

# Utilisation de BitTorrent

---

**INDEXATION DU DOCUMENT**

	<i>TITRE :</i> Utilisation de BitTorrent		
<i>ACTION</i>	<i>NOM</i>	<i>DATE</i>	<i>SIGNATURE</i>
RÉDIGÉ PAR	Asdrad Torres	1/5/05	

**SUIVI DU DOCUMENT**

INDICE	DATE	MODIFICATIONS	NOM
1	1/5/2005		
2	7/2/2015	Ajout de la section Distributed Hash Table	

## Table des matières

<b>1</b>	<b>Mise en garde</b>	<b>1</b>
<b>2</b>	<b>Principe du téléchargement</b>	<b>1</b>
2.1	Objectif . . . . .	1
2.2	Téléchargement en deux temps . . . . .	1
2.2.1	Fichier de description : le "torrent" . . . . .	1
2.2.2	Téléchargement à partir du torrent . . . . .	2
2.2.3	Trouver le torrent . . . . .	2
2.2.4	Résumons . . . . .	2
<b>3</b>	<b>Ça devrait marcher...</b>	<b>2</b>
3.1	Les problèmes génériques . . . . .	3
3.1.1	L'application . . . . .	3
3.1.2	Le fichier torrent . . . . .	3
3.1.3	L'accès à internet . . . . .	3
3.2	Problèmes spécifiques à BitTorrent . . . . .	3
3.2.1	Coordonnées invalides . . . . .	3
3.2.2	Partager . . . . .	3
<b>4</b>	<b>Publier</b>	<b>4</b>
4.1	Le traqueur (tracker) . . . . .	4
4.2	Création du torrent . . . . .	5
4.3	Publier un torrent . . . . .	5
4.4	Un serveur BitTorrent ? . . . . .	5
4.5	Publier le fichier-cible . . . . .	6
<b>5</b>	<b>Masquer la complexité de BitTorrent</b>	<b>6</b>
5.1	Au téléchargement . . . . .	6
5.2	À la publication . . . . .	6
<b>6</b>	<b>Petite histoire d'une diffusion</b>	<b>7</b>
6.1	Chronologie et cheminement . . . . .	7
6.2	Quelques commentaires . . . . .	9
<b>7</b>	<b>Précautions et problèmes</b>	<b>10</b>
7.1	Firewall et ports . . . . .	10
7.1.1	Téléchargement . . . . .	10
7.1.2	Partage . . . . .	10
7.2	Première publication . . . . .	10
7.3	Contrôler la bande passante . . . . .	11

---

---

<b>8</b>	<b>Distributed Hash Table (DHT)</b>	<b>11</b>
8.1	Motivations . . . . .	11
8.2	Principe de fonctionnement . . . . .	11
8.2.1	Précaution terminologique . . . . .	11
8.2.2	Répartir l'information . . . . .	11
8.2.3	Parager l'information . . . . .	12
8.2.4	Exploiter l'information . . . . .	12
8.2.5	Initialiser le processus . . . . .	12
8.2.6	DHT et tracker . . . . .	12
8.3	Magnets links . . . . .	12
8.4	Limites et optimisation de DHT . . . . .	13
8.4.1	Limites de DHT . . . . .	13
8.4.2	Optimisation de DHT . . . . .	13
<b>9</b>	<b>Glossaire</b>	<b>13</b>

---

## Table des figures

1	Publication et premier téléchargement . . . . .	7
2	On télécharge depuis deux sources . . . . .	8
3	L'éditeur ne diffuse plus le fichier-cible . . . . .	9

---

## 1 Mise en garde

Quelles que soient les qualités techniques (réelles ou imaginaires) de BitTorrent, ce système est tout sauf intuitif. Aussi, pour ne pas être dérouter, il faut impérativement assimiler son principe de base. On s'aperçoit alors que BitTorrent n'est pas très compliqué. Si vous ne comprenez pas BitTorrent, ce n'est pas parce que vous êtes trop bête ou pas assez calé en informatique. C'est probablement parce que vous l'abordez avec un a priori.

Ce document se veut autosuffisant, dans le cadre limité d'une introduction à l'utilisation simple et immédiate de Bittorrent. Pour une description complète, la meilleure source d'information que j'ai trouvée est une [faq](#) en anglais, à la fois sobre, exhaustive et dépourvue d'erreurs !

## 2 Principe du téléchargement

### 2.1 Objectif

Pour comprendre le principe de fonctionnement de BitTorrent, il faut d'abord en comprendre l'objectif. Ce système a été conçu pour faciliter la diffusion massive d'un même "gros" fichier informatique, sur une période de temps limitée. Il se distingue donc des systèmes prévus pour permettre à de nombreuses personnes d'échanger des fichiers très variés (ou de petits fichiers) tels que FTP ou eDonkey. Pourtant, faute de mieux, il arrive souvent que l'on utilise ces systèmes pour assurer une distribution en grand nombre de gros fichiers (700 Mo ou plus).

Le cas le plus caricatural est celui où l'on utilise un serveur FTP pour diffuser une nouvelle version de système d'exploitation ou une mise à jour de logiciel. Pendant les jours qui suivront la sortie de cette nouvelle version, des milliers d'utilisateurs vont vouloir télécharger le même fichier ! Le résultat est connu : saturation du serveur FTP, refus de connexion en FTP anonyme, temps de téléchargement prohibitifs. La parade ? Placer, dès de début, des copies du fichier sur plusieurs serveurs FTP (appelés miroirs). L'inconvénient de cette parade ? Elle augmente les coûts de la distribution du fichier puisque c'est celui qui met le fichier à disposition qui doit, d'une manière ou d'une autre, "payer" les serveurs miroirs et la bande passante consommée.

La réponse de BitTorrent est de faire en sorte que chaque personne qui aura téléchargé le fichier devienne une nouvelle source de téléchargement. Cette idée n'est pas révolutionnaire puisqu'on la trouve déjà dans de nombreux logiciel de PeerToPeer. BitTorrent prétend simplement le faire plus efficacement que d'autres systèmes (Kazaa, eDonkey, etc.), dans le cas de gros fichiers fortement demandés..

### 2.2 Téléchargement en deux temps

Ce qui déroute le plus dans BitTorrent c'est que *tout téléchargement se fait en deux temps*. Dans un premier temps, on télécharge un fichier de description et d'orientation. Dans un second temps, on utilise les informations contenues dans ce fichier pour télécharger le fichier qui nous intéresse vraiment. Il y a donc toujours un téléchargement préalable au "vrai" téléchargement.

#### 2.2.1 Fichier de description : le "torrent"

Pour pouvoir télécharger quoi que ce soit avec BitTorrent, il faut *préalablement* se procurer un fichier décrivant ce que l'on veut charger. Ce fichier de description <sup>1</sup> doit être au format BitTorrent, c'est-à-dire qu'il ne peut être lu que par le logiciel BitTorrent, exactement de la même manière qu'un fichier PDF doit être lu avec un logiciel de lecture PDF (gohstview, acrobat reader, etc.). Ce fichier, que l'on appelle un "torrent" (comme on dit un "pdf"), se présente généralement avec un nom se terminant par ".torrent" (par ex., "mon\_fichier.torrent").

Comme tout fichier informatique (un pdf, un jpeg, un mp3), un *torrent* peut être diffusé par n'importe quel système d'échange. On peut donc télécharger un torrent via le web, un serveur FTP, un système P2P ou en pièce-jointe d'un courriel. Ce fichier étant de petite taille <sup>2</sup>, on peut le diffuser efficacement et rapidement, même avec un système de distribution classique.

Une fois que l'on a récupéré le torrent du fichier que l'on désire, on peut enfin procéder au téléchargement effectif. Le torrent est une sorte de fichier-bookmark (signet) dans lequel serait indiqué l'emplacement du fichier-cible, quelque part dans l'internet.

---

1. Également appelé fichier de *métadonnées*.

2. La taille du fichier de description est très petite en comparaison de celle du fichier décrit. Par exemple, le torrent d'un fichier de 600 Mo occupera quelques dizaines de Ko.

### 2.2.2 Téléchargement à partir du torrent

On n'a pas à se demander où se trouve réellement, sur l'internet, le fichier qui nous intéresse (fichier-cible). On n'a donc pas besoin de connaître son "adresse" ou son URL. Cette information est inscrite dans le torrent décrivant ce fichier. Il suffit donc de demander à l'application BitTorrent d'aller récupérer cette information dans le torrent et de procéder au téléchargement, conformément aux indications récupérées. Une application BitTorrent fait donc deux choses bien différentes : elle déchiffre le contenu d'un fichier torrent qu'on lui fournit et elle télécharge le (ou les) fichier(s) décrit(s) dans ce torrent<sup>3</sup>.

On retiendra donc que lancer le téléchargement d'un fichier distribué via BitTorrent revient à **lire** le torrent avec une application - compatible - BitTorrent

### 2.2.3 Trouver le torrent

Admettons que le torrent permette au logiciel BitTorrent de trouver le fichier qu'il décrit. Mais comment fait-on pour trouver le torrent correspondant au fichier-cible qui nous intéresse ? En effet, au départ, on sait simplement que l'on veut récupérer un fichier, par exemple, l'image du CD d'installation de Knoppix. Pour la télécharger via BitTorrent, il nous faut impérativement trouver un torrent décrivant ce fichier-image. L'astuce ? On la connaît déjà : on procédera comme on le fait pour toute recherche sur internet... Si nos recherches aboutissent à un fichier appelé `knoppix.iso`, il s'agira probablement du fichier source (ici, une image iso servant à graver un CD). S'il s'appelle `knoppix.iso.torrent`, on comprendra qu'il s'agit du torrent du fichier `knoppix.iso`. Dans le premier cas, on pourra le télécharger directement mais avec les risques habituels d'encombrement. Dans le second on pourra utiliser le système BitTorrent.

---

#### Note

Il existe un **moteur de recherche spécialisé** dans la recherche de torrent. Mais rien ne garantit que le torrent qui vous intéresse y sera recensé.

---

### 2.2.4 Résumons

1. Au départ, on cherche un fichier que l'on souhaiterait télécharger.
2. Par nos moyens habituels, on trouve un torrent décrivant ce fichier
3. En utilisant un logiciel standard, on télécharge le torrent. Le logiciel utilisé pour récupérer le torrent peut être un navigateur web, un logiciel ftp, une application eDonkey ou kazaa, etc.).
4. On "lit" le torrent avec un logiciel BitTorrent. Cette lecture déclenche le téléchargement du fichier "cible".



#### Attention

On voit bien que les mots sont piégés. Ce qu'on appelle "lire" le torrent est, en fait, le téléchargement du fichier qui nous intéresse. Ce qu'on appelle "télécharger" le torrent, ne concerne que le fichier de description.

---

## 3 Ça devrait marcher...

Présenté comme ça, le plus difficile ne semble pas être l'utilisation de BitTorrent (la lecture du torrent) mais les étapes précédentes. C'est d'ailleurs ce qui fait dire à certains que BitTorrent est un système simple d'emploi. Il ne semble pas plus compliqué de lire un fichier torrent que de lire un fichier jpeg, pdf ou html ; c'est juste un peu plus long (le temps du téléchargement)... Il suffit d'avoir l'application reconnaissant le type de fichier voulu. Certes, mais quiconque a déjà eu des mésaventures en essayant de lire un fichier pdf, sait que lire parfaitement un fichier pdf peut ne pas être trivial. Alors un torrent...

---

3. Puisqu'un torrent est un fichier de description, il peut parfaitement décrire plusieurs fichiers "cibles". C'est pratique car lorsqu'on lancera la "lecture" du torrent, cela enclenchera le téléchargement de tous fichiers qui y sont décrits.

---

## 3.1 Les problèmes génériques

### 3.1.1 L'application

Première source possible de problème : l'application BitTorrent elle-même. Comme tout logiciel, la version peut être boguée, inadaptée ou mal installée. Contre ce genre de problème, il n'y a malheureusement pas de recette miracle : se renseigner, lire les fichiers "lisez-moi", ne pas installer n'importe quoi...

### 3.1.2 Le fichier torrent

L'autre source possible de problème est le fichier torrent. Puisque c'est un fichier que l'on a - généralement - récupéré sur internet, on n'est pas sûr de sa qualité. Heureusement, le format torrent contient une signature électronique qui permet de contrôler l'intégrité du fichier torrent. Cela garantit que le torrent récupéré est bien, au bit près, celui qui a été mis en distribution.



#### Attention

Cela ne signifie pas qu'il a été correctement fabriqué, ni que les informations qu'il contient sont sensées.

---

Toute application BitTorrent bien faite vérifiera l'intégrité d'un torrent avant d'essayer d'en interpréter le contenu et signalera toute erreur d'intégrité au lieu de "planter".

### 3.1.3 L'accès à internet

Comme expliqué précédemment, la lecture du fichier torrent par l'application BitTorrent déclenche le téléchargement du fichier-cible. Évidemment, cela ne peut fonctionner que si l'ordinateur dispose d'un accès à internet.

Comme toute application internet (web, ftp, chat...), BitTorrent utilise un canal de communication (port) spécifique. Si un *firewall* est interposé entre l'application BitTorrent et l'internet, ce canal sera généralement fermé. Il faudra donc procéder à son ouverture.

Le téléchargement par BitTorrent est capable de fonctionner derrière un firewall interdisant toute connexion **entrante**. Généralement, un firewall domestique autorise toutes vos connexions sortantes<sup>4</sup> et filtre les connexions entrantes. Tout devrait donc bien se passer, sans avoir à modifier les réglages de votre firewall.

Si vous êtes sur un réseau professionnel ou si vous avez réglé votre firewall de manière paranoïaque, il faut vous assurer que vous pouvez ouvrir des connexions sortantes sur les ports TCP 6969 et 6881 (et plus<sup>5</sup>). Votre vitesse de téléchargement sera meilleure si vous ouvrez également le port 6881 (et plus) aux connexions entrantes.

## 3.2 Problèmes spécifiques à BitTorrent

### 3.2.1 Coordonnées invalides

Pour que BitTorrent puisse retrouver le fichier, il faut bien que, d'une manière ou d'une autre, le torrent lui indique l'emplacement (du ou) des fichiers-cibles qu'il décrit. Et oui, il n'y a pas de miracle ! Si les coordonnées sont invalides ou si les serveurs indiqués par ces coordonnées sont arrêtés ou saturés, le téléchargement ne pourra pas se faire. C'est comme si on avait récupéré la mauvaise adresse d'un page web ou la bonne adresse d'un serveur surchargé.

### 3.2.2 Partager

On peut utiliser BitTorrent pour télécharger sans partager. Autrement dit, si vous n'arrivez pas à télécharger quoique ce soit, ça ne peut pas être dû au fait que vous ne partagez rien. En revanche, la vitesse à laquelle vous téléchargez sera modulée selon que vous partagiez ou non.

Comme le téléchargement, le partage ne peut fonctionner que si les canaux (ports) utilisés par BitTorrent sont ouverts aux autres utilisateurs (ordinateurs). Si l'utilisateur se trouve derrière un firewall, ces ports sont généralement fermés, par défaut.

---

4. Autorise à *aller sur* internet (web, mail, etc.).

5. BitTorrent utilise un port par téléchargement. Si vous ne téléchargez qu'un fichier à la fois, il suffit d'ouvrir le port 6881. Si vous voulez télécharger deux fichiers simultanément, il vous faudra ouvrir les ports 6881 et 6882. Et ainsi de suite... Pour lancer 10 téléchargement simultanés il faut ouvrir la plage 6881-6890.

---

## 4 Publier

Si vous pouvez télécharger des fichiers par BitTorrent c'est forcément parce que certaines personnes ont publié ces fichiers. Même si vous ne souhaitez pas publier, vous lirez avec intérêt comment on procède pour publier sur BitTorrent. Cela vous permettra de mieux comprendre comment fonctionne le téléchargement.

### 4.1 Le traqueur (tracker)

Tout fichier que l'on veut rendre accessible via BitTorrent doit être répertorié par un tracker.

**Attention**

Le tracker était indispensable aux débuts de BitTorrent. Ce n'est plus le cas aujourd'hui. Il existe des techniques permettant de se passer de tracker. À la fin de cet article, une section présente une de ces techniques : DHT (Distributed Hash Table).

---

Ce tracker est un programme de surveillance, implanté sur un ordinateur accessible à une adresse prévisible<sup>6</sup>. Il faut savoir que le tracker peut se trouver sur un ordinateur différent de celui qui héberge la copie originale du fichier que l'on souhaite publier. Il faut donc, soit activer un tracker sur son propre ordi, soit faire appel à un tracker *public*, c'est-à-dire un tracker qui accepte de référencer d'autres fichiers que ceux qu'il détient lui-même.

Au cours du processus de distribution<sup>7</sup>, le fichier original va se voir dupliqué sur un nombre croissant d'ordinateurs. Le tracker en sera informé et pourra donc indiquer, à tout moment, sur quels ordis trouver des fragments du fichier publié. Ainsi, toute nouvelle tentative de téléchargement pourra être orientée vers l'un quelconque des ordinateurs possédant le fragment demandé. Plusieurs fragments pouvant être téléchargés simultanément, on peut ainsi parvenir à un débit de téléchargement élevé, même si chacun des ordinateurs sollicités ne délivre qu'un faible débit.

**Attention**

Le canal (port) par défaut utilisé par les trackers BitTorrent est 6969.

---

Pour le lancement concret d'un tracker, reportez-vous au manuel d'utilisation de votre application BitTorrent. Sur un système Linux-Mandrake équipé du logiciel standard, la commande de lancement du tracker est:

```
[asdrad@test asdrad]$ bttrack.py --port 6969 --dfile dstate
```

Le paramètre `dstate` indique l'endroit où sera mémorisé l'état du tracker. Cela permet de récupérer l'information de "traque" après un arrêt du tracker.

**AVERTISSEMENT**

Si vous êtes derrière un firewall, il faut que le port indiqué dans cette commande (ici, 6969) soit ouvert aux requêtes entrantes.

---

Si vous avez utilisé la commande indiquée plus haut, votre tracker sera public. Cela signifie que n'importe qui, sur l'internet, peut l'utiliser pour assurer la diffusion de n'importe quel contenu. Si vous souhaitez limiter l'utilisation de ce tracker, il vous suffit d'indiquer un répertoire (sur la machine hébergeant le tracker) dans lequel devra figurer une copie des torrents autorisés. La commande prend alors la forme suivante :

---

6. Une adresse *prévisible* est soit une adresse IP fixe, soit un nom de domaine. Si l'ordinateur sur lequel tourne le *tracker* n'a qu'une adresse dynamique (cas de nombreux fournisseurs ADSL), on peut s'en tirer grâce à un système de domaine dynamique (<http://www.dyndns.org>).

7. Au fur et à mesure que des internautes téléchargent le fichier via BitTorrent.

---

```
[asdrad@test asdrad]$ bttrack.py --port 6969 --dfile dstate --allowed_dir mon_repertoire
```

Comment contrôler l'accès d'autres utilisateurs, en réseau, à ce répertoire "mon\_repertoire" dépasse le cadre de cette documentation. Pour une utilisation limitée, le plus simple est de demander à ce que l'on vous envoie le torrent, en pièce-jointe d'un courriel. Vous le placerez vous-mêmes, si vous en êtes d'accord, dans "mon\_repertoire".

---

**Note**

Les personnes intéressées par l'installation et la configuration d'un tracker liront avec intérêt "[Monter un tracker BitTorrent sur Linux](#)" qui explique comment ajouter une fonction "tracker" à un serveur web grâce à [PHPBTTracker+](#).

---

## 4.2 Création du torrent

Comme on l'a répété depuis le début, la particularité de BitTorrent se trouve dans le fichier de description du fichier-cible : le torrent. Pour que des utilisateurs puissent télécharger ce *torrent* (qui leur permettra de télécharger le fichier-cible), il faut bien que quelqu'un l'ait créé.

Pour la création concrète d'un torrent, reportez-vous au manuel d'utilisation de votre application BitTorrent. Sur un système Linux-Mandrake équipé du logiciel standard, la commande de création d'un torrent est :

```
[asdrad@test asdrad]$ btmakemetafile.py fichier_cible.ext http://my.tracker:6969/announce
```

On constate que le torrent créé fait référence au tracker sur lequel sera référencée la publication. Si ce tracker est en panne, il est clair que tout internaute qui tentera de télécharger le `fichier_cilbe` se cassera les dents.

La commande précédente va créer un fichier dénommé `fichier_cible.ext.torrent`. Enfin, on a créé le torrent ! Mais on est le seul à le savoir. Il va falloir le publier.

## 4.3 Publier un torrent

Maintenant que l'on dispose du fichier de description, il suffit de le mettre à disposition sur un serveur web, par exemple. Mais on pourrait aussi bien le mettre sur un serveur ftp ou eDonkey. L'essentiel est qu'il soit accessible aux internautes. Si l'on a opté pour un serveur web, le navigateur proposera probablement d'enregistrer le fichier torrent car il sera bien incapable de l'afficher. C'est le comportement habituel d'un navigateur vis-à-vis d'un fichier de type inconnu.

L'avantage à mettre le torrent sur un serveur web découle de ce que les internautes qui disposent d'un navigateur correctement configuré pourront traiter automatiquement le torrent. Encore faut-il que le serveur web soit, lui aussi correctement configuré. Ainsi, l'administrateur du serveur prendra soin d'indiquer à son serveur qu'il doit associer le type mime `application/x-bittorrent` à tous les fichiers `*.torrent`. Le navigateur de l'internaute qui téléchargera le torrent saura ainsi de quel type est ce fichier. Si, par ailleurs, il a indiqué à son navigateur le nom de l'application BitTorrent qui devait être utilisée pour "afficher" (lire, interpréter) un torrent, cette application sera automatiquement lancée et le téléchargement du fichier-cible amorcé.

Ce comportement est exactement le même que celui que l'on observe lorsqu'on clique sur un lien pointant vers un fichier pdf. Le navigateur se rend compte qu'il ne sait pas afficher ce type de fichier, reconnaît que c'est un pdf et appelle le plug-in ou l'application externe dont il sait qu'il peut lire les pdf. Or, n'oublions pas que *charger* le fichier cible décrit par un torrent est obtenu en faisant *lire* ce torrent par une application compatible BitTorrent.

## 4.4 Un serveur BitTorrent ?

Il découle de ce qui précède qu'on n'a pas besoin de logiciel "serveur" pour mettre à disposition des fichiers par BitTorrent. Il n'y a donc pas d'application serveur pour BitTorrent. On utilise la même application pour télécharger et pour publier. Dès que l'on télécharge, on devient co-distributeur des frangements du fichier que l'on a déjà chargés.

Pour les ordi se trouvant derrière un "firewall" ou un routeur, il n'y a pas d'impossibilité technique majeure. BitTorrent peut fonctionner dans ces conditions. Mais certains utilisateurs ne coopéreront qu'avec les ordi permettant un accès direct à leur machines. Seuls ceux-là jouent pleinement le jeu du partage. Pour se trouver dans cette situation, il faut ouvrir les connexions *entrantes* arrivant sur les port TCP 6881 à 6999. Si votre ordi est sur un réseau local où l'accès à internet est partagé, il faut indiquer à votre routeur de faire suivre ces requêtes sur l'ordi exploitant BitTorrent.

---

## 4.5 Publier le fichier-cible

Jusqu'ici, non n'avons publié que le torrent. Que se passera-t-il si un internaute arrive sur notre site web et clique sur lien pointant vers notre torrent ? Il pourra télécharger le torrent, mais c'est tout. S'il tente de faire lire ce torrent par son application BitTorrent, il ne se passera rien. En effet, l'application ira interroger le tracker déclaré dans le torrent, mais ce dernier sera bien incapable de lui indiquer où se trouvent les fragments du fichier-cible.

Pour boucler la boucle, il nous reste donc à publier le fichier-cible. C'est là que l'on touche à la grande spécificité de BitTorrent. Étant donné que le système du tracker est conçu pour tenir à jour la liste des copies disponibles, il ne fait pas vraiment de différence entre la copie originale et les copies suivantes. Pour l'informer de l'existence de cet original, on va utiliser l'application BitTorrent pour lui faire croire que l'on vient d'en terminer le téléchargement. Autrement dit, on lance, sur le poste de publication, une commande de téléchargement en indiquant le fichier-cible préexistant comme fichier de destination. C'est tordu, mais c'est bien ça qu'il faut faire...

En supposant que l'on ait publié notre torrent sur un serveur web, on utilisera une commande de téléchargement du genre :

```
[asdrad@test asdrad]$ btdownloadheadless.py --url http://monserveurweb.org/fichier_cible. ←  
ext.torrent --saveas fichier_cible.ext
```

Par cette commande on ordonne à l'application de procéder à la lecture du torrent. Or, l'application va se rendre compte que le téléchargement qui lui est demandé est terminé. En effet, elle va s'apercevoir que le fichier `fichier_cible.ext` possède exactement le même nom, la même taille et la même signature électronique que le fichier référencé dans le torrent<sup>8</sup>. Il va donc signaler au tracker (dont l'adresse figure dans le torrent), qu'il dispose d'une copie complète du fichier-cible. Le tracker va mémoriser cette information, si bien que la prochaine fois qu'un internaute lira une copie de notre torrent et interrogera le tracker, ce dernier lui indiquera au moins un endroit où trouver une version intégrale de notre fichier-cible.

On constate que si le téléchargement d'un fichier par BitTorrent n'est pas intuitif, la publication l'est encore moins. Si l'on résume le principe de publication on peut dire : pour publier un fichier, on le télécharge... Même si l'on a bien compris la logique du système, il est difficile de faire plus déroutant !

## 5 Masquer la complexité de BitTorrent

BitTorrent est un système astucieux mais complexe. On ne pourrait pas le simplifier sans le dénaturer mais on peut en masquer la complexité.

### 5.1 Au téléchargement

On l'a vu, une bonne configuration d'un navigateur web peut totalement masquer la complexité de BitTorrent. Pour cela, il faut que :

- l'ordi de l'utilisateur soit équipé d'une application de lecture de torrent (téléchargement de fichiers-cibles),
- le navigateur soit configuré pour reconnaître les fichier \*.torrent et/ou le type mime `application/x-bittorrent`,
- le navigateur (ou le système d'exploitation) soit configuré pour faire appel à l'application de lecture de torrent dès qu'on tente de lire un fichier torrent.

Moyennant ces précautions, l'utilisateur peut ne même pas se rendre compte qu'il utilise BitTorrent. Lorsqu'il cliquera sur un lien pointant vers un torrent, son navigateur passera automatiquement la main à la bonne application qui déclenchera automatiquement le téléchargement du fichier-cible.

### 5.2 À la publication

Simplifier la publication est une autre paire de manches... Il faut obligatoirement disposer d'une application spécialement développée à cet effet.

Un logiciel libre tel que [Azureus](#)<sup>9</sup> propose une interface graphique permettant de faciliter la publication. Il permet également de récupérer des torrent, de les lire (télécharger les fichiers-cibles) et de surveiller le déroulement du téléchargement.

8. Normal, puisque le torrent a été fabriqué à partir de ce fichier...

9. Logiciel écrit en Java, sous licence GNU GPL. Il fonctionne sous Linux, MacOS X et Windows.

## 6 Petite histoire d'une diffusion

### 6.1 Chronologie et cheminement

Récapitulons, à travers des graphiques, le cheminement d'un fichier diffusé via BitTorrent.

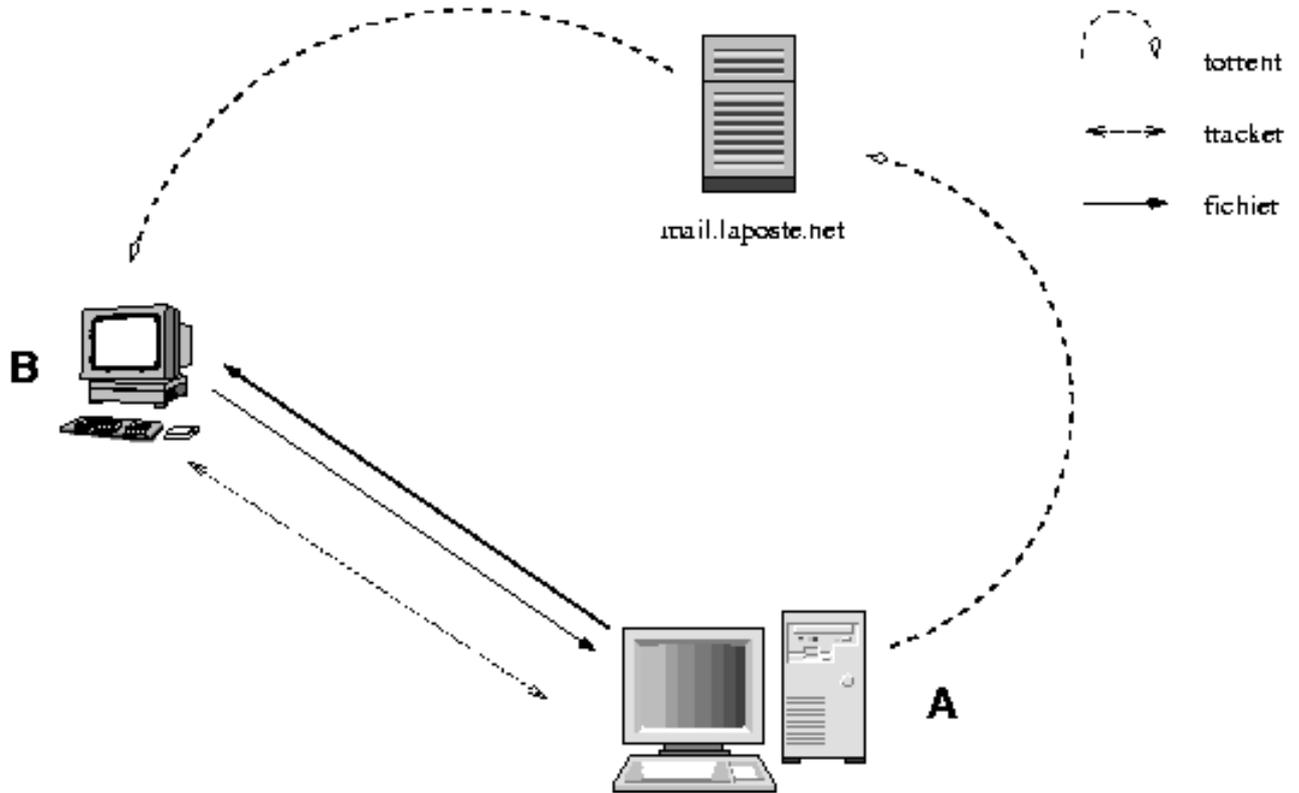


FIGURE 1 – Publication et premier téléchargement

1. La personne jouant le rôle d'éditeur active un tracker sur son ordi (A).
2. Elle crée le torrent du fichier-cible qu'elle souhaite publier (ici, un fichier contenant une vidéo au format mpeg4)..
3. Elle lance le pseudo-téléchargement de son fichier-cible
4. Elle place le torrent sur ses pages Web, hébergées chez son fournisseur d'accès
5. Elle envoie à ses ami-e-s une copie du torrent, en pièce-jointe d'un courriel
6. Une amie (B) télécharge le fichier-cible grâce à Azureus.

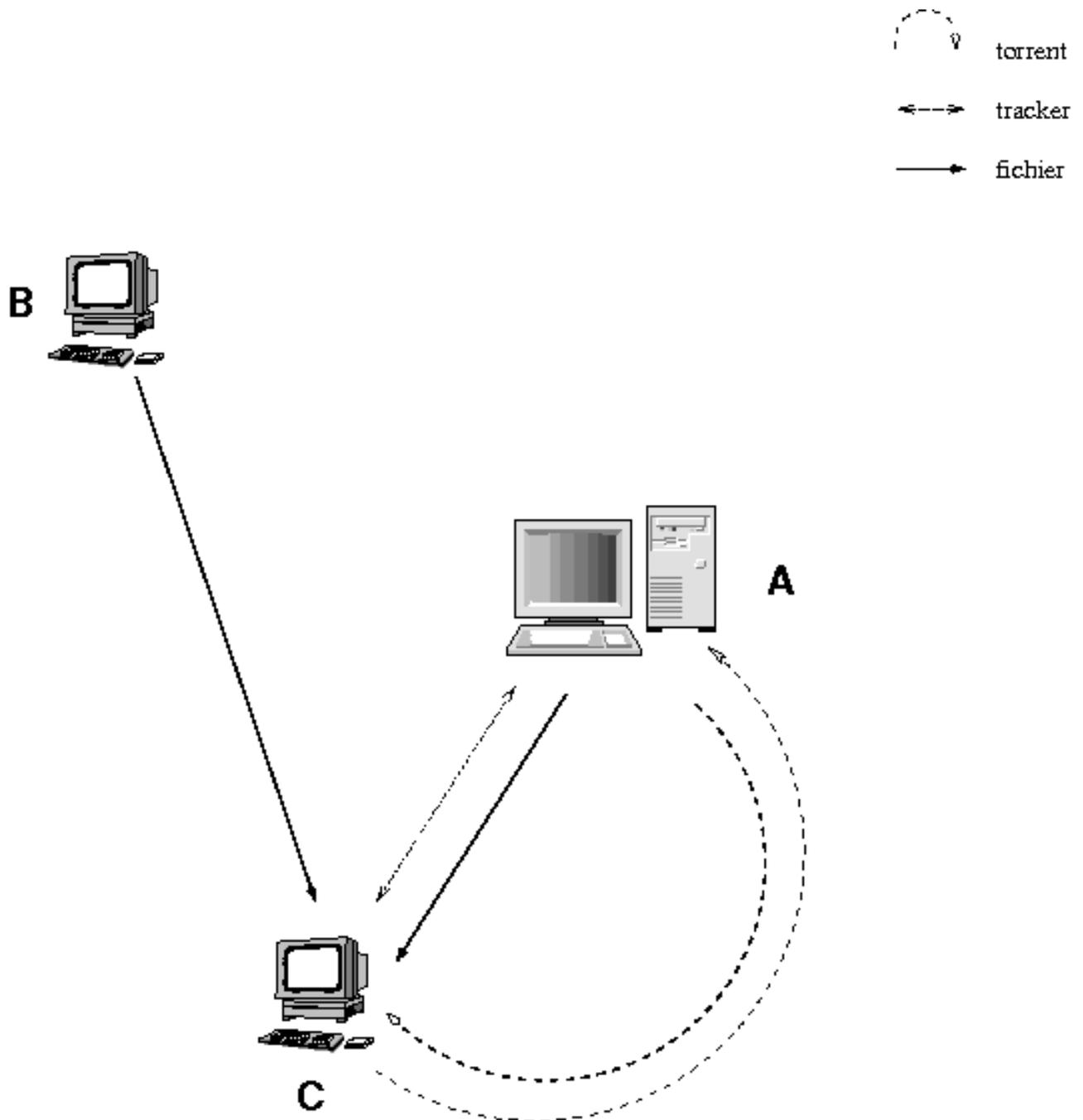


FIGURE 2 – On télécharge depuis deux sources

7. Un internaute (C) tombe sur la page de l'éditeur et lance directement (depuis un navigateur correctement configuré), le téléchargement du fichier-cible. Il télécharge simultanément depuis l'ordi de l'éditeur et celui de l'amie.

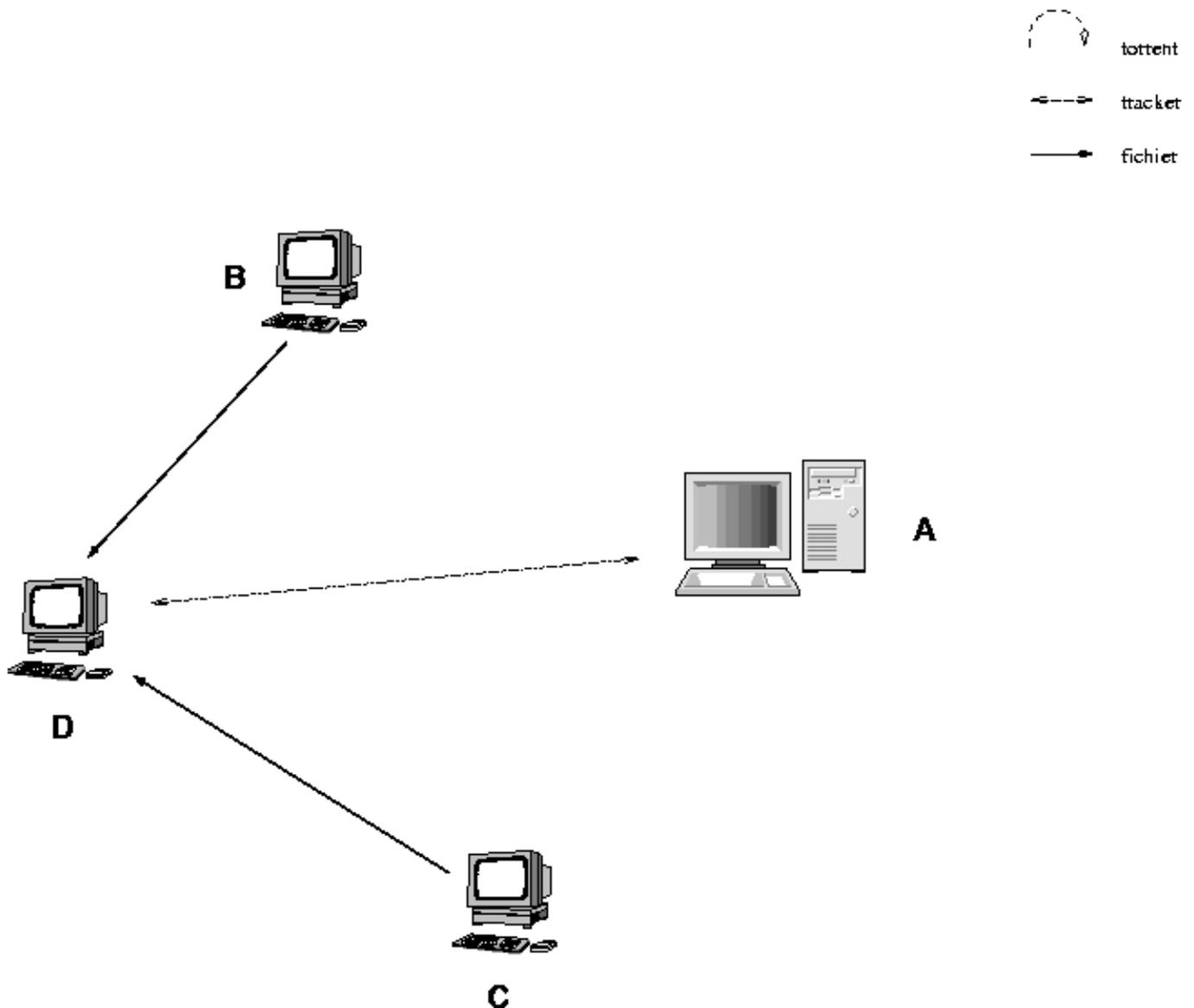


FIGURE 3 – L'éditeur ne diffuse plus le fichier-cible

8. L'éditeur arrête son pseudo-chargeement, mais laisse le tracker en marche
9. Un nouvel internaute (D) télécharge le fichier-cible depuis les ordis de l'amie et premier internaute.

## 6.2 Quelques commentaires

Étant donné que le torrent fait référence au tracker de l'éditeur, ce tracker doit être disponible en permanence. Cela signifie que l'ordi est allumé, la connexion internet active, l'application tracker lancée, l'adresse internet de l'ordi stable. En gros, l'ordi de l'éditeur fonctionne comme un serveur.

Le premier téléchargement se fera forcément à partir du fichier original de l'éditeur. Les téléchargements suivants pourront profiter des copies faites par les utilisateurs. Bien sûr, cela ne sera possible que si les utilisateurs laissent active leur application BitTorrent de même que leur accès à internet. Si les internautes ne partagent pas, l'éditeur devra supporter seul la charge de tous les téléchargements.

Comme on l'a vu aux étapes 8 et 9, dès que quelques copies ont été effectuées, l'éditeur n'a plus besoin d'assurer lui-même la distribution de l'original. Il se contente de maintenir le tracker. Or, à la différence d'un téléchargement, l'interrogation du tracker

ne consomme que très peu de bande passante et de puissance. L'ordi et la connexion internet de l'éditeur sont disponibles pour d'autres tâches.

S'il avait utilisé un tracker public, l'éditeur aurait pu éteindre son ordi. On aurait alors assisté au scénario idéal pour lequel BitTorrent a été conçu. L'éditeur publie un fichier. Tant qu'il est la seule source disponible, il supporte seul l'effort des premiers téléchargements. Dès qu'un nombre suffisant de sources alternatives ont été créées, ce sont les internautes eux-mêmes qui se répartissent la charge des téléchargements. Plus le fichier est diffusé, plus cette charge est répartie, moins elle est pesante. Un cercle vertueux est enclenché

## 7 Précautions et problèmes

Quand rien en marche, qu'on n'arrive même pas à télécharger un seul fichier-source, on ne sait pas par où prendre le problème. Je conseille vivement d'utiliser un logiciel tel que [Azureus](#). Certes, cela suppose que l'utilisateur a préalablement installé Java. Mais l'effort sera indubitablement rentabilisé, à court ou moyen terme.

### 7.1 Firewall et ports

L'une des première cause de problème est le firewall. Comme toujours, il faut bien faire la différence entre les requêtes sortantes et les requêtes entrantes. Un firewall est susceptible de bloquer les deux types de requêtes.

#### 7.1.1 Téléchargement

Au minimum, le firewall doit laisser passer les requêtes BitTorrent sortantes. Cela permet à l'application BitTorrent d'interroger le tracker puis de télécharger le fichier-cible depuis les sources proposées par le tracker. Cela est insuffisant pour permettre aux autres internautes de télécharger depuis notre ordi.

Pour vérifier que le téléchargement fonctionne, on ne s'escrimera pas à lire un torrent particulier. Au contraire, on essaiera de lire des torrent faisant référence à des fichiers dont tout permet de penser qu'ils sont réellement publiés. On n'hésitera donc pas à essayer de télécharger des fichiers, a priori sans intérêt.

#### 7.1.2 Partage

Afin de jouer le jeu du partage, il faut que notre firewall laisse également passer les requêtes entrantes. Ces requêtes sont adressées depuis les trackers ou depuis les applications de téléchargement. Elles permettent au trackers de savoir quels fichiers nous mettons à disposition et aux applications de téléchargement de récupérer des fragments de fichier-cible. Pour que le partage fonctionne, nous devons impérativement laisser tourner l'application BitTorrent (ici Azureus). Garder l'ordinateur allumé et la connexion internet active n'est pas suffisant.

Lors de l'installation d'Azureus, le logiciel permet de tester le bon fonctionnement du port utilisé pour le partage. C'est bien pratique car sinon on ne peut que constater que personne ne télécharge depuis notre ordi. Est-ce parce que nos publications n'intéresse personne ? Parce que notre partage ne fonctionne pas ? Parce que notre partage fonctionne mais que le firewall en bloque l'accès aux internautes ? Voilà le genre de question auxquelles Azureus vous aidera à répondre. En écartant ou en incriminant, d'entrée de jeu, le firewall vous vous économiserez bien des interrogations.

### 7.2 Première publication

Lorsqu'on veut publier un fichier pour la première fois, il ne faut pas oublier de lancer le pseudo-téléchargement. D'accord, mais quand peut-on arrêter ce pseudo-chargeement ? Tout dépend du comportement des internautes qui téléchargeront des copies de ce fichier. S'ils jouent le jeu du partage, vous pouvez cesser de "partager" votre fichier dès qu'un nombre suffisant de copies ont été faites **et rendues disponibles**. Sinon, vous serez obligé de maintenir ce téléchargement actif aussi longtemps que vous voudrez rendre votre fichier accessible.

Même si l'on cesse de partager le fichier, il ne faut pas oublier de laisser tourner le tracker. Sauf si l'on a déclaré notre fichier à un tracker public, on est condamné à laisser tourner notre tracker tant que l'on veut assurer la publication effective du fichier. N'oublions que le tracker est inscrit dans le torrent de notre fichier. Donc toute personne qui tentera de télécharger ce fichier interrogera son tracker.

## 7.3 Contrôler la bande passante

Lorsqu'on partage des fichiers, on risque de se voir sa bande passante totalement consommée par les internautes qui téléchargent sur notre ordi. Le système BitTorrent ne s'occupe pas de ce problème mais les applications BitTorrent peuvent le prendre en charge, plus ou moins bien.

Ainsi, si vous utilisez l'application standard sous Linux-Mandrake, vous pourrez uniquement contrôler le nombre de personnes autorisées à télécharger un même fichier. Mais si ces personnes possèdent des accès à haut débit, vous verrez rapidement votre flux montant totalement absorbé par eux. Pour limiter cette consommation, vous seriez obligé d'aller farfouiller dans les entrailles du système... Cela peut sembler pénalisant mais c'est le seul moyen de faire quelque chose d'aussi simple, en apparence, que d'allouer 50% du flux montant à BitTorrent quand vous utilisez votre accès internet et 100% le reste du temps.

Heureusement, des logiciels comme Azureus vous permettent, très simplement, de paramétrer votre partage. Vous pourrez fixer un débit maximum autorisé pour les internautes qui téléchargent. C'est mieux que rien mais ça reste rudimentaire. Quand vous n'utilisez pas internet, la bande passante non allouée explicitement à Azureus est perdue alors qu'elle aurait pu vous permettre de mieux diffuser vos fichiers.

## 8 Distributed Hash Table (DHT)

### 8.1 Motivations

Dans le fonctionnement originel de BitTorrent, un fichier ne peut être téléchargé que s'il existe un `tracker` tenant à jour, en temps réel, la liste des peers détenant des fragments du fichier souhaité (swarm). Ce principe de fonctionnement entraîne deux contraintes pénalisantes pour un système qui tend à fonctionner de manière horizontale (de pair à pair).

D'une part, pour qu'un tracker tienne à jour en temps réel la liste des membres d'un swarm, il doit être opérationnel en 24/7 ! Si le tracker est arrêté, il sera impossible de lancer le téléchargement d'un fichier référencé par ce tracker, même si, à ce moment là, il existe des milliers de peers actifs disposés à partager ce fichier.

D'autre part, le tracker joue un rôle de centralisation de l'information. Il devient ainsi le maillon faible du système, aussi bien du point de vue technique que juridique.

D'où l'idée de répartir l'information détenue par le tracker, comme cela se fait pour les contenus des fichiers-cibles sont répartis entre les peers. **DHT** est l'une des technologies permettant d'atteindre cet objectif. Étant désormais une extension du protocole BitTorrent, DHT est disponible sur tout logiciel BitTorrent digne de ce nom.

### 8.2 Principe de fonctionnement

#### 8.2.1 Précaution terminologique

**DHT** est une technique d'échange "de pair à pair" (peer to peer). Dans le cas qui nous intéresse, **DHT intervient parallèlement à BitTorrent** qui est aussi un système de pair à pair. Jusqu'ici, nous avons appelé `peer`, un participant au réseau BitTorrent. Pour éviter la confusion, nous appellerons `node`, un participant à DHT. Au sens général du terme "pair", les peers et les nodes sont des pairs, puisqu'ils participent à un système de pair à pair. Mais comme ils remplissent des rôles différents, on leur donne des noms différents. Si vous activez, sur votre ordi, un logiciel BitTorrent moderne, vous êtes potentiellement à la fois peer et node.

#### 8.2.2 Répartir l'information

Avec DHT, l'information sur qui participe au partage de quel fichier est répartie entre tous les nodes ayant choisi de contribuer à DHT. Chaque node détient deux types d'information. D'un côté, il tient à jour une liste des nodes avec lesquels il coopère dans le cadre de DHT. De l'autre il dispose d'un registre indiquant quels sont les peers contribuant à l'échange des fichiers associés aux torrents connus de lui. Dans ce système, chaque torrent est identifié par une clé unique, sur tout le réseau.

### 8.2.3 Parager l'information

Connaissant la clé associée à un torrent, un node souhaitant télécharger les fichiers associés à ce torrent peut s'appuyer sur son propre registre ou demander aux autres nodes s'ils disposent d'informations sur cette clé, dans leur propre registre. De proche en proche, chaque node peut relayer la requête auprès des nodes qu'il connaît jusqu'à trouver un node détenant l'information recherchée. Si l'information est disponible par au moins un node actif au moment de la requête, elle finira par être trouvée et transmise au node requérant.

### 8.2.4 Exploiter l'information

Une fois qu'il connaît (au moins partiellement) le `swarn` impliqué dans le partage d'un fichier, un peer n'a plus qu'à effectuer le téléchargement auprès de ce `swarn`, exactement comme il l'aurait fait qu'il avait récupéré l'information depuis un tracker.

### 8.2.5 Initialiser le processus

Tout ça bien gentil comment fait-on pour connaître la clé associée à un fichier ? Comment entre-t-on en relation avec au moins un node participant à DHT ? Pour connaître la clé associée à un fichier, si on n'utilise pas de tracker, il n'y a pas de miracle. Soit quelqu'un-e nous l'a donnée, soit on la trouve sur internet. Nous verrons **plus loin** à quoi ressemble cette clé et comment on peut la transmettre.

Quant aux nodes, ils sont en partie un sous-ensemble des peers participant à BitTorrent. Tout peer dont on découvre l'existence (via un tracker ou s'il se connecte à nous pour télécharger), est un node potentiel. Au fil des téléchargements, un logiciel BitTorrent utilisant DHT interrogera les peers avec lesquels il entre en relation et verra s'ils participent ou non à DHT. Même en partant d'une liste vide, aussitôt le premier node détecté, notre liste des nodes connus se remplira rapidement.

### 8.2.6 DHT et tracker

DHT offre une solution alternative aux trackers. Grâce à cette technique, un torrent peut être accessible sans être catalogué par un tracker<sup>10</sup> et donc sans dépendre de lui. Pour autant, DHT et trackers ne sont pas mutuellement exclusifs. Nous avons même vu qu'un tracker un moyen simple de contacter des nodes potentiels et ainsi d'amorcer sa propre participation au réseau DHT.

Un même torrent peut être catalogué par un ou plusieurs trackers et être référencé dans DHT. Dans ce cas, les deux systèmes agiront de manière combinée pour obtenir une vision aussi complète que possible du `swarn` qui sera sollicité pour télécharger les fichiers associés au torrent.

## 8.3 Magnets links

Un `magnet link` est une manière d'exprimer la clé associée à un torrent, sous une forme purement textuelle. On peut alors la publier ou la partager à travers toute technologie acceptant le texte (web, mail, chat, etc.). Un magnet est une sorte d'URL (adresse web), par exemple :

```
magnet:?xt=urn:sha1:YNCKHTQCWBTRNJIV4WNAE52SJUQCZO5C
```

Le `magnet link` n'est pas, stricto sensu, un technique liée à DHT. Mais elle permet de désigner un torrent par sa clé. Or nous avons vu que c'était une **condition nécessaire** à l'utilisation de DHT.

En pratique, sur un ordi doté des logiciels nécessaires correctement configurés, un magnet est automatiquement reconnu. S'il apparaît comme un lien, dans une page affichée dans un navigateur web, un simple clic sur le lien activera le logiciel BitTorrent-DHT, et lui transmettra la clé. Ce dernier interrogera alors le réseau DHT pour trouver des peers et lancera le téléchargement s'il en trouve.

Le cadre d'utilisation des **magnets links** ne se limite pas DHT, ni même à BitTorrent.

---

10. On parle de `trackerless torrent`.

## 8.4 Limites et optimisation de DHT

### 8.4.1 Limites de DHT

DHT permet de partager des fichiers via BitTorrent sans dépendre d'un tracker en décentralisant les informations sur les peers susceptibles de contribuer à ce partage. Mais le référencement d'un fichier dans DHT n'est pas une garantie de disponibilité réelle des peers. DHT intervenant en amont du téléchargement par BitTorrent, il ne protège pas contre les aléas de ce système<sup>11</sup>.

De plus, DHT apporte ses propres faiblesses. En effet, chaque node est libre de déclarer ce qu'il veut, autrement dit d'injecter des informations obsolètes ou malveillantes dans le réseau d'échange.

### 8.4.2 Optimisation de DHT

Une des premières précautions intégrées à DHT est la qualification des nodes. Parmi tous les nodes auto-déclarés, DHT procède à une évaluation périodique de la fiabilité et de la disponibilité de chaque node.

L'autre problème à résoudre est la répartition de l'information. En effet, il n'est pas envisageable que chaque node ait une connaissance de la totalité de ce qui est disponible sur le réseau. Le swarm de chaque torrent est en évolution incessante et la seule mise à jour de cette information saturerait le réseau si elle devait être diffusée à chaque node. C'est pourquoi chaque node n'entre en relation directe qu'avec un nombre limité de nodes, considérés comme voisins. Cette limitation introduit une notion de distance entre nodes.

## 9 Glossaire

### Quelques définitions

---

**Note**

D'après <http://btfaq.com/serve/cache/23.html>

---

**torrent**

On appelle ainsi le fichier de description (métadonnées) qui contient les informations nécessaires pour trouver les copies disponibles d'un fichier-cible. En fait, un torrent peut contenir les descriptions de plusieurs fichiers. Dans ce cas, la lecture du torrent par une application BitTorrent lancera le téléchargement de tous fichiers.

Par abus de langage, on appelle également torrent un ensemble de fichiers décrits par un fichier de description. Cet usage du mot sème une totale confusion. En effet, avec BitTorrent, l'une des choses les plus compliquées à comprendre est précisément la différence entre le fichier de métadonnées et le(s) fichier(s) de données qu'il décrit !

**peer**

On désigne ainsi un ordinateur à partir duquel on peut télécharger des morceaux de fichiers-cibles. En général, une personne qui est en train de télécharger un fichier autorise les autres utilisateurs à télécharger depuis son ordinateur (cela accélère son propre téléchargement). Donc, sauf exception, toutes les personnes utilisant BitTorrent, à un instant donné, sont des peers.

**leecher**

Un leecher est un utilisateur qui, soit ne partage pas du tout ses fichiers, soit ne partage que le fichier qu'il est en train de télécharger. Dès qu'il a récupéré le fichier qui l'intéressait, il en arrête le partage. C'est une utilisation très opportuniste et assez mal vue du Peer2Peer.

**seed**

Un seed est un peer qui détient une copie complète d'un fichier. L'éthique du Peer2Peer veut que toute personne qui a récupéré un fichier via BitTorrent (ou tout autre système) le conserve à la disposition de la communauté des utilisateurs pendant un certain temps.

Celui qui publie la version originale d'un fichier est obligé d'être seed, jusqu'au moment où d'autres seeders auront récupéré l'intégralité du fichier.

---

11. Même s'il peut les limiter, notamment en enrichissant le swarn.

---

**reseed**

À un moment donné, il peut se produire que plus aucune personne possédant une copie complète d'un fichier ne la partage. Il y a bien des *peers* mais aucun *seed*. Pour que les téléchargements en cours puissent se terminer, il faut que quelqu'un possédant une copie complète se dévoue et la partage de nouveau. C'est donc un nouveau *seed*.

**swarn**

On désigne ainsi un ensemble d'ordinateurs qui sont en relation, à un moment donné, pour partager un fichier. Si votre application BitTorrent vous indique que vous êtes en relation avec 10 *peers* et 3 *seeds*, il y a donc un *swarn* de 14 ordinateurs.

**tracker**

Un serveur sur l'internet qui coordonne l'activité d'ordis utilisant BitTorrent. Quand vous "lisez" un torrent, votre ordi va consulter le tracker indiqué dans ce torrent pour savoir quels sont les ordis disposant de morceaux du fichier-cible décrit par le torrent. Tout au long du téléchargement, votre ordi informe périodiquement le tracker des morceaux de fichier qu'il a déjà récupéré. Ainsi le tracker peut informer d'autres ordis qu'ils peuvent se les procurer auprès de vous.

Si un tracker tombe en panne pendant que des ordis s'échangent des bouts d'un fichier dont il coordonne la publication, les échanges entre *peers* qui ont débuté iront à leur terme. Mais aucun *peer* ne saura à quel ordi s'adresser pour poursuivre le téléchargement au delà des morceaux en cours.

**distributed copies**

Il peut arriver qu'aucun ordinateur ne possède la copie intégrale d'un fichier (aucun *seed*) mais que l'ensemble des ordis d'un *swarn* la possède. Soit un livre de 30 pages, si l'ordi A possède les pages 1 à 15 et l'ordi B les pages 16 à 30, alors les ordi A et B possèdent, à eux deux, une copie intégrale du livre. Certaines applications BitTorrent vous indiquent cette situation sous l'appellation *distributed copies*.

[Retour au sommaire](#)

**node**

On désigne ainsi un ordinateur participant au réseau DHT, utilisé conjointement à BitTorrent. DHT étant un réseau pair à pair, un *node* est, au sens littéral, un pair (*peer*) du réseau DHT. On utilise le terme *node* pour éviter la confusion entre *node* (DHT) et *peer* (BitTorrent). Un ordinateur peut être uniquement un *node*, uniquement un *peer*, ou les deux à la fois.

---