisés en un ou plusieurs blocs, comme le montre le schéma ci-dessous:



Le nombre de pièces dans le torrent est indiqué dans le fichier de metainfo. La taille de chaque pièce dans le torrent reste fixe et peut être calculée selon la formule suivante:

$$\text{fixed\_piece\_size} = \text{size\_of\_torrent} / \text{number\_of\_pieces}$$

La taille d'un bloc est définie une valeur choisie lors de l'implémentation qui ne dépend pas de la taille fixe de la pièce. Une fois une taille fixe définie, le nombre de blocs par pièce peut être calculé en utilisant la formule:

$$\begin{aligned} \text{number\_of\_blocks} = & (\text{fixed\_piece\_size} / \text{fixed\_block\_size}) \\ & + !!(\text{fixed\_piece\_size} \% \text{fixed\_block\_size}) \end{aligned}$$

## Le fichier de metainfo

Le fichier metainfo fournit au client des informations sur la localisation du tracker ainsi que le torrent à télécharger. Outre la liste des fichiers qui se traduira par le téléchargement de torrent, il répertorie également comment le client doit séparer et vérifier les pièces individuelles qui composent le torrent complet.

## Le tracker HTTP

Le protocole HTTP Tracker (THP) est un mécanisme simple pour introduire des pairs entre eux. Un tracker est un service HTTP qui doit être contacté par un pair pour rejoindre un réseau de pairs. En tant que tel le tracker constitue le seul élément centralisé en BTP / 1.0. Un tracker ne fournit pas par lui-même l'accès à toutes les données téléchargeables. Un tracker pairs repose sur l'envoi des demandes régulières. Il peut supposer qu'un pair est mort s'il manque une requête.

## Le protocole « Peer Wire »

Le but du PWP, est de faciliter la communication entre pairs voisins dans le but de partage de fichiers. PWP décrit étapes prises par un pair après qu'il a lu dans un fichier metainfo et contacté un tracker pour recueillir des informations sur d'autres pairs avec lesquels il peut communiquer. PWP est posée sur le dessus du protocole TCP et gère toute sa communication en utilisant des messages asynchrones.