Loi de conservation de la bande passante

Farid Benbadis, Fabien Mathieu

Loi de conservation de la bande passante

Farid Benbadis[1] and Fabien Mathieu[1]

[1]*Orange Labs, 38–40 rue du général Leclerc, 92794 Issy-les-Moulineaux, France*

Dans cet article, nous proposons des limites théoriques de performance obtenues par la seule loi de conservation de la bande passante. Pour des cas de débit constant (de type *streaming*), nous voyons comment, par effet de levier ou par usage des clients inactifs, diffuser un flot alors que les clients n'ont pas le débit montant (*upload*) moyen nécessaire. Pour le téléchargement pair-à-pair (débit élastique), nous montrons une limite de tolérance aux free-riders[†] en l'absence de sources, et à l'inverse, l'existence de régimes stationnaires de sur-approvisionnement (*sur-seeding*) où le téléchargement est arbitrairement rapide à condition que chacun s'acquitte ensuite d'un certain temps d'approvisionnement (*seeding*).

Keywords: pair-à-pair, loi de conservation, streaming, BitTorrent

1 Introduction

Dans tous les réseaux fermés (pas d'échanges avec l'extérieur), réseaux pair-à-pair inclus, à chaque instant, la somme des débits reçus (*download*) est égale, aux délais de transmission près, à la somme des débits émis (*upload*) : c'est la loi de conservation de la bande passante, qui peut se résumer par cette adaptation d'une citation célèbre de Lavoisier : *(presque) rien ne se perd, (presque)*[‡] *rien ne se crée, tout se transfère.*

Dans cette article, nous allons jouer avec la loi de conservation. En considérant la faisabilité en terme de bande passante, nous donnons des conditions nécessaires et des performances maximales théoriques. Ces résultats fixent des limites de référence à considérer dans la conception d'un système réel. Nous regardons d'abord le problème de la diffusion à débit constant (section 2), avant de nous intéresser à la diffusion à débit élastique (section 3).

2 Diffusion à débit constant

Nous commençons par l'étude du problème suivant : n utilisateurs désirent profiter d'un *service* qui nécessite un débit fixe d. Ce service peut être, par exemple, une émission diffusée en directe (de type *live* streaming, comme PPLive [HLL+06]), ou de la vidéo à la demande (comme Joost [Joo] et Push-to-Peer [SDK+07]). Chaque utilisateur (pair) dispose d'une capacité d'upload αd dédiée au service.

Pour $\alpha \geq 1$, la loi de conservation nous dit que le système formé par les pairs peut être auto-suffisant. Des solutions pour le live streaming [CDK+03] et la vidéo [BMdM+08] à la demande sont d'ailleurs proposées pour $\alpha \geq 1+\varepsilon$. Pour le cas considéré ici, $\alpha < 1$, que peut-on faire ?

2.1 Effet de levier

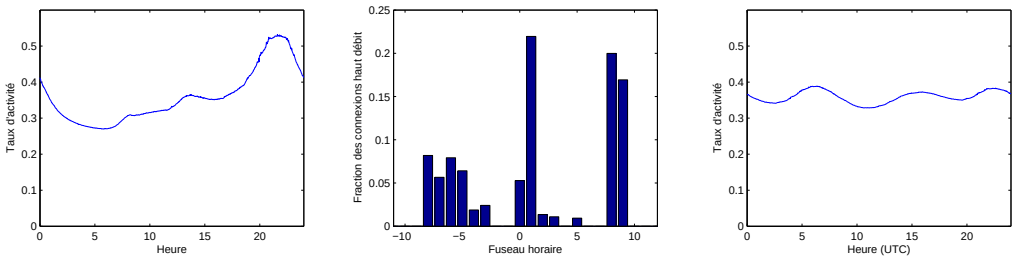
Supposons que le service provient d'un serveur de capacité $n_0 d$ (peut fournir directement n_0 pairs). La capacité du système (serveur,pairs) étant de $d(n_0+\alpha n)$, la condition de faisabilité est $n \leq n_0 + \alpha n$, soit $n \leq \frac{n_0}{1-\alpha}$. Grâce au pair-à-pair, la capacité initiale du serveur est multipliée par $\frac{1}{1-\alpha}$ (effet de levier). Un tel système, bien que non scalable, permet de réaliser une économie certaine au niveau serveur.

2.2 Utilisation des pairs inactifs

Dans le cas précédent, nous avons supposé que seuls les pairs désirant le service participaient à sa diffusion. Mais ces n pairs actifs font souvent partie d'un ensemble $N > n$ de pairs potentiels, dont $N-n$ sont donc inactifs. Ainsi, en utilisant les ressources des pairs inactifs, l'application Joost [Joo] est réglée pour tourner en tâche de fond en cas d'inactivité et sert alors de relais à la transmission de vidéos vers d'autres

[†] Terme désignant un individu usant d'un bien collectif et ne payant pas sa quote-part, supportée alors par les autres usagers.

[‡] Le premier *presque* fait référence aux éventuelles pertes durant la transmission, le deuxième rappelle que tout contenu doit bien être créé avant d'être diffusé.



(a) Journée-type de référence. (b) Répartition des connexions haut débit (c) Extrapolation à l'échelle mondiale.

FIG. 1: Taux d'activité et extrapolation mondiale sur une journée-type.

utilisateurs. Le système Push-to-Peer [SDK+07] utilise la bande passante des set-top boxes inactives dans le même but.

La condition de faisabilité de ses systèmes est $n \leq \alpha N + n_0$. En particulier, la condition de passage à l'échelle ($n \to \infty$) est d'avoir $\frac{n}{N} \leq \alpha$. Une diffusion pair-à-pair de contenus streaming nécessite donc que le taux d'activité ne dépasse pas la capacité relative α.

Pour fixer les idées, la figure 1(a) représente le taux d'activité sur une partie du réseau de diffusion de télévision sur IP (IPTV) de Orange. Les mesures sont issues de données effectuées la semaine du 4 au 10 février 2008, agrégées sur une journée-type de 24 heures. Nous constatons que le taux d'activité maximum est d'environ 53%. La condition de faisabilité du même service en pair-à-pair est donc $\alpha > 0.53$.

2.3 Moyennage temporel

Dans l'exemple précédent, il semble dommage de devoir imposer une capacité relative de plus de 53% alors que l'activité ne dépasse 50% que deux heures par jour. La loi de conservation, appliquée non plus instantanément mais sur la journée entière, donne une limite égale au taux moyen, qui est de 36%. Est-il possible d'abaisser la capacité relative requise du taux maximum au taux moyen ?

Une première solution serait de précharger le contenu pendant les heures creuses d'activité. Une telle solution nécessite hélas de connaître, *à priori*, les contenus qui seront demandés et est en particulier inapplicable à du contenu créé en direct.

Une autre solution est d'étendre le réseau à l'échelle mondiale. En effet, à cause du décalage horaire, les plages de fortes demandes ne surviennent pas au même moment. Ainsi, lorsque la demande est la plus forte en Europe (entre 20h30 et 22h30), le service peut bénéficier non seulement de l'apport des utilisateurs européens inactifs, mais également de celui de ceux inactifs d'autres régions du monde se trouvant en dehors de la plage de forte demande. Formellement, si $a(t)$ désigne l'activité locale, supposée indépendante du fuseau horaire, et P la répartition des utilisateurs par fuseau horaire, alors l'activité globale A est la convolée de a par P : $A(t) = \sum_f a(t-f) \times P(f)$.

À titre d'exemple, la figure 1(b) donne la répartition mondiale des utilisateurs haut-débit par fuseau horaire, déduite de données disponibles sur `http://www.internetworldstats.com/dsl.htm`. En supposant que P respecte cette répartition, l'activité globale est indiquée en figure 1(c). La convolution ayant aplati la courbe, l'activité maximale (et donc la capacité relative requise) est inférieure à 39%, ce qui est proche de la valeur minimale possible (36%), et beaucoup mieux que les 53% observés à l'échelle locale.

L'utilisation du décalage horaire peut donner un gain appréciable, qui devra être balancé avec son inconvénient, qui est de reposer sur des liens physiquement éloignés afin de baisser l'activité requise : il y a un compromis à trouver entre d'un côté le gain au niveau des ressources d'accès et de l'autre la surcharge des liens transcontinentaux et les problèmes potentiellement créés par des plus grandes latences.

3 Diffusion à débit élastique

Nous considérons maintenant un système de type BitTorrent [Coh03]. Deux types d'utilisateurs, les *seeders*, qui possèdent le fichier, et les *leechers*, qui cherchent à le récupérer, s'entraident afin de récupérer un fichier donné le plus rapidement possible. Quand un leecher a récupéré le fichier, il devient seeder jusqu'au moment où il quitte le système.

Si l'initialisation d'un ensemble des utilisateurs (*swarm*), où un seeder unique fait face à de multiples leechers, peut-être tumultueuse [MR06], il existe ensuite une phase plus stable [GCX+05] où une analyse

fluide est possible. Nous proposons de nous placer dans un tel régime stationnaire, suivant une version simplifiée du modèle proposé par Qiu et Srikant [QS04]).

3.1 Vitesse de téléchargement

Soit n_l le nombre de leechers à un instant donné, n_s le nombre de seeders. Appelons u_l la distribution d'upload des leechers, et u_s celle des seeders. Sous réserve que la capacité descendante des pairs n'est pas limitante, la loi de conservation revient à étudier la répartition de l'upload sur le download :

- Les seeders distribuent uniformément le fichier aux leechers [Coh03]. Chaque leecher va donc recevoir des seeders un débit $\bar{u_s}r$, avec $r = \frac{n_s}{n_l}$.
- Les échanges entre leechers sont plus complexes. Cependant, si l'on néglige le début et la fin des téléchargements, qui ont des fonctionnements spécifiques, le reste des échanges se décompose ainsi :
  - des échanges réciproques (*Tit-for-Tat*), d'intensité β ($\beta = 0.75$ dans les réglages par défaut de la version originale de BitTorrent). Sous la limite de Dirac proposée dans [GMdMR07], download et upload Tit-for-Tat sont égaux.
  - des échanges généreux uniformes, d'intensité $(1-\beta)$.

En combinant seeders généreux, leechers réciproques et leechers généreux, on obtient le download $d(u)$ d'un pair en fonction de son upload u :

$$d(u) = \beta u + (1-\beta)\bar{u_l} + r\bar{u_s}. \tag{1}$$

En régime non-stationnaire, d peut dépendre du temps. Et si k désigne la taille du fichier, le temps de téléchargement $T_l(u)$ (passé à être leecher) d'un pair d'upload u est la solution de $\int_0^{T_l(u)} d(u, t+t_0)dt = k$. Sous l'hypothèse de stationnarité, cette expression se simplifie en

$$T_l(u)d(u) = k. \tag{2}$$

Résoudre (2) n'est pas simple, car d est fonction de T_l. En appelant $T_s(u)$ le temps moyen passé par les pairs d'upload u à seeder, et p la distribution d'upload des pairs entrant, les valeurs utilisées dans (1) obéissent à

$$r = \frac{\bar{T_s}}{\bar{T_l}}, \text{ avec } \bar{T_s} = \int T_s(u)p(u)du \text{ et } \bar{T_l} = \int T_l(u)p(u)du. \tag{3}$$

$$\bar{u_l} = \frac{\int uT_l(u)p(u)du}{\bar{T_l}} \tag{4}$$

$$\bar{u_s} = \frac{\int uT_s(u)p(u)du}{\bar{T_s}} \tag{5}$$

Regardons maintenant pour quelques cas particuliers comment résoudre (2).

3.2 Système sans seeder

S'il est correctement initialisé et soumis à suffisamment d'arrivées, un swarm peut fonctionner sans seeders [MR06], les leechers quittant le système dès la fin de leur téléchargement. On a alors $T_s = 0$, d'où $T_l(u) = \frac{k}{\beta u + (1-\beta)\bar{u_l}}$. Il s'agit donc de déterminer $\bar{u_l}$. D'après (4), $\bar{u_l}$ vérifie

$$\int \frac{\bar{u_l}p(u)}{\beta u + (1-\beta)\bar{u_l}}du = \int \frac{up(u)}{\beta u + (1-\beta)\bar{u_l}}du \tag{6}$$

Valeurs limites Pour $\beta = 0$ (générosité totale), (6) donne $\bar{u_l} = \int up(u)du = \bar{u}$: l'upload moyen des leechers est l'upload moyen des arrivées (et tout le monde télécharge à la même vitesse). Pour $\beta = 1$ (réciprocité totale), chacun télécharge à sa vitesse d'upload ($T_l(u) = \frac{k}{u}$), et $\bar{u_l}$ est la moyenne harmonique des uploads d'arrivée : $\frac{1}{\bar{u_l}} = \int \frac{p(u)}{u}du$ (et donc $\bar{u_l} \leq \bar{u}$). De manière générale, la réciprocité β ralentit la vitesse moyenne de download : les leechers rapides partent plus vite grâce à la réciprocité et leur bande passante profite moins au système.

Résolution en distribution bi-homogène Si l'on suppose que l'arrivée est constituée de pairs d'upload u_1, avec probabilité p_1, et de pairs d'upload u_2, avec probabilité $p_2 = 1 - p_1$, alors (6) se simplifie en

$$(1-\beta)\bar{u_l}^2 + ((\beta - p_1)u_1 + (\beta - p_2)u_2)\bar{u_l} - \beta u_1 u_2 = 0, \tag{7}$$

qui n'admet qu'une solution positive pour $0 < \beta < 1$, $u_1 > 0$,$u_2 > 0$.

Résistance aux free-riders En prenant (7) avec $u_2 = 0$ (une partie des leechers, les *free-riders*, ne contribuent pas du tout), on trouve deux solutions : $\bar{u_l} = 0$ ou $\bar{u_l} = \frac{p_1-\beta}{1-\beta}u_1$. Ceci nous donne un résultat de tolérance aux free-riders en régime stationnaire : si $p_1 > \beta$, u_l est positif, et tout le monde, free-riders compris, peut être servi en régime stationnaire. Si $p_1 < \beta$, $u_l = 0$: il n'y a pas de régime stationnaire, car les free-riders s'accumulent continuement dans le swarm (ils ne téléchargent pas assez vite par rapport à leur taux d'arrivée). Remarquons que la population p_1 reste stationnaire et télécharge à vitesse βu_1. Avec $\beta = 0,75$ (réglage par défaut), le système peut donc tolérer au plus 25% de free-riders (en tout cas dans le modèle bi-homogène).

3.3 Résolution dans le cas homogène

T_s n'est maintenant plus forcément nul, mais nous supposons que tous les pairs ont le même upload u. T_s et T_l sont alors des constantes, on a $\bar{u_l} = \bar{u_s} = u$ et donc $d = u(1+r)$. Pour T_s fixé, les équations (2) et (3) donnent $T_l u(1+\frac{T_s}{T_l}) = k$. Si l'on pose $T_u = k/u$ (T_u est le temps de transfert d'une copie du fichier à vitesse u), on obtient

$$T_l = T_u - T_s \tag{8}$$

Pour $T_s \geq T_u$, la solution trouvée n'a plus de sens. On peut montrer que lorsque T_s approche T_u, n_l tend vers 0. On sort alors du modèle fluide pour atteindre un état où la capacité des seeders est telle qu'il y a rarement plus d'un leecher à la fois (on parlera de *sur-seeding*). L'équation (1) devient alors inégalité, car de la bande passante est inutilisée : celle de l'unique leecher quand il y en a un, celles des seeders quand il n'y a pas de leecher à fournir. Il suffit donc pour rentrer en sur-seeding que chacun reste seed T_u, soit le temps nécessaire au téléchargement en l'absence de seed. La vitesse de téléchargement devient alors arbitrairement grande (on peut montrer qu'en régime stationnaire, elle est proportionnelle au taux d'arrivée et ne dépend plus de u). Il nous dit aussi qu'au-delà de T_u, le téléchargement s'améliore toujours (on augmente n_s, et n_l vaut 0 ou 1), mais l'upload n'est plus utilisé à son maximum (l'upload moyen utilisé est $u\frac{T_u}{T_s}$).

3.4 Condition de sur-seeding dans le cas général

En s'inspirant du cas homogène, il est possible de trouver la zone critique du cas général. En effet, autour de la zone critique, la quasi-totalité des transferts viennent des seeders, et l'on a donc $d(u) \approx r\bar{u_s}$, et $T_l(u) \approx \bar{T_l}$. En injectant ces approximations dans (2), on obtient la condition de sur-seeding :

$$\int up(u)T_s(u)du \geq k \tag{9}$$

En particulier, si T_s est constant, on obtient la condition $T_s \geq T_{\bar{u}}$, qui étend naturellement le cas homogène.

Références

[BMdM+08] Y. Boufkhad, F. Mathieu, F. de Montgolfier, D. Perino, and L. Viennot. Achievable catalog size in peer-to-peer video-on-demand systems. In *IPTS*, 2008.

[CDK+03] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream : high-bandwidth multicast in cooperative environments. In *SOSP*, pages 298–313, New York, NY, USA, 2003. ACM.

[Coh03] B. Cohen. Incentives build robustness in bittorrent. In *P2P ECON*, 2003.

[GCX+05] L. Guo, S. Chen, Z. Xiao, E. Tan, X. Ding, and X. Zhang. Measurements, analysis, and modeling of bittorrent-like systems. In *IMC*, 2005.

[GMdMR07] A-T. Gai, F. Mathieu, F. de Montgolfier, and J. Reynier. Stratification in P2P networks : Application to bittorrent. In *ICDCS*, 2007.

[HLL+06] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross. Insights into pplive : A measurement study of a large-scale p2p iptv system. In *IPTV Workshop*, 2006.

[Joo] Joost. http ://www.joost.com/.

[MR06] F. Mathieu and J. Reynier. Missing piece issue and upload strategies in flashcrowds and P2P-assisted filesharing. In *P2PSA*, 2006.

[QS04] D. Qiu and R. Srikant. Modeling and performance analysis of bittorrent-like peer-to-peer networks. In *SIGCOMM*, pages 367–378, New York, NY, USA, 2004. ACM.

[SDK+07] K. Suh, C. Diot, J. F. Kurose, L. Massoulié, C. Neumann, D. Towsley, and M. Varvello. Push-to-Peer Video-on-Demand system : design and evaluation. *IEEE JSAC*, 25(9), 2007.