

# Optimisation de la bande passante dans un réseau pair-à-pair : la stratégie BitTorrent face à ses challengers

Fabien de Montgolfier

LIAFA, Université Paris 7  
fm@liafa.jussieu.fr

---

Les protocoles d'échanges Pair-à-pair actuels incitent leurs clients à échanger les fichiers grâce à un système d'*échanges de blocs* rappelant l'économie libérale. Cette étude présente des algorithmes génétiques permettant de déterminer «la» meilleure stratégie de négociation des blocs dans un réseau Pair-à-pair de type BitTorrent, c'est-à-dire celle qui maximise la vitesse de téléchargement.

**Keywords:** pair-à-pair, optimisation, protocole BitTorrent, algorithmes génétiques

---

## 1 Introduction

Les protocoles de téléchargement pair-à-pair ont beaucoup évolué depuis leurs débuts (Napster, Gnutella...) Ils étaient initialement basés sur un concept client/serveur distribué : plusieurs pairs<sup>†</sup> possédaient un fichier qui était téléchargé par d'autres pairs. Mais la quantité de gens «partageant» un fichier se réduisait trop par-rapport à celle des *leechs* (téléchargeurs passifs), ce qui ramenait à la problématique classique de la saturation de bande passante en sortie des serveurs.

### 1.1 Le protocole BitTorrent

Les protocoles de la génération suivante tentèrent de créer un intérêt du client à partager ses fichiers. Parmi les meilleurs systèmes sont ceux où la générosité des premiers protocoles a été remplacée par un modèle «libéral» : chaque fichier est découpé en *blocs*. Les clients ne possédant qu'une partie du fichier peuvent échanger les blocs qu'ils ont déjà contre d'autres. Ainsi, il suffit que très peu de pairs aient la totalité des blocs pour que le système fonctionne, sans surcharger la bande passante de personne.

Le protocole BitTorrent a été inventé par Cohen dans ce but [Coh03]. Il est basé sur le *generous tit-for-tat* : chaque client donnera les blocs qu'il possède préférentiellement aux pairs qui lui ont fourni le plus de blocs. De plus, afin d'alimenter les nouveaux arrivants ou ceux qui ont un débit très limité, on donne aussi des blocs «généreusement». Il s'établit donc une négociation implicite entre les pairs, chacun essayant par essais-erreurs de trouver ceux qui lui fourniront le plus de blocs, et en même temps alimente ses clients<sup>‡</sup>.

### 1.2 Objet de cette étude

Dans cette étude nous ne nous intéressons pas à la manière dont un pair recherche un fichier ni comment il se connecte à d'autres pairs, mais seulement à son protocole de téléchargement. Nous proposons une méthode à base d'algorithmes génétiques, semblable à ce qui a été fait pour le dilemme du prisonnier itéré [Axe87, BDM96] afin d'identifier les meilleures stratégies. On trouvera donc une description des paramètres des stratégies, du protocole de simulation, et quelques résultats d'expérience.

---

<sup>†</sup> un *pair* est un utilisateur, situé sur une machine et avec un logiciel, qui «partage» ses fichiers

<sup>‡</sup> on appellera par la suite «*clients*» d'un pair tous les autres pairs avec lesquels il est en relation, peu importe le sens du transfert

## 2 Paramètres d'une stratégie

Examinons maintenant comment écrire une stratégie dans un réseau BitTorrent. Les stratégies considérées doivent être assez simple afin que leur espace puisse être «parcouru» par un algorithme génétique, et suffisamment complexes pour comprendre la stratégie BitTorrent originelle ainsi que la plupart des stratégies simples auxquelles on peut penser. Il me semble que les points suivants doivent être pris en compte :

- **Récompense de la fidélité** Le principe de base est que ceux qui ont le plus envoyé seront ceux qui recevront le plus. Ce renforcement mutuel assure la stabilité des échanges.
- **Générosité** Une stratégie peut avoir intérêt à envoyer des données vers des gens qui n'ont pas de relation avec elle. Loin d'être un acte de pur générosité, il s'agit en fait de se signaler : c'est la seule façon d'entamer les «négociations» avec le pair.
- **Horizon temporel** L'évaluation de la performance d'un client est sa performance (réceptions et émissions) moyenne sur une certaine période. Un petit horizon temporel permet d'être réactif tandis qu'un grand lisse les irrégularités éventuelles de transmission.
- **Durée des choix** Une stratégie peut maintenir ses choix de clients un certain temps : cela permet par exemple d'attirer l'attention des nouveaux clients en leur donnant plus. BitTorrent ne réévalue que toutes les 10 secondes.
- **Auto-limitation** Pour ne pas «gaspiller» la bande passante, on peut majorer la différence *envoyé – reçu*. Par exemple une stratégie *donnant-donnant* majore les émissions par les réceptions, et BitTorrent n'a pas de majorant. Les «roublards» aiment un majorant négatif !

En fonction de ces considérations, voici un algorithme qui représente la stratégie universelle.

- Répéter
  - Avec une probabilité  $g$  envoyer un paquet de  $r$  blocs à un client aléatoire
  - trier les clients selon leurs  $h$  dernières actions suivant un certain critère  $T$
  - répartir  $u$  blocs aux  $c$  meilleurs clients
    - soit proportionnellement à ce qu'ils ont envoyé (mais tel que  $\text{don} - \text{reçu} < 1$ )
    - soit selon un schéma prédéfini  $S$  (mais tel que  $\text{don} - \text{reçu} < 1$ )
  - envoyer les blocs restant, par paquets de  $r$ , à des pairs choisis au hasard
  - Envoyer de la sorte pendant  $d$  secondes ( $d$  tours)

L'algorithme de tri  $T$  peut être aléatoire (ce qui n'est pas très intéressant), par tailles de téléchargement décroissantes (on privilégie ceux qui ont donné le plus) ou par rapports réception/émission décroissantes (on privilégie ceux qui ont le meilleur «rendement»).

## 3 Protocole de simulation

Un tournoi contient un certain nombre de pairs (100 dans cette expérience). Chaque pair dispose de la même bande passante et tous doivent télécharger un fichier de la même taille (autant de blocs ; un bloc est téléchargé atomiquement). A chaque instant, un client peut envoyer un nombre maximal de blocs donnés. Par analogie avec une ligne ADSL, ce nombre est fixé à 16, et il n'y a pas de limite de réception.

Si le schéma de répartition est prédéfini, on donnera toujours  $c_1$  blocs au meilleur client,  $c_2 \leq c_1$  au second et ainsi de suite. équité. Comme il y a au plus 16 blocs envoyés, cela fait 916 possibilités pour le schéma  $S$ . le *patrimoine génétique* d'une stratégie est donc  $[S, g, r, h, T, u, c, l, d]$

### 3.1 Sélection

Les meilleures stratégies sont choisies par une méthode darwinienne. La performance d'une stratégie est le temps qu'elle met à compléter ses dix premiers téléchargements (dès qu'un téléchargement est terminé, un pair jouant la même stratégie mais inconnu des clients en place remplace le précédent) Les 90 meilleurs survivent pour le tournoi suivant (en subissant une mutation) et les 10 éliminés sont remplacés par des «enfants» des 10 premiers.

### 3.2 Mutations

Les stratégies survivantes subissent toutes une mutation : chacun des paramètres est remplacé avec probabilité  $p_r$  par une valeur aléatoire prise uniformément dans l'intervalle et avec une probabilité  $p_g$  par une valeur proche (choisie avec une loi gaussienne de moyenne l'ancienne valeur et d'écart-type 10% de l'intervalle). Nous avons pris  $p_r = 1\%$  et  $p_g = 3\%$ .

### 3.3 Reproduction

La reproduction de deux stratégies suit les lois de Mendel : chaque paramètre de l'enfant a 25% de chances d'être la valeur du «père», 25% d'être la valeur de la «mère» et 50% une moyenne (pour les 7 paramètres numériques). Pour maintenir constante la taille de la population, l'enfant remplace un des deux parents (en plus des 10 enfants des plus forts). Il y a 10 couples qui se reproduisent à chaque nouveau tournoi.

### 3.4 Population initiale

Nous avons fait des expériences avec deux types de population initiales : une population «totalement aléatoire» (chaque paramètre pris uniformément dans son intervalle) et une population fixée comprenant les stratégies suivantes (qui sont assez «naturelles») :

- 5 *méchantes* : ne donnent rien
- 5 *aléatoires* : donnent 1 bloc à 16 clients pris au hasard
- 5 *échos* : donnent autant qu'ils ont reçu aux meilleurs donateurs
- 5 *uniques* : donnent tout (16) au meilleur donateur
- 10 *proportionnelles* : donnent proportionnellement aux 4 meilleurs donateurs
- 10 *proportionnelles réactives* : idem mais avec un horizon de 1 au lieu du maximum
- 40 *BitTorrent classiques* : donne 4 aux 4 meilleurs donneurs, plus proba de 33% de don aléatoire de 4, réévaluation tous les 10 coups
- 10 *roublardes* : comme BitTorrent mais l'émission est majorée par la réception moins 1
- 10 *roublardes sur rendement* : idem mais en triant par rapports reçu/émis décroissants, au lieu de trier par réception décroissantes comme toutes les autres

## 4 Résultats

### 4.1 Comparaison statique des stratégies

Un tournoi (sans algorithmes génétiques)<sup>§</sup> donne comme résultat (le temps moyen de téléchargement étant de 7000 blocs / don de 16, soit 437.5)

1er	roublarde	388.3	6ème	proportionnelle réactive	425.6
2nd	proportionnelle	379.4	7ème	BitTorrent	492.0
3ème	écho	391.8	8ème	aléatoire	582.4
4ème	unique	392.0	9ème	méchante	1868.6
5ème	roublarde sur rendement	418.83			

On peut être surpris de la mauvaise performance de BitTorrent. Cela vient sans doute du fait qu'elle est la seule à réévaluer tous les 10 coups alors que les autres réévaluent à chaque fois. Cependant, comme on va le voir, les algorithmes génétiques vainqueurs sont «à la BitTorrent».

### 4.2 Algorithmes génétiques

La surprise des algorithmes génétiques<sup>¶</sup> est la disparition totale des stratégies proportionnelles. Au bout d'une cinquantaine de tournois, ne survivent que des stratégies à schéma fixé qui trient les clients par nombre de blocs reçus décroissants. Les 10 meilleurs stratégies après un million de tours de simulation (300 cycles

<sup>§</sup> programme disponible sur <http://www.liafa.jussieu.fr/~fm/P2P/SimuStat.tgz>

<sup>¶</sup> programme disponible sur <http://www.liafa.jussieu.fr/~fm/P2P/SimuGen.tgz>

de reproduction) dans une simulation donnée sont données par ce tableau, où la première colonne est le schéma de don (par exemple [4 3 2] indique que l'on donne 4 blocs au meilleur client, 3 au second et 2 au troisième). Si la somme n'est pas 16, les autres sont toujours donné au hasard. Les résultats d'une autre simulation diffèrent très peu si l'on garde les mêmes paramètres (nombre de clients, taux de sélection) mais que l'on change la population initiale.

schéma	horizon	durée	limitation	taille blocs	générosité
	h	d	l	aléa. r	g
[3 3 3 3 2 2]	3	17	+5	5	4%
[3 2 2 2 2 2]	10	16	+7	7	4%
[3 3 3 3 2 2]	14	13	+10	8	7%
[3 3 3 3 2 2]	11	11	+4	6	5%
[4 4 4 4]	6	13	-2	9	2%
[4 4 3 2 2 1]	8	15	+15	8	4%
[3 3 3 3 2 2]	17	14	+11	4	6%
[3 3 3 3 2 2]	18	11	+1	4	1%
[5 4 4 2 1]	17	13	+8	7	0%
[3 3 3 3 2 2]	16	13	+9	6	6%
[4 4 4 4]	20	10	+16 (= +∞)	4	33%

Il est surprenant de constater la grande similitude de ces stratégies avec la stratégie BitTorrent classique (ce que le tournoi classique ne laissait pas du tout présager d'ailleurs) qui est donnée pour mémoire dans la dernière ligne du tableau. Cela valide le choix d'un horizon temporel long, mais également d'une longue durée de maintien des choix. Les principales différences des stratégies gagnantes sont dans le nombre plus élevé de clients (6 contre 4) et le coefficient de générosité beaucoup plus bas (presque négligeable). Certaines stratégies ont une limitation assez forte (on trouve même des stratégies «roublardes» qui donnent moins que ce qu'elles ont reçu) mais les stratégies à limitation faible semblent très bien survivre aussi : le paramètre n'a pas l'air d'être un facteur évolutif majeur.

## 5 Conclusion

Lorsque j'ai commencé ces expériences, je me demandais (sur une question qui m'a été posée par Laurent Viennot) s'il existait des stratégies meilleures que BitTorrent sur le protocole BitTorrent. Cette étude semble répondre par la négative. Les principales différences sont le nombre de clients (que tous les clients actuels permettent de régler) et le facteur de générosité (qui semble trop grand dans BitTorrent). Le principe originel de BitTorrent est un *generous tit for tat*, mais il semble que le côté «donnant-donnant» prédomine sur le côté «généreux» pour une efficacité maximale. À ce détail près, il est validé.

On peut se demander pourquoi ne pas individualiser les blocs : la rareté des blocs peut être utilisée dans la négociation. Le *bloc rare* [MR04] finit par disparaître du réseau et bloque les téléchargements de tous les utilisateurs à 99%. Des protocoles en développement utilisent un mécanisme de redondance des blocs permettant d'éviter ce problème en échange d'une augmentation minimale de la taille du fichier. Cette étude dissocie ces deux facettes d'une stratégie pair-à-pair qui sont assez orthogonales.

## Références

- [Axe87] R. Axelrod. *Genetic Algorithms and the Simulated Annealing*, L. Davis ed., chapter The Evolution of Strategies in the Iterated Prisoner's Dilemma, pages 32–41. Pitman, 1987.
- [BDM96] Bruno Beaufile, Jean-Paul Delahaye, and Philippe Mathieu. Our meeting with gradual : A good strategy for the iterated prisoner's dilemma. In *Proceedings of the International Conference on Artificial Life V (ALIFE V)*, pages 159–165, 1996.
- [Coh03] Bram Cohen. Incentives build robustness in bittorrent, 2003.
- [MR04] Fabien Mathieu and Julien Reynier. File sharing in p2p : Missing block paradigm and upload strategies. Technical Report RR 5193, INRIA, 2004. <http://www.inria.fr/rrrt/rr-5193.html>.