

PROXSEM : Mesure de proximité sémantique pour les systèmes de partage de fichiers pair-à-pair

Yann Busnel*, Anne-Marie Kermarrec†

Institut de Recherche en Informatique et Systèmes Aléatoires,
Campus Universitaire de Beaulieu,
35042 Rennes Cedex - France
{Yann.Busnel, Anne-Marie.Kermarrec}@irisa.fr

Résumé

Les systèmes de partage de fichiers pair-à-pair (SPF P2P) représentent actuellement la plus importante part du trafic sur Internet. L'amélioration de la performance des mécanismes de recherche dans ces systèmes génère de nombreux travaux tant académiques qu'industriels. Dans un réseau P2P, les nœuds sont connectés à un sous-ensemble d'autres nœuds du système avec lesquels ils peuvent communiquer directement. Au sein d'un SPF, chaque nœud possède un ensemble de fichiers qu'il met à disposition des autres nœuds (*cache*). Connecter entre eux les nœuds ayant des centres d'intérêts similaires dans le contexte d'une application donnée permet d'améliorer l'efficacité de la recherche. La plupart des approches proposées se contentent de comparer le contenu des caches de nœuds pour déterminer une proximité d'intérêt. Nous proposons dans cet article PROXSEM, une mesure de proximité affinée prenant également en compte la générosité des nœuds et la popularité des fichiers, facteurs identifiés comme ayant un impact sur la proximité sémantique et la répartition de charge. Nous avons évalué PROXSEM par simulation sur une trace réelle d'un SPF P2P (eDonkey) et les résultats obtenus montrent l'efficacité de cette approche. Alors que la générosité des nœuds peut être aisément évaluée localement, la popularité des fichiers nécessite une connaissance globale du système. Nous proposons également un algorithme épidémique obtenant de manière décentralisée une approximation de la popularité des fichiers.

Mots-clés : Réseaux pair-à-pair, proximité sémantique, protocole épidémique, recherche, équilibrage de charge.

1. Introduction

Système de partage de fichiers pair-à-pair

Le paradigme pair-à-pair (P2P) a récemment montré son efficacité au regard d'applications large échelle très diverses. L'émergence des systèmes P2P a ouvert de nouvelles voies de recherche, notamment concernant le réseau recouvrant³ [4, 16, 20], les flots de données, l'archivage, *etc.* [2, 6, 18]. Le paradigme P2P a été introduit dans les systèmes de partage de fichiers (SPF) qui représentent encore la plus grande consommation de bande passante Internet [17], tels KaZaA [14], eDonkey [7] ou Gnutella [8].

* IRISA / ENS Cachan

† IRISA / INRIA

³ Ensemble des liens logiques connectant les nœuds dans les systèmes P2P

Il est usuel de classifier les systèmes P2P selon leur structuration. Trois grandes classes peuvent ainsi être identifiées : structuré, non structuré et hiérarchique. Ces derniers reposent sur l'existence de super-pairs auxquels les clients sont connectés, comme dans les réseaux KaZaA [14] et eDonkey [7], et ont la charge de servir les requêtes de clients. D'autre part, les réseaux non structurés reposent sur l'absence de structure sous-jacente au réseau. Les mécanismes de recherche utilisés le plus souvent sont le principe d'inondation et la marche aléatoire. C'est le cas des premières versions de Gnutella [8]. A l'inverse, plusieurs systèmes structurés ont été proposés, répondant pour la plupart à la problématique des tables de hachage distribuées [6, 16, 15, 20].

Proximité sémantique

Les infrastructures P2P ont d'abord été optimisées pour prendre en compte la proximité physique [3] (e.g. la latence entre deux nœuds). Récemment, de nombreux travaux ont également étudié d'autres formes de proximité telles que la localité d'intérêt – ou sémantique⁴. Cette proximité prend en compte les comportements des utilisateurs de ces SPF. Par exemple, il est possible d'exploiter les similarités entre certains comportements de téléchargement ou entre le contenu des caches de fichiers des nœuds. Les études présentées dans [11, 21] montrent la pertinence de ces approches par rapport à un choix de voisin aléatoire.

Capturer la proximité sémantique

Différentes approches peuvent être utilisées pour détecter cette proximité entre les nœuds du système. Par exemple, la création de liens sémantiques entre certains nœuds [10, 19] ou l'utilisation d'ontologies prédéfinies (classification sémantique) [5] ont été proposées.

Une analyse fondée sur l'intersection des caches de fichiers dans les SPF P2P a montré l'existence de groupes sémantiques [9]. La taille de cette intersection peut être utilisée comme mesure de proximité sémantique entre tout couple de nœuds du système. L'observation des résultats de cette analyse montre l'existence d'effets de bords dans cette mesure, majoritairement induits par la présence de nœuds généreux et/ou de fichiers très populaires dans le SPF.

Contribution

La contribution majeure de cette étude est la prise en compte de ces facteurs. Nous présentons PROXSEM [1], une mesure de proximité sémantique affinée prenant en compte la générosité des nœuds et la popularité des fichiers. Quelques approches ont été proposées utilisant une proximité sémantique mais, à ce jour, à notre connaissance, aucune ne prend en compte ces facteurs. Cet article est organisé comme suit : la section 2 présente succinctement le paradigme des protocoles épidémiques. Nous introduisons PROXSEM dans la section 3. La section 4 présente l'évaluation de cette mesure. Nous présentons également, en section 5, un algorithme permettant d'estimer la popularité des fichiers de manière totalement distribuée, avant de conclure dans la section 6.

2. Protocole de mise en œuvre de PROXSEM

2.1. Choix du système

Parmi les différents systèmes P2P existants, nous nous plaçons dans le cadre des systèmes de partage de fichiers. Ce choix nous permet de valider notre modèle par la simulation sur différentes traces réelles de SPF.

⁴ Dans le reste de l'article, nous utiliserons le terme de proximité sémantique

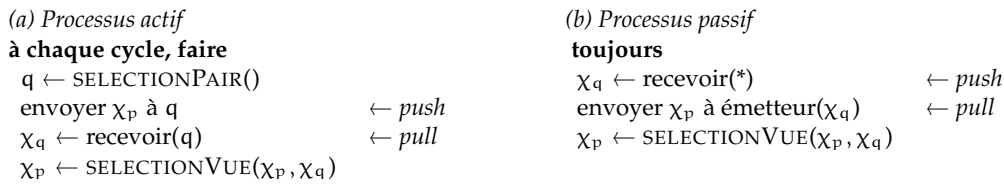


FIG. 1 – Protocole de réorganisation épidémique des liens entre nœuds

Les mots clés en capitales indiquent l'appel à l'une des fonctions paramétrables des protocoles épidémiques. La présence de ($\leftarrow \text{push}$) ou ($\leftarrow \text{pull}$) sur une ligne précise l'utilité de cette dernière en fonction de la propagation choisie. Ici, nous avons représenté le schéma total dit de push/pull.

Désireux d'obtenir un modèle simple et non contraignant, nous choisissons l'utilisation de réseaux pair-à-pair non-structurés dans lequel nous introduisons une détection implicite de la sémantique. Chaque nœud maintient une liste de voisins sémantiques, appelée *vue*. Les nœuds de cet ensemble représentent la seule vision du réseau qu'un nœud possède (vision uniquement locale du système). Chaque nœud possède également, dans ce contexte, un ensemble de fichiers – appelé *cache* – qu'il met à la disposition des autres participants du SPF.

Parmi les différents systèmes de construction et de restructuration du réseau recouvrant³, nous choisissons un modèle de protocole de restructuration épidémique, utilisé par exemple dans T-Man [12] et dans Vicinity/Cyclon [22]. Les raisons de ce choix sont dictées par :

Simple Le modèle est constitué uniquement de deux processus courts, échangeant peu de données.

Efficace L'étude [12] illustre l'efficacité de ce type de protocole pour l'auto-organisation. Celle-ci permet de se rendre compte de la rapidité de convergence de tels protocoles. Pour une heuristique donnée, comme la proximité géographique, la découverte et la mise en place des liens logiques optimaux est faite en seulement 15 cycles.

Décentralisé Tous les nœuds du système jouent le même rôle dans ce protocole permettant un passage à l'échelle aisé.

Non-structuré Ce protocole permet d'éviter d'imposer une structure au réseau recouvrant. Ceci est un choix de mise en œuvre, étant donné qu'il n'est pas prouvé que les aspects structurés soit un désavantage.

La suite de cette partie présente de manière plus précise un modèle architectural des protocoles épidémiques.

2.2. Modèle de protocole épidémique

Une étude [13] propose un cadre et une évaluation des protocoles épidémiques. Celle-ci introduit une structure de base des deux processus acteurs du système, hébergés par chacun des nœuds. Cette structure est présentée en figure 1. Chaque nœud possède une *vue*, notée χ . Le squelette du protocole peut être paramétré à l'aide des trois fonctions :

- Sélection du voisin participant à l'échange : SELECTIONPAIR
- Propagation de la vue : *push*, *pull* ou *push/pull*
- Sélection de la vue : SELECTIONVUE

Dans notre étude, la vue χ a une taille c donnée et les nœuds au sein de celle-ci sont ordonnés selon notre mesure de proximité sémantique PROXSEM par valeur décroissante. Les paramètres du protocole sont les suivants :

- Sélection du voisin : choix d'un nœud aléatoire au sein de χ , pour l'échange.
 - Propagation de la vue : méthode *push/pull* permettant une propagation plus rapide de l'information utile.
 - Sélection de la vue : choix des c nœuds les plus proches selon PROXSEM choisis dans l'union des vues des deux nœuds participant à l'échange.
- Afin de pouvoir l'appliquer dans un contexte sémantique, nous devons déterminer une mesure de proximité d'intérêt entre deux nœuds du système. C'est l'objet de la section suivante.

3. PROXSEM : Mesure de proximité sémantique

Cette partie décrit notre mesure de proximité sémantique. Nous posons, pour le reste de cet article, les notations suivantes :

Soit A et B deux nœuds du système ;

- $|\gamma|$: le cardinal d'un ensemble γ quelconque ;
- $\xi_A(B)$: la mesure de proximité sémantique de B par rapport à A ;
- κ_A : le cache du nœud A ;
- $\sigma_{A,B}$: l'intersection des caches des nœuds A et B
ie. $\sigma_{A,B} = \kappa_A \cap \kappa_B$;
- $\tau_{A,B}$: le nombre de fichiers populaires présents dans $\sigma_{A,B}$
ie. $\tau_{A,B} = |\{f \mid f \in \sigma_{A,B} \wedge f \text{ est populaire}\}|$.

Nous nous intéressons au calcul sur A de $\xi_A(B)$. Plus A est proche sémantiquement de B , plus $\xi_A(B)$ tend vers 1. Inversement, plus les profils sémantiques de A et B sont éloignés, plus $\xi_A(B)$ tend vers 0.

La mesure de proximité fondée sur la taille de l'intersection des caches présentée dans l'introduction est définie de la manière suivante :

$$\xi_A^1(B) = \frac{|\sigma_{A,B}|}{|\kappa_A|}$$

Nous allons l'affiner afin de prendre en compte divers biais : générosité des nœuds et popularité des fichiers.

3.1. Générosité des nœuds

Dans un système de partage de fichiers, chaque nœud propose un nombre plus ou moins important de fichiers aux autres utilisateurs du SPF. Nous nommons cette quantité la *générosité d'un nœud*. Celle-ci peut biaiser l'évaluation de la distance sémantique [10].

Soit A , B et C , trois nœuds ayant respectivement un cache de taille 15, 100 et 15. Supposons que l'intersection des caches de A et B soit de même taille que celle de A et C (10 fichiers en commun). Dans le contexte défini précédemment, le nœud B a la même probabilité que le nœud C d'être sélectionné comme voisin de A . Or, A et C possèdent 66% de leur cache respectif en commun, tandis que pour le nœud B , cela représente seulement 10 % de son cache, il est alors plus probable que A et C soient plus proches sémantiquement.

Afin de déterminer une mesure de proximité sémantique prenant en compte la générosité des nœuds, il est nécessaire d'introduire une relation d'ordre entre les différents cas de figure qui ne sont pas déterminés initialement par la valeur de $|\sigma_{A,B}|$. Ceux-ci correspondent aux cas dans lesquels $\sigma_{A,B}$ a une taille constante mais où κ_A et κ_B sont plus ou moins importants.

La figure 2 représente les différents cas de figure possibles pour une valeur de $|\sigma_{A,B}|$ fixée. Il est nécessaire de les ordonner selon leurs impacts sur la localité d'intérêt. Certains ordres sont triviaux. En effet, sur chaque ligne et chaque colonne de la figure, l'ordre se définit directement

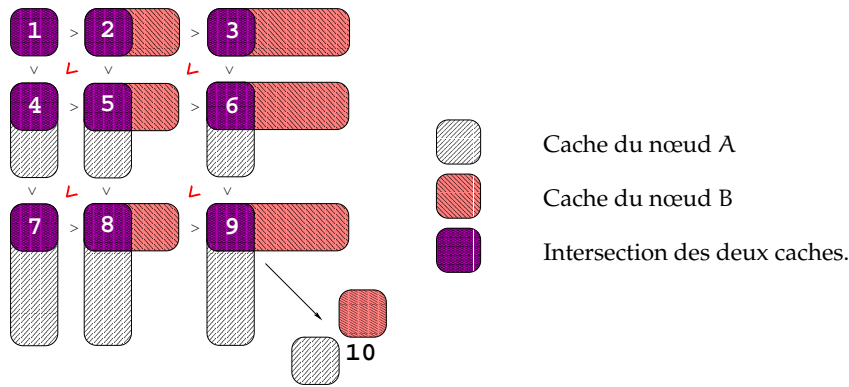


FIG. 2 – Représentation de la relation d'ordre sémantique prenant en compte la générosité des nœuds

(représenté par des symboles ">"). Si $|\sigma_{A,B}| = |\kappa_A| = |\kappa_B|$, alors $\xi_A(B) = 1$. Puis, plus la taille d'un des deux caches grandit, plus leurs profils s'éloignent, et plus la valeur de $\xi_A(B)$ diminue. Nous cherchons donc ξ telle que :

$$\text{Pour } |\sigma_{A,B}| \text{ fixé, } \lim_{|\kappa_A| \rightarrow \infty} \xi_A(B) = 0 \text{ et } \lim_{|\kappa_B| \rightarrow \infty} \xi_A(B) = 0$$

Nous modélisons cette mesure de proximité de la manière suivante :

$$\xi_A^2(B) = \alpha \cdot \frac{|\sigma_{A,B}|}{|\kappa_A|} + \beta \cdot \frac{|\sigma_{A,B}|}{|\kappa_B|} \text{ où } \alpha + \beta = 1$$

Nous devons à présent choisir un ordre entre les cas transversaux. Par exemple, sur la figure 2, quelle est la relation entre le cas 2 et le cas 4? Nous décidons de favoriser l'ordre tel que $\xi_A(B)_{\text{cas 2}} > \xi_A(B)_{\text{cas 4}}$. En effet, les profils sont proches dans chacun de ces cas. La seule différence notable est le potentiel d'obtention de nouveaux fichiers. En effet, il est évident que A dans le cas 4 ne pourra obtenir de nouveaux fichiers de la part de B sans évolution du contenu des caches. Or, dans le cas 2, il ne possède pas de fichiers que B ne possède pas lui-même, mais le reste du cache de B ($\kappa_B \setminus \sigma_{A,B}$) contient des fichiers susceptibles d'intéresser A par la suite. Ainsi, il faut fixer les valeurs de α et β de telle sorte que la mesure de proximité prenne en compte l'ordre ainsi fixé (représenté sur la figure 2). Nous obtenons alors la mesure de proximité affinée suivante :

$$\xi_A^2(B) = \alpha \cdot \frac{|\sigma_{A,B}|}{|\kappa_A|} + \beta \cdot \frac{|\sigma_{A,B}|}{|\kappa_B|} \text{ où } \alpha + \beta = 1 \text{ et } \alpha < \beta$$

3.2. Popularité des fichiers

La popularité des fichiers partagés peut également biaiser la prise en compte de la proximité sémantique. En effet, chaque nœud possède un certain nombre de fichiers dans son cache, et il n'est pas rare qu'un même fichier se retrouve dans de nombreux caches. La valeur de la popularité d'un fichier correspond au nombre de réplicats de celui-ci existant dans le système. Soit A, B et C trois nœuds tels que l'intersection des caches de A et B est de même taille que celle de A et C, soit 15 fichiers en communs. Supposons que $\tau_{A,B} = 10$ et que $\tau_{A,C} = 1$, il est désirable de conclure que A soit plus proche sémantiquement de C qu'il ne l'est de B selon notre mesure.

Supposons qu'à chaque fichier du système soit associée sa valeur de popularité au sein de celui-ci. Ainsi, en fixant un seuil de popularité, nous pouvons déterminer si un fichier est populaire ou non, et de surcroît, le nombre de fichiers populaires présents dans un cache de fichiers ($\tau_{A,B}$). Afin de prendre en compte la popularité des fichiers, nous devons réduire la valeur de $\xi_A^2(B)$ en fonction de la proportion de fichiers populaires présents au sein de $\sigma_{A,B}$. Pour cela, nous avons fait le choix d'ajouter un facteur multiplicatif à notre mesure afin d'en conserver l'homogénéité. Ce facteur, noté $\lambda_{A,B}$, doit suivre les propriétés suivantes :

$$\begin{cases} \lim_{\tau_{A,B} \rightarrow 0} \lambda_{A,B} = 1 \\ \lim_{\tau_{A,B} \rightarrow |\sigma_{A,B}|} \lambda_{A,B} = 0 \end{cases}$$

Ainsi, nous obtenons :

$$\lambda_{A,B} = 1 - \frac{\tau_{A,B}}{|\sigma_{A,B}|}$$

Il est également nécessaire de pouvoir adapter le poids de ce facteur multiplicatif sur la mesure de proximité sémantique. Nous choisissons d'y ajouter un paramètre de puissance γ . L'impact de $\lambda_{A,B}^\gamma$ sur la mesure de proximité, en fonction de différentes valeurs de γ , est la suivante :

- si $\gamma > 1$ alors le facteur permet de favoriser très fortement les nœuds possédant une grande majorité de fichiers rares ;
- si $\gamma < 1$, alors le facteur permet d'exclure les nœuds possédant un grand nombre de fichiers populaires ;
- si $\gamma = 1$, alors la limitation est proportionnelle au nombre de fichiers populaires dans le cache.

Nous obtenons alors la mesure résultante suivante, version complète de PROXSEM :

$$\xi_A(B) = \left(\alpha \cdot \frac{|\sigma_{A,B}|}{|\kappa_A|} + \beta \cdot \frac{|\sigma_{A,B}|}{|\kappa_B|} \right) \cdot \left(1 - \frac{\tau_{A,B}}{|\sigma_{A,B}|} \right)^\gamma \quad \text{où} \quad \begin{cases} \alpha + \beta = 1 \\ \alpha < \beta \\ \gamma \geq 0 \end{cases}$$

La section suivante présente la mise en œuvre pour l'évaluation de cette mesure de proximité ainsi que les résultats des expérimentations et leur analyse.

4. Évaluation des performances

Nous avons développé un simulateur à événements discrets, reproduisant le comportement des nœuds dans un SPF. Afin d'évaluer l'impact de PROXSEM, nous disposons d'une trace réelle, enregistrée sur le SPF eDonkey [7] en novembre 2003. Cette trace a été utilisée pour évaluer d'autres systèmes P2P sémantiques [9, 22], permettant de comparer directement nos différentes stratégies. Celle-ci contient un ensemble de 11 291 nœuds partageant 1 268 536 fichiers distincts, répartis au sein des caches des nœuds. Ces derniers sont utilisés dans le calcul de PROXSEM. Les simulations sont produites avec une stratégie identique à [22] : chaque nœud émet des requêtes sur les fichiers rares de son cache, choisis aléatoirement. Celles-ci ne concernent que les fichiers rares car nous considérons que les fichiers populaires sont plus aisés à obtenir, dû à leur forte répllication sur le système.

Nous présentons ici des résultats globaux pour commencer, puis, nous discutons plus avant chacun des points apportés par PROXSEM.

Afin d'homogénéiser les résultats, les valeurs des paramètres sont les mêmes quelle que soit la simulation :

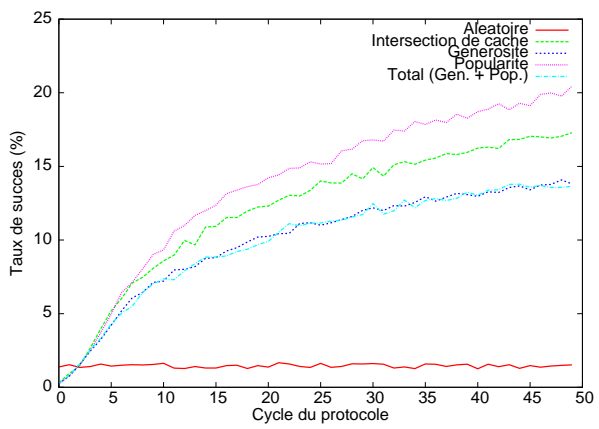


FIG. 3 – Taux de succès à un saut en fonction de l'avancement du protocole épidémique

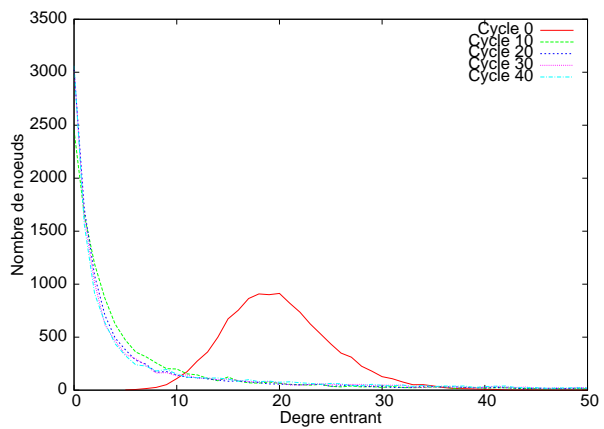


FIG. 4 – Distribution des nœuds du système en fonction de la taille de leur degré entrant et de l'avancement de la procédure

Cette courbe présente la distribution des degrés entrants du graphe de connexions du réseau recouvrant tous les 10 cycles du protocole de réorganisation épidémique.

- taille de la vue : $c = 20$;
- nombre de cycles : $n = 50$;
- limite de générosité : 300 fichiers ;
Les nœuds possédant plus de 300 fichiers seront considérés comme généreux par le système ;
- limite de popularité : 10 réplicas ;
Les fichiers possédant plus de 10 réplicas seront considérés comme populaires par le système ;
- nombre de fichiers minimum dans le cache : 20 ;
Ce choix permet de ne pas prendre en compte les nœuds consommateurs ou *free-riders*.

4.1. Taux de succès

Une estimation correcte de l'efficacité de notre mesure dans un système de partage de fichiers correspond à la mesure du taux de succès de requêtes sur des fichiers peu populaires, à un saut seulement (*ie.* en ne contactant que ses voisins sémantiques directs).

La figure 3 présente les résultats obtenus pour chacune des mesures de proximité suivantes :

- Aléatoire (*ie.* non sémantique) ;
- Intersection de cache ($\xi_{\lambda}^1(B)$) ;
- PROXSEM avec $\alpha = 1/3$, $\beta = 2/3$ et $\gamma = 0$ (correspondant à $\xi_{\lambda}^2(B)$, inhibe la prise en compte de la popularité) ;
- PROXSEM avec $\alpha = 1$, $\beta = 0$ et $\gamma = 1$ (inhibe la prise en compte de la générosité) ;
- PROXSEM avec $\alpha = 1/3$, $\beta = 2/3$ et $\gamma = 1$.

Nous pouvons observer que si la sémantique n'est pas prise en compte, le taux de succès est très faible. Nous basons donc nos comparaisons sur les résultats obtenus par la stratégie d'intersection des caches.

Par la suite, nous revenons sur chacun des effets de bords. Nous analysons la figure 3 pour chacun d'entre eux afin d'explicitier les différents comportements du système.

Stratégie	Degré entrant maximum
Intersection de cache	1702
PROXSEM	990

TAB. 1 – Maximum In-Degree for different strategies

4.2. Prise en compte de la générosité

Pour tous les résultats présentés ici, la valeur de γ est fixée à 0 afin d'inhiber les effets du facteur multiplicatif prenant en compte la popularité des fichiers.

4.2.1. Taux de succès

Nous observons sur la figure 3 une diminution non négligeable du taux de succès dans le cas titré « Générosité ». Cette diminution s'explique logiquement : la prise en compte de la générosité permet d'obtenir un meilleur équilibrage de charge des nœuds du réseau en diminuant la probabilité de choisir un nœud généreux. Le taux de succès est donc logiquement limité, étant donné que les sources riches, *ie.* les nœuds possédant de très nombreux fichiers, seront moins souvent choisies comme voisins sémantiques et que la mesure du taux de succès s'effectue à un saut.

4.2.2. Équilibrage de charge

En revanche, l'équilibrage de charge illustre l'effet positif de la prise en compte de la générosité des nœuds.

La figure 4 présente la répartition de la charge entrante des nœuds du système *ie.* le nombre de nœuds ayant choisi comme voisin sémantique un même nœud⁵. Cette courbe présente la même allure quelle que soit la valeur de α et β .

Au cycle 0, le réseau recouvrant est initialisé aléatoirement. Nous observons donc logiquement une gaussienne, centrée autour de la taille de la vue des nœuds (ici, 20). Après seulement dix cycles, nous observons que la majorité des nœuds du système possède un degré entrant inférieur à 10. Puis, plus la procédure progresse, plus l'équilibrage de charge s'affine.

En revanche, le tableau 1 permet de comparer l'avantage de la prise en compte de la générosité des nœuds par rapport à la méthode d'intersection des caches. Nous pouvons remarquer que le nœud le plus chargé avec la mesure reposant sur l'intersection des caches possède un degré près de deux fois plus élevé que celui avec PROXSEM ou $\gamma = 0$ (1702 liens contre 990 dans notre méthode).

4.2.3. Influence de α et β

Le tableau 2 résume les plus grands degrés entrants ainsi que les valeurs moyennes du taux de succès pour différentes valeurs de α et β .

Nous pouvons observer que plus α et β sont proches, meilleur est l'équilibrage de charge des nœuds. L'impact de ces derniers sur le taux de succès moyen n'est pas réellement important.

4.3. Prise en compte de la popularité

Nous présentons ci-après les conséquences de la prise en compte de la popularité dans la mesure de proximité. Pour tous les résultats présentés ici, les valeurs de α et β sont fixées respectivement à 1 et 0 afin d'inhiber dans ξ la prise en compte de la générosité des fichiers.

⁵ Le degré entrant d'un nœud correspond au nombre de nœuds l'ayant choisi comme voisin sémantique, soit au nombre d'instance de ce nœud dans la totalité des vues du système.

α	β	Degré maximum	Taux de succès moyen au cycle 50
$\frac{10}{21}$	$\frac{11}{21}$	898	14,53%
$\frac{1}{3}$	$\frac{2}{3}$	990	13,80%
$\frac{1}{51}$	$\frac{50}{51}$	1210	12,23%

TAB. 2 – Influence de α et β sur l'équilibrage des nœuds

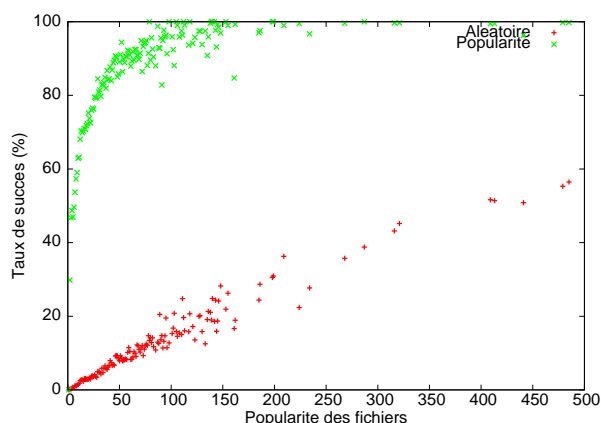


FIG. 5 – Taux de succès en fonction de la popularité des fichiers

Cette courbe permet de mettre en valeur l'efficacité de la recherche à un saut avec une méthode de prise en compte de la sémantique par rapport à une méthode aléatoire

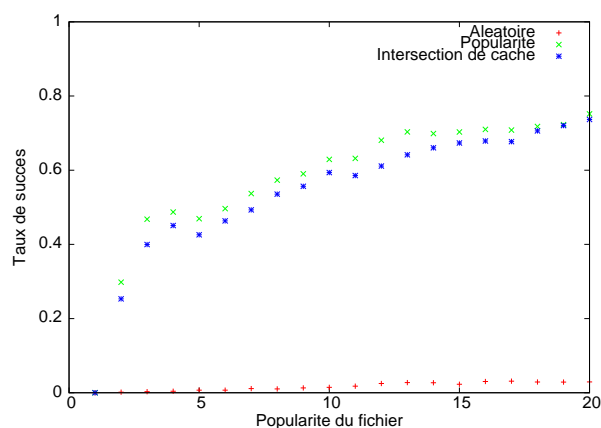


FIG. 6 – Taux de succès en fonction de la rareté des fichiers

Cette courbe permet de mettre en valeur l'efficacité de la recherche à un saut avec ou sans prise en compte de la popularité des fichiers.

4.3.1. Taux de succès

Nous pouvons observer sur la figure 3 une nette amélioration du taux de succès. Comparativement à une moyenne de 17,28 % de taux de succès moyen pour la méthode d'intersection des caches, au 50^{ème} cycle de la procédure, la mesure de proximité prenant en compte la popularité des fichiers permet d'atteindre une moyenne de 20,82 % avec $\gamma = 1/3$.

Cette augmentation non négligeable s'explique par le fait que nous cherchons à optimiser la recherche de fichiers rares. En limitant l'influence des nœuds possédant beaucoup de fichiers populaires, la recherche sur fichiers rares est plus efficace, et présente donc un meilleur taux de succès.

4.3.2. Taux de succès en fonction de la popularité des fichiers

La figure 5 permet d'évaluer précisément le taux de succès en fonction de la popularité.

Cette courbe est issue d'un test effectué au cycle 50 du protocole de réorganisation épidémique. Le cache de chaque nœud est parcouru et, pour chaque fichier, une requête est envoyée aux voisins sémantiques. Un taux de succès est alors associé à chaque fichier, normalisé par sa popularité. Nous obtenons donc la courbe de la figure 5 qui représente le taux de succès en fonction de la popularité des fichiers.

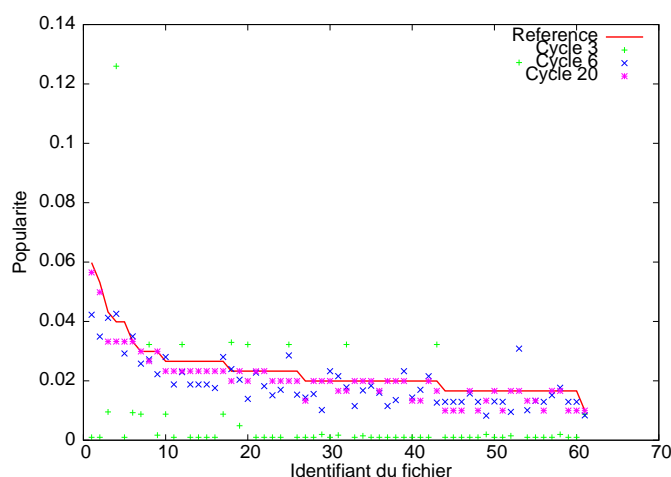


FIG. 7 – Estimation de la popularité avec un protocole épidémique

Celle-ci permet de se rendre compte de l'amélioration de l'efficacité de la recherche de fichiers rares sur ce système de partage de fichiers. En effet, pour les fichiers peu populaires (moins de 20 répliqués sur le réseau), le taux de succès atteint 80 % contre 4 % avec la méthode aléatoire. La figure 6 illustre la différence du taux de succès sur les fichiers rares (moins de 20 répliqués) entre la méthode d'intersection de caches et PROXSEM prenant en compte la popularité des fichiers. Par comparaison, nous représentons également la méthode de réorganisation aléatoire. Nous pouvons observer une nette amélioration pour les fichiers rares : jusqu'à 12% de succès supplémentaire pour PROXSEM.

4.4. Évaluation de $\xi_A(B)$

Nous pouvons observer que lorsque la générosité des nœuds et la popularité des fichiers sont pris en compte séparément, les biais associés sont limités par PROXSEM.

Nous avons mesuré les performances en combinant les deux approches. Nos expériences ont montré que la prise en compte de la popularité n'arrive pas à compenser la diminution du taux de succès induit par la prise en compte de la générosité, dans le cadre des traces disponibles. Comme l'illustre la figure 3, la courbe « Total » côtoie la courbe « Générosité » sans être nettement supérieure. Combiner efficacement ces deux approches représente une perspective attractive.

5. Approximation de la popularité

Précédemment, nous avons supposé que chaque nœud connaissait de manière omnisciente la popularité des fichiers rencontrés.

La générosité d'un nœud peut être aisément évaluée localement, chacun des nœuds n'ayant besoin d'échanger et de comparer que la taille de leur cache. À l'inverse, estimer la popularité d'un fichier, proportionnelle au nombre de répliqués dans un SPF, nécessite une connaissance de tous les caches du système. Afin de déterminer la valeur de τ dans PROXSEM, chaque nœud doit connaître la popularité de chacun des fichiers de son cache. L'objectif de l'algorithme présenté ici est de fournir une approximation de la popularité d'un fichier de manière totalement distribuée.

Pour cela, nous utilisons un protocole épidémique additionnel. La structure de celui-ci est identique à celle présentée en figure 1. À chaque cycle, les nœuds échangent les valeurs de popularité de tous les fichiers qu'ils connaissent. Lors de la d'un nœud au SPF, la valeur de tous ses fichiers est fixée à 1. Pour tout fichier découvert durant un cycle du protocole, la popularité est fixée à 0. Enfin, chacun des nœuds du SPF conserve une moyenne de chaque valeur de popularité échangée. Ainsi, pour chaque fichier rencontré, les nœuds calculent la limite suivante :

$$\text{pop}_f = \frac{\text{nombre de replicats au sein du SPF}}{\text{nombre de noeuds dans le SPF}}$$

De manière à évaluer ce protocole, des expérimentations identiques à la section 4 ont été effectuées. Chaque valeur estimée par le protocole est ainsi comparée à la popularité réelle du fichier. La figure 7 présente les résultats de la simulation. Dès le cycle 6, 60% des valeurs sont correctement ordonnées et après le cycle 20, plus de 80% le sont.

Ce protocole épidémique induit un surcoût acceptable : à chaque cycle, les nœuds du système envoient en moyenne deux messages contenant une liste d'identifiants de fichiers et leur popularité associée (un message en tant qu'acteur et un autre, en moyenne, en tant que nœud choisi). Le surcoût principal de cette approche est l'utilisation mémoire. En effet, chaque nœud doit conserver des informations sur chaque fichier rencontré. Nous avons estimé le maximum de surcharge mémoire à 10 Mo par nœud pour un SPF contenant 1 000 000 fichiers.

6. Conclusions et perspectives

La proximité sémantique a été identifiée comme une métrique pertinente pour améliorer la recherche dans les systèmes de partage de fichiers pair-à-pair. De plus, jusqu'à présent, seules des approches simples, reposant sur un historique récent des requêtes ou sur la comparaison des caches de fichiers ont été proposées. Dans cet article, nous avons évalué PROXSEM, une mesure de proximité sémantique affinée permettant de capturer et d'exploiter des relations sémantiques observées entre les nœuds d'un SPF. Le but de PROXSEM est de prendre en compte la popularité des fichiers et la générosité des nœuds au sein de la mesure, ces facteurs ayant été identifiés comme masquant potentiellement l'authenticité des relations sémantiques. La mesure PROXSEM est intégrée dans un protocole épidémique, lequel permet de réorganiser les liens logiques du réseau recouvrant. Nous proposons également un algorithme distribué permettant d'estimer la popularité des fichiers.

L'évaluation des résultats de simulation a montré que la prise en compte de la générosité des nœuds permet d'obtenir un meilleur équilibre de charge. Compte tenu de la popularité des fichiers partagés, le taux de succès moyen est amélioré, particulièrement pour les fichiers rares. La prise en compte d'autres facteurs déterminants tels que la distance géographique entre divers profils sémantiques proches est une éventualité pour l'évolution de PROXSEM. Une perspective attractive est l'intégration de PROXSEM dans d'autres protocoles et systèmes pair-à-pair.

Bibliographie

1. Y. Busnel and A.-M. Kermarrec. Integrating file popularity and peer generosity in proximity measure for semantic-based overlays. Research Report RR-5731, INRIA, IRISA, Rennes, France, Octobre 2005.
2. M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron. Proximity neighbor selection in tree-based structured peer-to-peer overlays. Technical Report MSR-TR-2003-52, Microsoft Research, Cambridge, UK, Septembre 2003.

3. M. Castro, P. Drushel, Y. C. Hu, and A. Rowstron. Exploiting network proximity in peer-to-peer overlay network. Technical report, Microsoft Research, Cambridge, UK, 2003.
4. Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making gnutella-like p2p systems scalable. In *Special Interest Group on Data Communications (ACM SIGCOMM'03)*, San Diego, CA, USA, Août 2003.
5. A. Crespo and H. Garcia-Molina. Semantic overlay networks for P2P systems. Technical report, Database Group, Stanford University, Stanford, CA, USA, Septembre 2002.
6. P. Druschel and A. Rowstron. Past : A large-scale, persistent peer-to-peer storage utility. In *The 18th ACM Symposium on Operating Systems Principles (SOSP'01)*, Lake Louise, AL, Canada, Octobre 2001.
7. The eDonkey 2000 project. <http://www.edonkey2000.com/>
8. The Gnutella project. <http://www.gnutella.com/>
9. S. Handurukande, A.-M. Kermarrec, F. L. Fessant, and L. Massoulié. Clustering in peer-to-peer file sharing workloads. In *The 3rd International Workshop on Peer-to-Peer Systems (IPTPS'04)*, San Diego, CA, USA, Février 2004.
10. S. Handurukande, A.-M. Kermarrec, F. L. Fessant, and L. Massoulié. Exploiting semantic clustering in the edonkey P2P network. In *The 11th ACM SIGOPS European Workshop (SIGOPS'04)*, Leuven, Belgium, Septembre 2004.
11. S. Handurukande, A.-M. Kermarrec, F. L. Fessant, L. Massoulié, and S. Patarin. Peer sharing behaviour in the eDonkey network, and implication for the design of server-less file sharing systems. In *EuroSys'06*, Leuven, Belgium, Avril 2006.
12. M. Jelasity and O. Babaoglu. T-Man : Fast gossip-based construction of large-scale overlay topologies. Technical Report UBLCS-2004-7, University of Bologna, Department of Computer Science, Bologna, Italy, Mai 2004.
13. M. Jelasity, R. Guerraoui, A.-M. Kermarrec, and M. van Steen. The peer sampling service : Experimental evaluation of unstructured gossip-based implementations. In *ACM/IFIP/USENIX 5th International Middleware Conference (Middleware'04)*, Toronto, Ontario, Canada, Octobre 2004.
14. The KaZaA project. <http://www.kazaa.com/>
15. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Special Interest Group on Data Communications (ACM SIGCOMM'01)*, San Diego, CA, USA, Août 2001.
16. A. Rowstron and P. Druschel. Pastry : Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *The IFIP/ACM International Conference on Distributed Systems Platforms – Middleware (IFIP'01)*, Heidelberg, Germany, Novembre 2001.
17. S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy. An analysis of internet content delivery systems. In *The 15th Symposium on Operating Systems Design and Implementation (OSDI'02)*, Boston, MA, USA, Décembre 2002.
18. The Skype project. <http://www.skype.com/>
19. K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM'03)*, San Francisco, CA, USA, Avril 2003.
20. I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord : A scalable peer-to-peer lookup service for internet applications. In *Special Interest Group on Data Communications (ACM SIGCOMM'01)*, San Diego, CA, USA, Août 2001.
21. S. Voulgaris, A.-M. Kermarrec, L. Massoulié, and M. v. Steen. Exploiting semantic proximity in peer-to-peer content searching. In *The 10th International Workshop on Future Trends in Distributed Computing Systems (FTDCS'04)*, Suzhou, China, Mai 2004.
22. S. Voulgaris and M. van Steen. Epidemic-style management of semantic overlays for content-based searching. In *Euro-Par'05*, Lisboa, Portugal, Août 2005.